Journal of
Experimental and Theoretical Analyses

MDPI

Article

Enhancing Pest Detection in Deep Learning Through a Systematic Image Quality Assessment and Preprocessing Framework

Shuyi Jia [1], Maryam Horri Rezaei [2] and Barmak Honarvar Shakibaei Asli [1,*]

[1] Faculty of Engineering and Applied Sciences, Cranfield University, Cranfield MK43 0AL, Bedfordshire, UK
[2] Independent Researcher, Milton Keynes MK10 9AA, Buckinghamshire, UK
* Correspondence: barmak@cranfield.ac.uk

Abstract

This study addresses the critical challenge of variable image quality in deep learning-based automated pest identification. We propose a holistic pipeline that integrates systematic Image Quality Assessment (IQA) with tailored preprocessing to enhance the performance of a YOLOv5 object detection model. The methodology begins with a No-Reference IQA using BRISQUE, PIQE, and NIQE metrics to quantitatively diagnose image clarity, noise, and distortion. Based on this assessment, a tailored preprocessing stage employing six different filters (Wiener, Lucy–Richardson, etc.) is applied to rectify degradations. Enhanced images are then used to train a YOLOv5 model for detecting four common pest species. Experimental results demonstrate that our IQA-anchored pipeline significantly improves image quality, with average BRISQUE and PIQE scores reducing from 40.78 to 25.42 and 34.94 to 30.38, respectively. Consequently, the detection confidence for challenging pests increased, for instance, from 0.27 to 0.44 for Peach Borer after dataset enhancement. This work concludes that a methodical approach to image quality management is not an optional step but a critical prerequisite that directly dictates the performance ceiling of automated deep learning systems in agriculture, offering a reusable blueprint for robust visual recognition tasks.

Keywords: automated pest identification; deep learning; image preprocessing; image quality assessment; convolutional neural networks; YOLOv5; computer vision

check for updates

Academic Editor: Marco Rossi

Received: 1 September 2025
Revised: 6 November 2025
Accepted: 10 November 2025
Published: 20 November 2025

Citation: Jia, S.; Horri Rezaei, M.; Honarvar Shakibaei Asli, B. Enhancing Pest Detection in Deep Learning Through a Systematic Image Quality Assessment and Preprocessing Framework. *J. Exp. Theor. Anal.* **2024**, *3*, 39. https://doi.org/10.3390/jeta3040039

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The integration of advanced image processing and deep learning represents a frontier in automating complex visual recognition tasks, with significant implications for industrial and engineering applications [1]. A critical, yet often underexplored, prerequisite for the success of these data-driven models is the quality and integrity of the input imagery [2]. Real-world images are frequently degraded by noise, blur, and complex environmental conditions, which can severely impair the performance of even the most sophisticated algorithms [3]. Consequently, the development of robust, integrated pipelines that systematically address image quality before analysis is a fundamental research challenge in computer vision and machine learning.

This work tackles this challenge by proposing and validating a holistic, automated framework for object detection that integrates a systematic Image Quality Assessment (IQA) phase with advanced preprocessing and a state-of-the-art deep learning model. We demonstrate the efficacy and generalizability of this methodology through its application in the domain of food engineering, specifically for automated pest identification—a critical

task for ensuring food security and sustainable production [4]. The core problem we address is a computer vision one: achieving reliable automated detection from a dataset of images captured in unpredictable, non-laboratory conditions [5].

A review of the existing literature reveals that while many studies successfully apply deep learning models to pest detection [6–20], a significant gap remains in the development of integrated, end-to-end workflows that systematically address the foundational issue of input image quality. Many approaches feed raw, often degraded, field images directly into complex models without a principled pre-screening and enhancement phase. This work tackles this gap by proposing and validating a novel, holistic framework. Our primary contribution is a generalizable pipeline that begins with a quantitative No-Reference Image Quality Assessment (IQA) to diagnose issues, proceeds to a tailored preprocessing stage for enhancement, and culminates in robust object detection using a YOLOv5 model. We demonstrate that this IQA-driven approach is a critical enabler for achieving high precision and robustness in real-world agricultural settings.

The variability and noisiness of such field-captured imagery necessitate a methodical approach that begins not with raw data ingestion, but with a quantitative evaluation and enhancement of image quality. Our primary contribution is a novel, IQA-anchored pipeline that first employs no-reference metrics to quantitatively diagnose clarity, noise, and distortion [2]. This diagnostic phase directly informs a tailored preprocessing stage where specialised denoising and deblurring algorithms are applied to rectify common degradations [21]. The enhanced images are then processed through sophisticated segmentation techniques to accurately isolate targets from complex backgrounds. Finally, the refined dataset is used to train a YOLOv5 model for high-performance detection and classification.

