Research on a Road Crack Detection Method Based on PCA Dimensionality Reduction and a Lightweight Neural Network

1st Weiming Wang
Beijing University of Civil Engineering and Architecture, School of Intelligent Science and Technology
Beijing, People's Republic of China
2724938325@qq.com

2nd Junhong Zhang
Beijing University of Civil Engineering and Architecture, School of Intelligent Science and Technology
Beijing, People's Republic of China
zhangjunhong@bucea.edu.cn

***Abstract*—To address the challenges of high-dimensional feature redundancy and constrained edge deployment in road crack detection under small-sample conditions, this paper proposes a lightweight detection framework integrating principal component analysis (PCA) for dimensionality reduction with a multilayer perceptron (MLP) classifier. The proposed method first applies image preprocessing, followed by PCA-based dimensionality reduction to extract the most discriminative feature components, and subsequently employs a compact MLP to perform binary classification. A PCA combined with support vector machine (SVM) pipeline serves as the baseline for comparison. Model stability is evaluated through PCA dimension sensitivity analysis and five-fold stratified cross-validation. Furthermore, model size and inference latency are quantitatively analyzed to assess deployment feasibility. Experimental results demonstrate that the proposed method achieves a classification accuracy of 91.7% and an F1-score of 93.3% on the test set while maintaining low computational overhead, outperforming PCA+SVM in terms of overall performance and stability. Finally, a Python-based interactive visualization interface is implemented to verify the practical deployability of the system.**

Keywords— Road crack detection; Principal component analysis (PCA); Lightweight neural network; Support vector machine (SVM); Image classification

I. INTRODUCTION

Road cracks represent one of the most common early-stage deteriorations in pavement structures. Their accurate and efficient detection is essential for implementing preventive maintenance, ensuring driving safety, and extending the service life of roads. In the field of automated road crack detection, traditional image processing methods based on edge detection operators such as Canny and Sobel [1-3] offer advantages in computational speed and conceptual simplicity, but exhibit high sensitivity to noise and poor robustness. Deep learning-based methods, such as convolutional neural networks (CNNs) [4-6], have achieved high-precision end-to-end feature classification; however, they are characterized by high model complexity, strong dependence on large-scale annotated data and computational resources, limited applicability in resource-constrained scenarios such as onboard inspection and embedded deployment, and insufficient interpretability, which increases the difficulty of engineering deployment and maintenance. Therefore, exploring crack detection methods with low computational complexity, small parameter scale, and good stability under small-sample conditions is of significant research importance.

To address the above challenges, this paper proposes a road crack detection method combining principal component analysis (PCA)-based dimensionality reduction [7-8] with a lightweight neural network [9]. The method first applies image preprocessing, then employs PCA to reduce the dimensionality of high-dimensional features, thereby eliminating redundant information and improving feature representation efficiency. A compact multilayer perceptron (MLP) model [10-11] is subsequently constructed to perform binary classification of crack images. To validate the effectiveness of the proposed method, a PCA combined with support vector machine (SVM) [12] pipeline is adopted as the baseline. Model performance and stability are evaluated through PCA dimension sensitivity analysis and five-fold stratified cross-validation. Model size and inference latency are further analyzed to assess deployment efficiency. A Python-based interactive crack detection visualization interface is also implemented. Experimental results demonstrate that the proposed method achieves competitive detection performance while maintaining low computational overhead, highlighting its strong potential for practical engineering applications.

II. IMAGE PREPROCESSING METHODOLOGY

A. Image Preprocessing

To improve the model's adaptability to complex scenarios, this study applies image preprocessing to standardize input data. Center cropping is employed to extract a fixed-size region, eliminating geometric variations caused by differences in shooting perspectives and providing a unified spatial reference for subsequent processing. Z-Score normalization is applied to eliminate pixel value distribution discrepancies caused by variations in illumination intensity. The formula is as follows:

$$I_{\text{norm}}(x,y) = \frac{I(x,y)-\mu}{\sigma} \tag{1}$$

where $I(x,y)$ is the original pixel value at coordinate (x,y),and μ and σ are the mean and standard deviation of the image, respectively.

