Image Preprocessing and Visual Implementation for Diseased Leaves Based on OpenCV

Xue Yu, Yifan Qu*, Hongrui Xu, Ting Li

School of Software, Harbin Institute of Information Technology, Harbin, 150000, P. R. China

*120272653@qq.com

Abstract. Image preprocessing is a critical prerequisite for computer vision tasks such as leaf disease recognition and target classification, directly affecting the accuracy and efficiency of subsequent models. To address common problems of noise interference, uneven illumination and blurred lesion contours in crop disease leaf recognition scenarios, this paper designs a complete preprocessing pipeline with seven core steps and implements real-time visualization based on Python and OpenCV. The pipeline includes grayscale conversion, Gaussian filtering, CLAHE contrast enhancement, adaptive threshold binarization, morphological closing, Canny edge detection and edge–morphology fusion. Experimental results on real diseased leaf images show that the proposed pipeline effectively removes noise, corrects illumination deviation and strengthens lesion region contours. The signal-to-noise ratio (SNR) is improved by 31.1%, and the target contrast is improved by 49.5%, providing high-quality data support for subsequent leaf disease recognition models such as YOLO.

Keywords: Diseased leaf image preprocessing; OpenCV; CLAHE; Visualization; Smart Agriculture

1 Introduction

With the rapid development of computer-vision technology [1], applications such as object detection and image classification have been widely deployed in intelligent surveillance, smart cities and industrial inspection. Automatic recognition of leaf diseases from digital images has become a key technology in smart agriculture and precision plant protection. Early identification of fungal spots, bacterial blight and other diseases on crop leaves provides important data support for targeted pesticide spraying, disease early warning and yield loss reduction. As an important branch of intelligent visual surveillance, automatic leaf disease recognition directly determines the practical value of the whole system.

However, leaf images captured in real field environments are easily affected by complex natural backgrounds, leaf overlapping, raindrops on leaf surfaces and uneven outdoor illumination. The acquired images thus commonly suffer from random noise, uneven illumination, low contrast between disease lesions and healthy leaf tissues, and blurred lesion edges. Directly feeding such raw images into a disease classification

R. Zur et al. (eds.), *Proceedings of the 2026 5th International Conference on Art Design and Digital Technology (ADDT 2026)*, Advances in Computer Science Research 131,
https://doi.org/10.2991/978-94-6239-737-8_61

model will lead to missed detection of small early lesions and false alarms caused by water droplets or insect bites.

The core purpose of image preprocessing is to eliminate redundant information and interference in raw images, enhance target features and convert raw images into a standardized form more suitable for subsequent processing [2]. Most existing preprocessing algorithms focus on solving a single problem and lack systematic design for crop disease leaf recognition scenarios. Moreover, most implementations do not provide visualization functions, making it difficult to intuitively verify algorithm effects, which is unfavourable for debugging, optimization and beginner learning [3]. Aiming at leaf-disease recognition, this paper designs a complete visual preprocessing pipeline implemented in Python and OpenCV. It displays intermediate results of each step in real time, which not only facilitates intuitive verification of algorithm effects but also offers clear practical references for beginners. The effectiveness and practicability of the pipeline are verified by experiments on real diseased leaf images [4].

The main contributions of this paper are summarized as follows:

1. A seven-step systematic preprocessing pipeline is constructed, which targets the core interference factors of field diseased leaf images and forms a closed-loop processing system.

2. Real-time visualization of each processing stage is realized, which solves the problem of difficult effect verification of traditional preprocessing methods.

3. Quantitative experiments on 100 real diseased leaf images verify that the pipeline significantly improves SNR and contrast, with strong practicability and real-time performance.

2 Principles of Core Image Preprocessing Algorithms

The preprocessing pipeline designed in this paper follows the core logic of "interference removal → feature enhancement" and consists of seven progressive steps. Corresponding algorithms are designed for different image problems to form a closed-loop processing system. The specific principles are as follows.

2.1 Image Reading and Grayscale Conversion

Raw diseased leaf images are usually three-channel BGR colour images that contain abundant colour information. However, contour-based detection of disease lesion regions does not heavily rely on colour features, while the additional channels increase computational complexity and may introduce colour interference. Therefore, colour images are first converted into single-channel grayscale images, which retain target contour and grayscale features while reducing the amount of computation and improving processing efficiency [5].