The innovation of this paper is its generalizable, image-centric workflow, which is validated within the challenging context of food production. We provide a strong methodological contribution by demonstrating that a rigorous, IQA-driven preprocessing regimen is a critical enabler of deep learning performance, a finding applicable to numerous image-based analysis tasks in food engineering and beyond [22]. Our experimental validation confirms that this integrated approach markedly increases the precision and robustness of automated identification systems, offering a reusable blueprint for building accurate and reliable vision-based detection systems where input data quality is variable.

The primary aim of this study is to develop and validate a holistic, automated framework for plant pest identification that systematically addresses the critical challenge of input image quality. Specifically, we propose a pipeline that integrates a No-Reference Image Quality Assessment (IQA) phase to quantitatively diagnose image degradations, followed by a tailored preprocessing regimen for enhancement, and culminates in robust pest detection using a deep learning model. We demonstrate the efficacy of this IQA-anchored workflow using a dataset of field-captured pest images to significantly improve the precision and robustness of automated identification systems.

This paper is organised as follows: Section 2 provides a review of related work in pest and plant disease identification, image processing techniques, and Image Quality Assessment (IQA). Section 3 details the proposed methodology, including image acquisition, IQA analysis, preprocessing techniques, and the YOLOv5-based detection framework. Section 4 presents and discusses the experimental results of image enhancement, segmentation, and pest detection. Finally, Section 6 concludes the paper and suggests directions for future research.

2. Related Work

2.1. Pest and Plant Disease Identification

There are many types of pests in nature. The most common pests are shown in Figure 1.

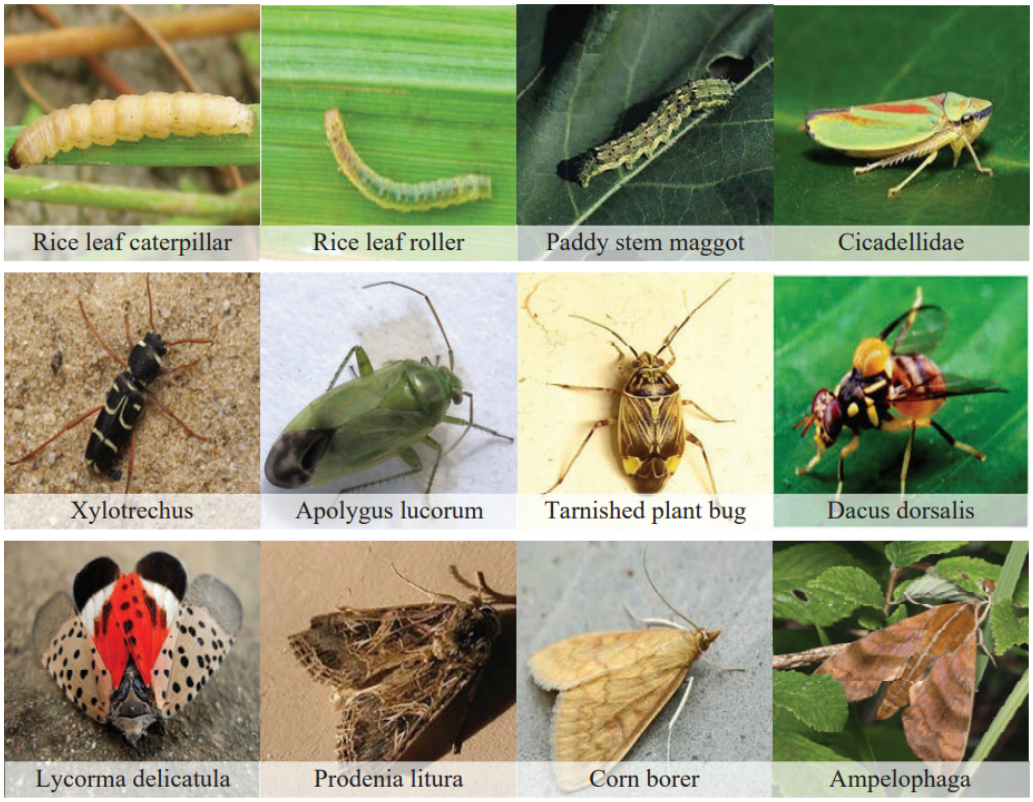


Figure 1. Example images from the IP102 benchmark dataset, showcasing the visual diversity across different insect pest species [23].