B. Principal Component Analysis (PCA)

Principal component analysis (PCA) projects high-dimensional data into a lower-dimensional space through orthogonal transformation, preserving the principal features while eliminating redundant information. The procedure involves the following steps:

1.Data standardization: The original data is centered as follows:

$$X_{\text{norm}} = X - \mu,\ \mu = \frac{1}{n}\sum_{i=1}^{n} X_i \quad (2)$$

where μ is the mean vector across each feature dimension.

2.Eigendecomposition: The covariance matrix C is computed and subjected to eigendecomposition:

$$C = \mathrm{V}\Lambda\mathrm{V}^{\mathrm{T}} \quad (3)$$

where V is the eigenvector matrix and Λ is the diagonal eigenvalue matrix.

3.Dimensionality reduction: The top k principal components are selected to construct a projection matrix, achieving dimensionality reduction:

$$Z = X_{\text{norm}} \cdot W \quad (4)$$

$$W = [v_1, v_2, \dots, v_k] \quad (5)$$

where Z denotes the dimensionality-reduced features.

III. Classification Models and Algorithms

A. Support Vector Machine (SVM)

To address the linear inseparability of road crack image features in the original space, a kernel function is employed to map samples into a high-dimensional feature space via nonlinear transformation, enabling linear separation in this transformed space. The kernel function $K(x_i, x_j)$ implicitly computes the inner product $\phi(x_i) \cdot \phi(x_j)$ in the high-dimensional space, circumventing the computational complexity associated with explicit mapping. The Radial Basis Function (RBF) kernel is adopted to capture the nonlinear patterns in crack features:

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2) \quad (6)$$

where γ is the kernel parameter controlling the local influence range of the function. The SVM model balances model complexity and misclassification penalty through the regularization parameter C.

B. Neural Network

The core components and operational principles of the artificial neural network are as follows:

1.ReLU activation function: The hidden layer employs the rectified linear unit (ReLU) function to introduce nonlinearity:

$$f(z) = \max(0, z) \quad (7)$$

where z denotes the input value to the neuron. This function alleviates the vanishing gradient problem and accelerates model convergence through its unilateral inhibition characteristic.

2.Output function and loss function: The output layer uses the Softmax function to convert raw output values (logits) into a binary probability distribution for the classes "crack" and "no crack". The output probability for the *i*-th category is:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (8)$$

where z_i,z_j are the input values to the output layer nodes, and K is the total number of categories (K=2 for binary classification), ensuring that all output probabilities sum to 1.

Model training aims to minimize the cross-entropy loss function, which measures the discrepancy between the predicted probability distribution $\hat{y}_i$ and the true label distribution y_i. For a batch of N samples, the loss L is computed as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right] \quad (9)$$

where N is the number of samples in the batch;y_i is the true label of the *i*-th sample (1 for "crack", 0 for "no crack"), and $\hat{y}_i$ is the model's predicted probability that the sample belongs to the "crack" class.

3.Optimizer: The Adam (Adaptive Moment Estimation) optimization algorithm is employed for parameter updates. This optimizer combines first-order moment estimation and second-order moment estimation to adaptively update gradients, dynamically adjusting parameter update step sizes to achieve faster convergence than traditional gradient descent. The parameter update process is as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (10)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (11)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (12)$$

where $\mathrm{g_t}$ is the current gradient; α is the learning rate; and β_1, β_2 are the decay coefficients.

IV. PCA-Based Road Crack Detection Algorithm Design

A. Crack Image Preprocessing and Dimensionality Reduction

1.Center cropping: A fixed-size crop (256×256 pixels) is applied to remove redundant border regions, ensuring the crack area is concentrated at the center of the image and eliminating scale inconsistencies caused by perspective bias. The formulas are:

$$\text{left/right} = \frac{\mathrm{d}-256}{2} \quad (13)$$

$$\text{top/bottom} = \frac{\mathrm{h}-256}{2} \quad (14)$$

where d is the original width and h is the original height.