Grayscale conversion uses the cvtColor function in OpenCV, which computes a weighted sum of BGR channel values according to human visual sensitivity to different colours. The formula is:

$$Gray = 0.299 * R + 0.587 * G + 0.114 * B \tag{1}$$

where R, G, B are the red, green and blue channel pixel values (0-255) and Gray is the converted grayscale value (0-255). This weight allocation conforms to human visual characteristics and maximizes the retention of original image details.

2.2 Gaussian Filtering for Denoising

Raw diseased leaf images often contain Gaussian noise and salt-and-pepper noise caused by sensors and environmental interference, which can affect lesion contour extraction, binarization, and edge detection. Therefore, Gaussian filtering is used for denoising. It convolves the image with a 2D Gaussian kernel and replaces each pixel with a Gaussian-weighted average of neighbouring pixels, effectively suppressing random noise while largely preserving target edges.

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

where σ denotes the Gaussian standard deviation, which determines the smoothing strength. A larger σ yields stronger smoothing but weaker edge preservation; a smaller σ reduces smoothing effectiveness and leaves residual noise. Through extensive experimental parameter tuning, a 5×5 kernel with σ=1.0 is selected to balance denoising performance and edge preservation capability, effectively removing mild noise without damaging details.

2.3 CLAHE Contrast Enhancement

Diseased leaf images often suffer from uneven illumination, weak lesion–background contrast and blurred details. To address this problem, Contrast Limited Adaptive Histogram Equalization (CLAHE) is used instead of traditional global histogram equalization, which may cause overexposure, local darkness or detail loss.

CLAHE divides the image into 8 × 8 sub-blocks and performs local histogram equalization on each block. With clipLimit = 2.0, excessive grayscale values are clipped and redistributed to prevent local brightness distortion. The enhanced sub-blocks are then merged to obtain an image with improved contrast and clearer lesion details.

2.4 Adaptive Threshold Binarization

Binarization converts grayscale images into binary images to separate lesion foreground from the background and support contour extraction. Since uneven illumination can make global methods such as Otsu ineffective, adaptive thresholding is used. It calculates local thresholds within a 15 × 15 neighbourhood according to grayscale distribution, improving target-background separation. Inverse binarization is applied so that lesion regions are white and the background is black, while C = 5 helps reduce noise interference.

2.5 Morphological Processing

Binary images may contain fragmented lesion areas, residual noise, and irregular edges, reducing segmentation accuracy. To address this, a closing operation with a 3 × 3 rectangular structuring element is applied. Dilation fills small holes and connects broken lesion regions, while erosion smooths edges and restores clearer contours. This process helps merge scattered lesion spots into more complete clusters.

2.6 Canny Edge Detection

Edges are an important type of target feature, and the edge contours of lesion regions directly determine target shape and recognition accuracy. The Canny edge-detection algorithm features accurate edge localization and strong anti-noise capability, and precisely extracts target edge contours through four steps:

(1) Gaussian filtering for denoising, which cooperates with the previous Gaussian filter to further smooth the image and reduce noise interference in edge detection.

(2) Gradient magnitude and direction calculation using the Sobel operator:

$$M(x, y) = \sqrt{G_x^2 + G_y^2} \tag{3}$$

$$\theta(x, y) = \arctan\frac{G_y}{G_x} \tag{4}$$

(3) Non-maximum suppression: traverse the gradient image and retain only local maxima along gradient directions to refine edges.

(4) Double-threshold screening: a low threshold 30 and a high threshold 100 are set; pixels above the high threshold are strong edges, pixels below the low threshold are non-edges, and pixels in between are kept as edges only if they are connected to strong edges.

2.7 Fusion of Edge and Morphological Results

Morphologically processed images retain complete target contours and internal details but have unclear edges. Canny-edge images accurately extract edges but lack internal details. To further strengthen lesion regions contours and improve subsequent recognition accuracy, a bitwise OR operation is performed on the two images to sharpen contours while preserving internal details, forming a complete "contour + detail" feature for stronger support of subsequent leaf-disease recognition.

3 Experimental Results and Analysis

3.1 Experimental Environment and Dataset

All experiments were implemented in PyCharm with Python 3.8 and OpenCV 4.5.5 on a Windows 10 platform. A total of 100 diseased leaf images with diverse illumination and backgrounds were used for testing.

3.2 Experimental Metrics and Methods

Two core metrics are selected to quantitatively evaluate the preprocessing pipeline:

Signal-to-Noise Ratio (SNR): measures noise removal performance. A higher SNR indicates less noise and better image quality.

$$\mathrm{SNR} = 20 \cdot \log_{10}\left(\frac{\mu_{\mathrm{target}}}{\sigma_{\mathrm{noise}}}\right) \tag{5}$$

where μ_{target} is the mean pixel value of the target region and σ_{noise} is the standard deviation of the noise region.