Huddar et al. [6] introduced a plant pest segmentation and identification algorithm based on the Relative Differential Intensity (RDI) technique for whitefly detection. Attaining an accuracy of 96%. Martin et al. [7] extended region-growing methods for pest identification and counting, employing threshold-based region splitting and merging. Espinoza et al. [8] proposed a novel method combining image processing and artificial neural networks to detect whiteflies and thrips in greenhouses. Detection, segmentation, morphological analysis, and colour estimation were executed through image-processing algorithms. Classification employed a multi-layer neural network, yielding high precision (0.96), recall (0.95), and F-measure (0.95) for whiteflies and thrips. Nesarajan et al. [9] utilised image processing and deep learning for coconut disease prediction via an Android app. SVM and CNN algorithms achieved pest identification accuracies of 93.54% and 93.72%, respectively. Eddeen et al. [10] developed a MATLAB-based application for identifying similar butterfly species with a 99% accuracy rate based on wing colour analysis. Chen et al. [11] proposed a deep convolutional neural network for tea tree pest identification, detecting 14 species with symmetry, as illustrated in Figure 2.

The convolutional neural network achieved a classification of 97.75%. Yao et al. [12] introduced a deep convolutional neural network using feature fusion for pest detection. Their approach, utilising Mask R-CNN and OSTU with an automatic threshold segmentation algorithm, demonstrated enhanced effectiveness in identifying pests in intricate environments.

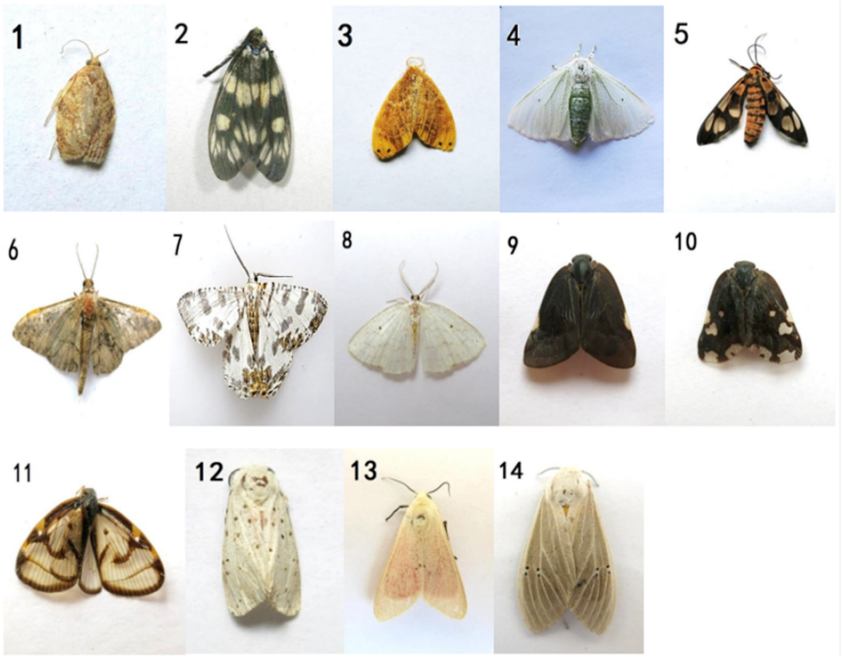


Figure 2. Representative images of 14 common tea tree pest species used for deep convolutional neural network-based identification [11]. The species are: (**1**) *Homona coffearia* (Meyrick), (**2**) *Eterusia aedea* (Linnaeus), (**3**) *Euproctis pseudoconspersa* (Strand), (**4**) *Arctornis alba* (Bremer), (**5**) *Amata germana* (Felder), (**6**) *Rikiosatoa vandervoordeni* (Prout), (**7**) *Culcula panterinaria* (Bremer et Gray), (**8**) *Scopula subpunctaria* (Herrich-Schaeffer), (**9**) *Ricania sublimbata* (Jacobi), (**10**) *Ricania speculum* (Walker), (**11**) *Euricania ocellus* (Walker), (**12**) *Spilosoma menthastri* (Esper), (**13**) *Spilarctia subcarnea* (Walker), and (**14**) *Ceratonoros transiens* (Walker).