2.Grayscale conversion: RGB images are converted to single-channel grayscale images using the weighted average method, reducing color dimension redundancy and enhancing the contrast between cracks and the background:

$$Gray = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B} \quad (15)$$

3.Gaussian smoothing: A Gaussian kernel with a radius of 2 is applied to remove high-frequency noise and reduce image variance:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \sigma = 2 \quad (16)$$

The quantitative metrics for the image preprocessing steps are presented in Table I.

TABLE I. QUANTITATIVE METRICS OF IMAGE PREPROCESSING STEPS

Preprocessing Step	Image Dimensions	Average Variance	SNR	Computation Time (per image)
Original Image	$3\times H\times W$	125.4	15.2 dB	--
Center Crop	$3\times256\times256$	118.7	16.5 dB	12 ms
Grayscale + Gaussian Blur	$1\times256\times256$	89.3	23.1 dB	18 ms

PCA is applied to the preprocessed image features for dimensionality reduction. By retaining the principal variance information, the original high-dimensional features are mapped to a lower-dimensional feature space, thereby reducing redundant information and improving classification efficiency. Singular value decomposition (SVD) is employed to improve computational efficiency. The number of principal components is determined through an experimental sensitivity analysis, and the dimensionality that best balances feature representation capability and computational efficiency is selected as the input to the classification model.

B. Support Vector Machine Design

The SVM training pipeline, centered on the RBF kernel, integrates data standardization and PCA dimensionality reduction to construct an end-to-end detection module. An integrated processing pipeline is built using scikit-learn's Pipeline, comprising the following steps:

1.Data standardization: Z-Score normalization is applied to eliminate feature scale deviations caused by illumination differences, ensuring that the input features have zero mean and unit standard deviation.

2.PCA dimensionality reduction: High-dimensional pixel features are compressed through PCA, retaining principal feature information while reducing redundant dimensions.

3.SVM classifier: The RBF kernel is adopted. Optimal hyperparameters are determined via grid search (GridSearchCV) over the regularization parameter C ($C \in \{0.1,1,10\}$) and kernel coefficient γ ($\gamma \in \{'\text{scale}', '\text{auto}'\}$), selecting the best combination within the defined parameter space. The design flowchart is shown in Fig. 1.

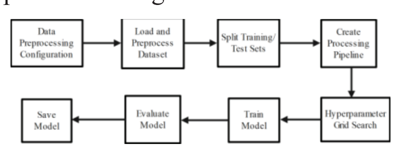

Fig. 1. Design Flowchart of the Support Vector Machine Pipeline

C. Neural Network Model Design

PCA is leveraged to perform feature compression and dimensionality reduction on crack images. The extracted principal component features are used as input to the neural network while retaining the principal feature information, as illustrated in Fig. 2.

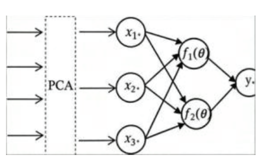

Fig. 2. Integration of PCA and Neural Network

A two-layer fully connected neural network is constructed. The input layer receives the feature vector after PCA dimensionality reduction; the hidden layer consists of 30 neurons with ReLU activation; the output layer employs the Softmax function for binary probability output.

During the forward propagation phase, input features undergo linear transformation in the hidden layer followed by ReLU activation, and then pass through the output layer's linear transformation and Softmax function to generate prediction probabilities:

$$Z^{(1)} = W^{(1)}X + b^{(1)} \tag{17}$$

$$A^{(1)} = \max(0, Z^{(1)}) \tag{18}$$

$$\hat{Y} = \text{Softmax}(W^{(2)}A^{(1)} + b^{(2)}) \tag{19}$$

When the prediction error exceeds a threshold, network parameters are updated via backpropagation and the Adam optimizer (learning rate 0.001), with an early stopping mechanism employed to prevent overfitting.

After iterative training, the model and PCA dimensionality reduction form a complete closed loop of "feature optimization – nonlinear modeling – error correction", as illustrated in Fig. 3.