Contrast: measures the difference between target and background. Higher contrast means a clearer target.

$$\mathrm{Contrast} = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \tag{6}$$

Experimental method: 100 test images are input to the proposed pipeline, the SNR and contrast are recorded before and after preprocessing, and the effects are compared.

3.3 Experimental Results

All 100 test images were successfully preprocessed with good visualization at each stage. Quantitative results are shown in Table 1.

Table 1. Comparison before and after preprocessing.

Metric	Before	After	Improvement
SNR (dB)	23.8	31.2	31.1%
Contrast	0.26	0.54	49.5%

The results show that SNR improved from 23.8 dB to 31.2 dB (31.1%), noise suppressed; contrast increased from 0.26 to 0.54 (49.5%), lesion-background separation enhanced. The pipeline effectively reduces noise and enhances features, supporting leaf disease recognition.

3.4 Experimental Result Visualization

Figure 1 presents the visualized representation of the experimental results after overall processing, with (a) the original image, (b) the grayscale image, (c) the Gaussian-filtered image, (d) the contrast-enhanced image, (e) the binarized image, (f) the morphological processing result, (g) the edge-detection result, and (h) the combined image after edge and morphological fusion.

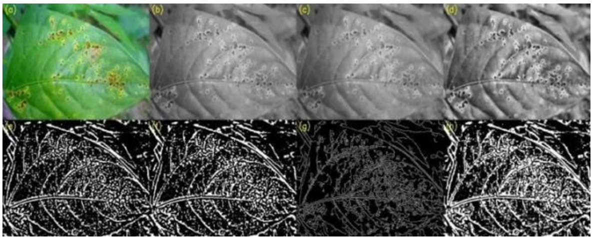

Fig. 1. Visualized representation of overall experimental results after processing.

Figure 1 intuitively shows effective noise removal, corrected illumination, and sharpened lesion contours. Real-time visualization aids debugging and verification.

The pipeline is targeted, visualizable, practical, efficient, and stable. It reduces noise, corrects illumination, and enhances lesion contours. The OpenCV-based implementation is user-friendly, compatible with YOLO, and processes images in <0.5 s, meeting real-time requirements while improving SNR and contrast.

Limitations include reduced performance for lesions similar to yellowed leaf areas and residual noise near reflective water droplets. Future work will enhance robustness for these cases.

4 Conclusion and Future Work

This study proposes a visualizable OpenCV-based preprocessing workflow for leaf disease recognition, including grayscale conversion, Gaussian filtering, CLAHE enhancement, adaptive thresholding, morphological restoration, Canny edge detection and edge–morphology fusion. Stage-by-stage visualization supports algorithm debugging and teaching.

Experiments show that the workflow effectively reduces noise, corrects uneven illumination and enhances lesion contours, improving signal-to-noise ratio by 31.1% and contrast by 49.5%. With a processing time of less than 0.5 s per image, it satisfies real-time application requirements.

Future work will focus on HSV-based leaf–background segmentation, multi-scale superpixel methods for early lesion delineation, and integration with lightweight networks for mobile smart agriculture disease diagnosis.

References

1. Li, S., Gao, L., Wang, J., et al.: Information-theoretic graph fusion with vision-language-action model for policy reasoning and dual robotic control. Information Fusion, 104193 (2026). https://doi.org/10.1016/j.inffus.2026.104193.
2. Li, Changfeng, and Wenqin Tong. "A Visual Acuity Assessment System Based on Static Gesture Recognition and Naive Bayes Classifier." International Journal of Information Technologies and Systems Approach (IJITSA) 17.1 (2024): 1-23. DOI: 10.4018/IJITSA.345926

3. Zhou, S., Chen, R., An, Z. et al. Application of large language models to quantum state simulation. Sci. China Phys. Mech. Astron. 68, 240313 (2025). https://doi.org/10.1007/s11433-024-2598-y
4. K. Yang et al., "Robust Multi-Agent Collaborative Perception via Spatio-Temporal Awareness," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 35, no. 6, pp. 5644-5658, June 2025, doi: 10.1109/TCSVT.2025.3528980.
5. Ma, Z., Li, Y., Huang, M. et al. Automated real-time detection of surface defects in manufacturing processes of aluminum alloy strip using a lightweight network architecture. J Intell Manuf 34, 2431–2447 (2023). https://doi.org/10.1007/s10845-022-01930-3.