Revathi et al. [13] introduced a uniform pixel counting method (HPCCDD) achieving 98.1% accuracy for detecting eight cotton leaf diseases. Li et al. [14] proposed a framework for tea pest and disease recognition. They employed Mask R-CNN for initial disease and insect spot segmentation in tea leaves, followed by a two-dimensional discrete wavelet transform to enhance feature extraction. This resulted in four frequency images, subsequently fed into a four-channel residual network (F-RNet) for tea pest and disease symptom identification. The paper showcased images of high-incidence tea leaf diseases and insect stress, namely brown blight (BB), target spot (TS), tea coal disease (TC), and Apolygus lucorum endanger leaves (AL), as depicted in Figure 3.

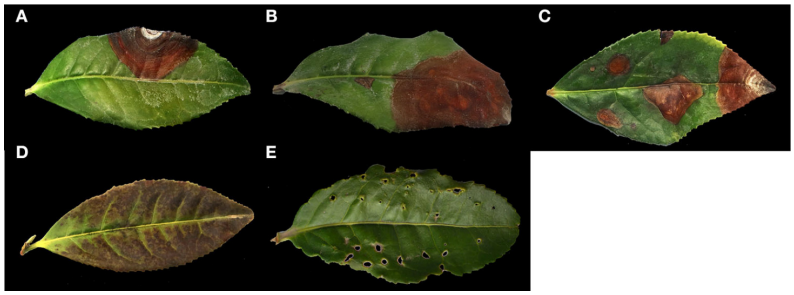


Figure 3. Original photographs of high-incidence tea leaf diseases and insect stress: (**A**) Brown Blight (BB), (**B**) Target Spot (TS), (**C**) Co-occurrence of BB and TS, (**D**) Tea Coal Disease (TC), and (**E**) *Apolygus lucorum* damage (AL) [14].

Among them, minimal distinction existed in phenotypic attributes between brown blight (BB) and target spot (TS), two co-occurring tea leaf diseases. Results demonstrated Mask R-CNN's 98.7% detection of disease and insect spots (DSIS), ensuring near-complete leaf extraction. The F-RNet model achieved 88% accuracy, surpassing other models

(e.g., SVM, AlexNet, VGG16, ResNet18). Abdu et al. [15] proposed the extended region of interest algorithm (EROI) for characterising leaf disease regions. The paper employs a colour thresholding-based segmentation algorithm tailored to foliar disease symptom attributes. Method efficacy was assessed via three deep learning pre-training models: AlexNet, ResNet, and VGG. Jiang et al. [16] enhanced the Visual Geometry Group Network-16 (VGG16) through multi-task-based learning. Pre-training on ImageNet was followed by migration and alternate learning for identifying three rice leaf diseases and two wheat leaf diseases. Rice and wheat disease sample images are depicted in Figure 4. Experimental findings indicated 97.22% accuracy for rice disease recognition and 98.75% for wheat disease recognition in the test set.

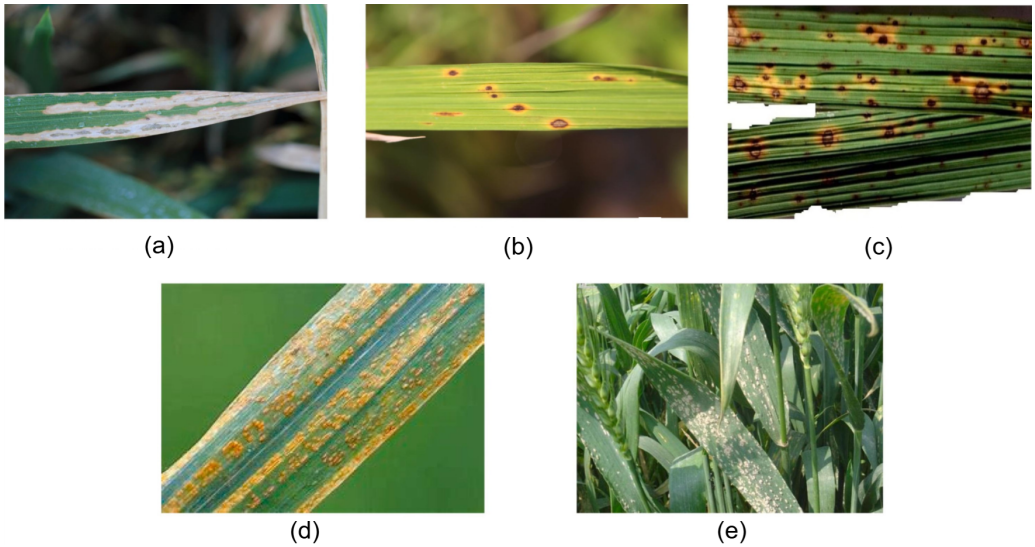


Figure 4. Sample images of rice and wheat leaf diseases: (**a**) Rice Bacterial Leaf Blight, (**b**) Rice Brown Spot, (**c**) Rice Leaf Smut, (**d**) Wheat Rust, and (**e**) Wheat Powdery Mildew [24].