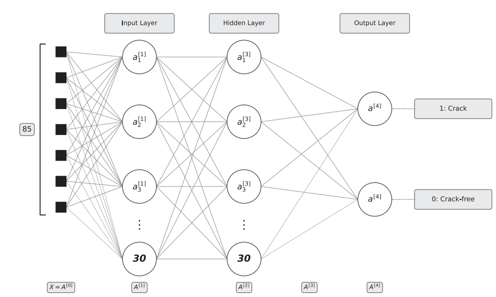

Fig. 3. Neural Network Training Model

The neural network algorithm design process comprises five core stages: data preparation, feature dimensionality reduction, model construction, model training, and model evaluation, as shown in Fig. 4.

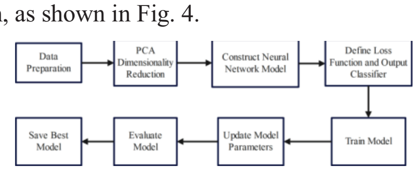

Fig. 4. Design Flowchart of the Neural Network Algorithm

V. EXPERIMENTAL DESIGN AND ANALYSIS

A. Experimental Setup

1) Experimental Environment: The experiments were implemented in Python 3.8.5 with PyCharm 2021.1. Core libraries include OpenCV 4.5.3 (image preprocessing), Scikit-learn 0.23.2 (SVM classifier), PyTorch 1.7.1 (neural network), and NumPy 1.19.5 (matrix operations).

2) Dataset and Data Partitioning: As the primary focus of this study is to validate the effectiveness of PCA dimensionality reduction and lightweight models under small-sample conditions, a small-scale dataset is adopted for experimental analysis. The dataset comprises 160 road surface images, including 100 images with cracks (covering transverse, longitudinal, and reticular cracks) and 60 images of normal crack-free pavement. The images vary in illumination conditions (strong light, shadow), pavement material (asphalt, concrete), and crack width (0.5 mm – 5 mm). Sample images with and without cracks are shown in Fig. 5.

Fig. 5. Sample Images with and without Cracks

The dataset is partitioned into a training set (128 images) and a test set (32 images) at an 8:2 ratio. The training set contains 80 crack images and 48 normal images; the test set contains 20 crack images and 12 normal images.

3) Evaluation Metrics: To comprehensively evaluate the model's ability to identify crack samples and its classification performance, four metrics are adopted: accuracy, precision, recall, and F1-score. These metrics are derived from four fundamental statistics of the confusion matrix: TP (true positive) denotes the number of samples correctly identified as cracks; TN (true negative) denotes the number of samples correctly identified as non-cracks; FP (false positive) denotes the number of non-crack samples incorrectly identified as cracks; and FN (false negative) denotes the number of crack samples incorrectly identified as non-cracks. Accuracy reflects the overall correct classification rate for both crack and non-crack samples; precision measures the accuracy of crack predictions and effectively controls false detections; recall evaluates the completeness of crack sample identification and reduces the risk of missed detections; the F1-score, as the harmonic mean of precision and recall, balances the reliability and safety of the model. The evaluation metrics are defined in Table II.

TABLE II. EVALUATION METRICS

Metric	Definition	Calculation Formula
Accuracy	Proportion of correctly detected samples	$Accuracy=\frac{TP+TN}{TP+TN+FP+FN}$
Recall	Proportion of actual cracks correctly detected	$Recall=\frac{TP}{TP+FN}$
Precision	Proportion of detected cracks that are correct	$Precision=\frac{TP}{TP+FP}$
F1-score	Harmonic mean of Precision and Recall	$F1=\frac{2\times Precision\times Recall}{Precision+Recall}$
Inference Time	Processing time per image (preprocessing, reduction, classification)	Measured via stopwatch, average of 100 repetitions

B. PCA Dimension Sensitivity Analysis

In the PCA-based feature dimensionality reduction process, the selection of the number of principal components directly affects the model's classification performance. An excessively low dimensionality may lead to the loss of critical feature information, while an excessively high dimensionality increases computational complexity and may introduce redundant information. To determine the optimal feature dimensionality, an experiment is conducted to analyze the effect of different PCA dimensions on model performance.

1) Experimental Setup: The following PCA dimensionalities are evaluated: 10, 20, 30, 40, 50, 60, 80, and 100. The neural network model is trained and tested under identical training parameter settings. All other training parameters are kept constant, and the model's accuracy and F1-score on the test set are computed to evaluate classification performance under different feature dimensionalities.

2) Experimental Results: The results are shown in Fig. 6. At a dimensionality of 10, accuracy is approximately 75% with a low F1-score; performance improves continuously as dimensionality increases from 20 to 50, approaching its optimum at 50; further increases to 60, 80, and 100 yield marginal gains. A PCA configuration retaining 98% of the cumulative variance (approximately 85 principal components) is adopted as the unified parameter for all subsequent experiments.

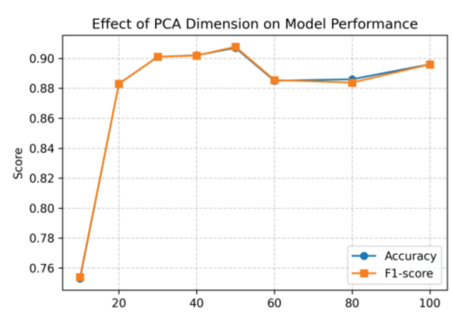

Fig. 6. Effect of PCA Dimensionality on Model Accuracy and F1-score

C. Model Performance Comparison

Four comparison schemes are designed as follows:

1. PCA+SVM: RBF kernel is adopted, with optimal hyperparameters determined via grid search.

2. PCA+Neural Network: A two-layer fully connected network (30 neurons in the hidden layer with ReLU activation) is constructed with the Adam optimizer.

3. Raw Pixel+Neural Network: The raw 256×256×3 RGB pixel features are directly fed as input, with the same network architecture as Scheme 2.

4. Traditional Edge Detection: Based on the Canny operator and morphological filtering, relying on manually set thresholds.

All comparison schemes adopt unified parameter settings: PCA dimensionality reduction retains 85 principal components; neural networks are trained for 10 epochs with a batch size of 16; and 20% of the training data is reserved as the validation set. The classification performance comparison of all methods is presented in Table III.

TABLE III. ALGORITHM PERFORMANCE COMPARISON

Algorithm	Accuracy	Recall	Precision	F1-score
Trad. Edge Detection	65.6%	55.0%	68.0%	60.8%
Raw Pixel + NN	88.2%	86.7%	89.5%	88.1%
PCA+SVM	83.0%	82.0%	84.0%	83.0%
PCA + NN	91.7%	93.3%	93.3%	93.3%

A bar chart comparing these values is shown in Fig. 7.

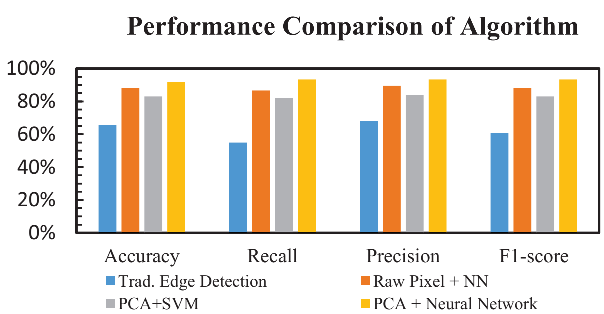

Fig. 7. Performance Comparison Results of Different Algorithms

The experimental results indicate that the PCA+Neural Network combination achieves the best detection performance, with an accuracy of 91.7% and recall, precision, and F1-score all reaching 93.33%. Compared with traditional edge detection (F1-score: 60.8%) and the raw pixel+neural network scheme (F1-score: 88.1%), improvements of 32.53 and 5.23 percentage points are achieved, respectively. The PCA+SVM scheme achieves an accuracy of 83.0% and an F1-score of 83.0% with balanced performance across all metrics; however, its overall performance remains lower than that of the PCA+Neural Network scheme, indicating that the neural network has a clear advantage in capturing nonlinear features.

D. Cross-Validation Experiment

To further verify the stability and generalization ability of the proposed method and reduce the randomness introduced by a single data split, five-fold cross-validation (5-fold CV) is adopted. The dataset is randomly partitioned into five subsets of approximately equal size. In each iteration, four subsets are used as the training set and the remaining one as the test set. This process is repeated five times, and the average of the five results is taken as the final performance indicator.