Lu et al. [17] employed a deep CNN technique for rice disease identification, effectively classifying 10 common diseases. Jiang et al. [16] combined CNNs for feature extraction and SVMs for rice leaf disease classification, achieving 96.8% recognition accuracy. Fuentes et al. [18] proposed a real-time tomato pest and disease detector utilising deep learning, acquiring and processing data through a GPU-based software system. Zhu et al. [19] introduced an image analysis-based grape leaf disease detection method using wavelet filtering, Otsu segmentation, and Prewitt edge feature extraction. This method attained an average diagnosis rate of 91%. Zhao et al. [20] tackled leaf edge wormholes through an improved genetic wavelet neural network reconstruction algorithm, surpassing traditional models in performance based on validation tests.

2.2. Image Processing Techniques

Image processing techniques play a key role in pest detection and can improve the accuracy and efficiency of detection. This subsection will review several common image-processing methods used in pest detection, including image pre-processing, image segmentation, image edge detection, feature recognition, and neural network approaches.

2.2.1. Image Pre-Processing

Pest detection images are often disturbed by noise, which can reduce the accuracy of the detection. Therefore, image pre-processing, including image denoising and image enhancement, is first required. Traditional noise removal methods such as mean filtering, median filtering, and Gaussian filtering [25] are widely used for pest detection. These

methods can effectively reduce the noise in the image and enhance the quality of the image. In addition, noise removal methods based on wavelet transform have also been studied and applied, such as wavelet threshold denoising [26]. These methods can better handle different types of noise and improve the clarity and details of pest detection images. In recent years, deep learning-based image-denoising methods have made significant progress in pest detection. Using methods such as convolutional neural networks (CNNs) [27] and generative adversarial networks (GANs) [28], it is possible to learn complex image noise models and generate high-quality denoised images. These methods are able to adapt to different types of noise and reduce the impact of noise while maintaining pest characteristics.

2.2.2. Edge Detection

Edge detection is a key step in pest detection, as it can accurately extract the contour information of pest targets. Classical edge detection methods such as the Canny [29] edge detection algorithm, the Sobel operator [30], and the Laplacian operator [31] are widely used for pest detection. These methods are based on image gradients or second-order derivatives. Figure 5 shows an example of image edge detection with Sobel algorithm.

In recent years, significant breakthroughs have been made in edge detection methods based on deep learning. Edge detection methods using convolutional neural networks are able to learn richer feature representations and provide more accurate edge detection results. For example, methods such as Otsu thresholding segmentation [12] and CannyGAN [32] exploit the layered feature extraction capabilities of deep neural networks to achieve more refined and accurate edge detection.



Figure 5. Demonstration of the Sobel edge detection operator: (**Left**) original input image, (**Right**) resulting image after edge detection.

2.2.3. Feature Recognition

Feature extraction is pivotal in pest detection, enabling identification and localisation. Convolutional neural networks (CNNs) [33] are widely used for feature recognition, capturing local and global image features via convolution and convergence operations.

Deconvolutional neural networks also contribute to localising pests by gradually restoring image features to their original size, enhancing detailed restoration from abstract feature spaces.

In summary, image processing in pest detection covers denoising, enhancement, segmentation, edge detection, and feature extraction. While traditional methods offer simplicity, deep learning excels in learning complex patterns and semantics, enhancing accuracy. As technology advances, image processing's role in pest detection and agricultural support remains crucial.

2.2.4. Image Segmentation Techniques

Image segmentation is pivotal in pest detection, isolating pests from the background and enhancing accuracy. Methods encompass threshold [34], region-based, and edge-based segmentation [35]. Thresholding uses grey-scale values to classify pixels against a preset threshold, suitable for distinct pest-background contrasts. Region-based segmentation partitions images into contiguous regions based on pixel similarity, adept at intricate backgrounds. Edge-based methods locate pests by identifying image edges. See Figure 6 for a binarisation-based segmentation example.



Figure 6. Demonstration of image binarisation for segmentation: (**Left**) original input image, (**Right**) binary image resulting from threshold-based segmentation.