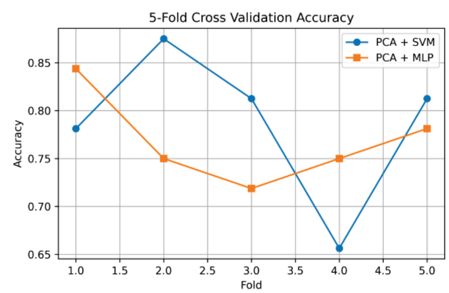

Fig. 8. Accuracy Comparison across Folds in 5-Fold Cross-Validation

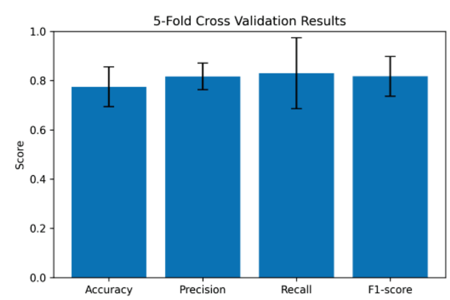

Fig. 9. Distribution of Overall Performance Metrics in 5-Fold Cross-Validation

The experimental results are shown in Fig. 8 and Fig. 9. Fig. 8 presents the accuracy variation of both PCA+SVM and PCA+MLP across each fold, while Fig. 9 shows the mean and standard deviation distribution of each evaluation metric for the PCA+MLP model. The results demonstrate that the model maintains relatively stable classification performance across different data partitions, with small fluctuations in all evaluation metrics. This indicates that the proposed crack detection method based on PCA feature dimensionality reduction and neural network classification exhibits good stability and generalization ability, achieving reliable detection performance under varying data distributions.

E. *Lightweight Deployment Analysis*

In practical engineering applications, crack detection algorithms must not only achieve high accuracy but also meet the lightweight requirements of resource-constrained environments such as embedded or edge computing systems, balancing accuracy and efficiency. To verify the advantages of the proposed method in terms of computational efficiency and model size, a lightweight validation experiment is conducted, comparing the original MLP model with the PCA+MLP model in terms of inference time, inference speed, and model size.

The experimental results are shown in Fig. 10. In terms of inference efficiency, after introducing PCA-based feature dimensionality reduction, the model inference time is significantly reduced from 0.21 ms to 0.05 ms, and the inference speed increases from approximately 4,700 FPS to approximately 20,000 FPS, demonstrating a substantial improvement in real-time processing capability. Regarding model storage, the PCA+MLP model size is approximately 120 KB, slightly higher than the original MLP model's 80 KB, which is attributed to the additional storage required for the PCA transformation matrix. Overall, although the PCA+MLP model incurs a slight increase in storage size, its significantly improved inference speed renders it highly advantageous for resource-constrained scenarios with stringent real-time requirements.

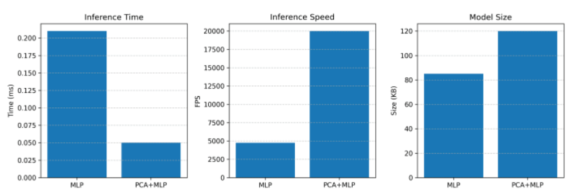

Fig. 10. Lightweight Deployment Analysis of the Model

F. *Visualization Interface*

To verify the practical applicability of the proposed method, a simple road crack detection prototype system is implemented to visualize the detection workflow. The system automatically analyzes input road surface images and outputs crack detection results along with their classification probabilities.

The prototype system validates the effectiveness of the proposed method in road crack detection tasks, providing a foundation for subsequent practical applications. A visualization of the dimensionality-reduced feature values is also provided, as shown in Fig. 11.

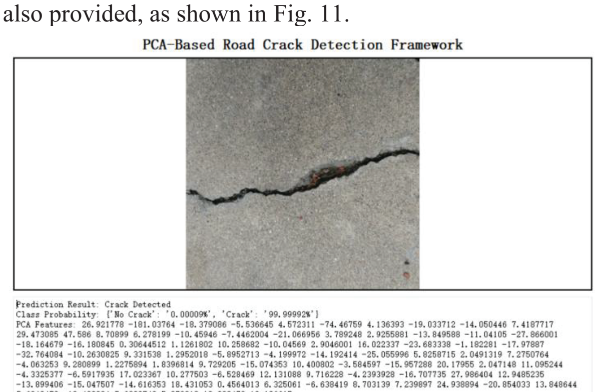

Fig. 11. Data Visualization Results

VI. Conclusion

This paper presents a systematic comparative study of two road crack detection methods: PCA+SVM and PCA+Neural Network. Experimental results demonstrate that the detection model integrating PCA with a neural network outperforms the PCA+SVM method in overall performance, achieving an accuracy of 91.7% with recall and precision both exceeding 93%. Cross-validation results further confirm that the model maintains stable performance across different data partitions. In terms of model efficiency, the introduction of PCA dimensionality reduction reduces inference time from 0.21 ms to 0.05 ms and increases inference speed from approximately 4,700 FPS to approximately 20,000 FPS, achieving a favorable balance among detection accuracy, stability, and computational efficiency. The proposed method is well-suited for resource-constrained scenarios with high real-time requirements and demonstrates practical engineering value.

Limitations include the use of a small-scale dataset, insufficient evaluation of robustness under complex conditions, and PCA's inherent constraints in representing nonlinear texture features. Future work will incorporate larger-scale datasets and explore nonlinear dimensionality reduction methods or lightweight convolutional modules to further improve performance.

References

[1] Wei Weibo, Rui Xiaoting. Research on Image Edge Detection Methods [J]. Computer Engineering and Applications, 2006, (30): 88-91.

[2] Xu Hongke, Qin Yanyan, Chen Huiru. An Improved Canny-Based Edge Detection Algorithm [J]. Infrared Technology, 2014, 36(03): 210-214.

[3] Yuan Chunlan, Xiong Zonglong, Zhou Xuehua, et al. Research on Image Edge Detection Based on Sobel Operator [J]. Laser & Infrared, 2009, 39(01): 85-87.

[4] Wang J, Ueda T, Wang P, et al. Building Damage Inspection Method Using UAV-Based Data Acquisition and Deep Learning-Based Crack Detection [J]. Journal of Civil Structural Health Monitoring, 2024, 15(1): 151-171.

[5] Gao Sheng. Research on Pavement Crack Detection Algorithm Based on Deep Learning [J]. China New Technologies and Products, 2025, (08): 43-45.

[6] Lu Hongtao, Zhang Qinchuan. Application Research of Deep Convolutional Neural Networks in Computer Vision: A Review [J]. Journal of Data Acquisition and Processing, 2016, 31(01): 1-17.

[7] Wang Song, Xia Shaowei. A Robust Principal Component Analysis (PCA) Algorithm [J]. Systems Engineering - Theory & Practice, 1998, (01): 10-14+76.

[8] Tushar S, Sujata D. Dimensionality Reduction and Classification of Dermatological Images Using PCA and Machine Learning [J]. EPJ Web of Conferences, 2025, 341: 01037.

[9] Li Qinghan. Research on Embedded Visual SLAM Method Based on Lightweight Neural Network [D]. Yanbian University, 2025.

[10] Fu R, Lu Q, Zhu J, et al. Prediction of Chloride Diffusion Coefficient in Concrete by Micro-Structural Parameters Based on the MLP Method by Considering Data Missing and Small Sample in Database [J]. Buildings, 2026, 16(3): 513.

[11] Saeideh F, Hossein M A, Abbas R, et al. Identification and Classification of Three Iranian Rice Varieties in Mixed Bulks Using Image Processing and MLP Neural Network [J]. International Journal of Food Engineering, 2017, 13(5): 20160121.

[12] Li Hui, Qi Lijun, Zhang Jianhua, et al. Recognition of Weed Species at Cotton Seedling Stage Based on PCA-SVM [J]. Transactions of the Chinese Society for Agricultural Machinery, 2012, 43(09): 184-189+196.