Recent years have witnessed image segmentation strides in deep learning. Fully convolutional networks (FCNs) [36] transform pixel-level tasks into classification. Semantic networks like U-Net [37] and Mask R-CNN [38] excel in pest detection, leveraging deep neural networks for precise semantic capture and fine segmentation.

2.3. Image Quality Assessment

Image Quality Assessment (IQA) plays a key role in plant pest detection, helping to determine the clarity and image quality of the original dataset and thus providing a reliable image base. There are three types of IQA: Full-Reference IQA, Reduced-Reference IQA and No-Reference IQA.

2.3.1. Full-Reference IQA

In Full-Reference IQA, the image being evaluated is compared to a reference image, which is usually the original distortion-free (or ideal) image. The difference between the evaluated image and the reference image is thus measured, and the quality of the image is determined. It is measured based on various image quality assessment metrics. Common metrics include peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), multi-scale structural similarity index (MS-SSIM) [39], etc.

2.3.2. Reduced-Reference IQA

Reduced-Reference IQA compares the evaluated image to partial reference data, typically extracted features or image content such as structure, texture, and colour statistics. This method assesses image quality to some extent, offering advantages in requiring less reference data compared to Full-Reference IQA. This enhances practicality, especially when full reference images are unavailable, boosting algorithm efficiency and reducing storage and transmission demands. Common quasi-reference metrics include Reduced SSIM [40], Reduced FSIM, etc.

Nonetheless, quasi-reference methods may trade some accuracy for reduced reference data. Limited reference may hinder full consideration of image features and details. Thus,

when selecting an assessment method, application contexts, resource constraints, and accuracy should be weighed.

2.3.3. No-Reference IQA

No-Reference IQA solely employs intrinsic features of the evaluated image for quality assessment, obviating reliance on reference data. Internal attributes encompass sharpness, noise level, contrast, colour distribution, etc.

The method's advantage lies in its independence from reference images, rendering it suitable when references are absent or implementing full/quasi-reference approaches is impractical. No-Reference IQA autonomously gauges image quality, dispensing with comparisons against standards.

Common techniques encompass statistical, machine learning, and deep learning-based methods [3], such as BRISQUE, Perceptual Image Quality Evaluator (PIQE), and Naturalness Image Quality Evaluator (NIQE). These methods employ diverse feature extraction, analysis, and modelling to infer image quality [2].

- BRISQUE (Blind/Referenceless Image Spatial Quality Evaluator): BRISQUE assesses image quality without a reference image, deriving a quality score based on statistical features like brightness, contrast, and texture, relative to a training dataset. It's employed for evaluating image compression and processing algorithms, gauging their impact on quality.
- PIQE (Perceptual Image Quality Evaluator): PIQE gauges perceptual image quality, comparing quality to human perception. It derives a quality score through analysis of structure, contrast, and colour. PIQE finds use in image processing and transmission to assess algorithm and system effects on quality.
- NIQE (Naturalness Image Quality Evaluator): NIQE gauges image naturalness and realism, deriving a quality score by analysing statistical features like texture, contrast, and edges. NIQE is prevalent in evaluating image enhancement and restoration algorithms, discerning their effects on realism.

3. Methodology

The methodology is a sequential pipeline that starts with the collection of a pest image dataset. Initially, an Image Quality Assessment (IQA) is conducted to evaluate the dataset. These IQA scores directly inform the next stage, where images are preprocessed (through denoising and deblurring with optimal filters) and the dataset is augmented via operations like rotation and translation. The resulting enhanced images then undergo segmentation and edge detection to highlight pest contours. This prepared data is finally input into computer vision models for automated classification and detection, with the outputs being exported for comprehensive analysis.

3.1. Image Classification and Acquisition

Pests are diverse in nature and widely distributed in different environments, with significant impacts on ecosystems and human activities. In this paper, we have studied four common and economically significant pests selected from the IP102 dataset. The selection was based on their prevalence and the distinct morphological challenges they present for detection. The pests are as follows:

- Rice Leaf Roller (*Cnaphalocrocis medinalis*)
- Grub (Larvae of beetles in the family Scarabaeidae, e.g., *Holotrichia parallela*)
- Peach Borer (*Conogethes punctiferalis*)
- Muscomorpha (A subsection of the order Diptera; for this study, we focused on representative fruit fly species, e.g., *Bactrocera dorsalis*)

Examples of these four categories are shown in Figure 7.



(**a**) Rice Leaf Roller



(**b**) Grub



(**c**) Peach Borer



(**d**) Muscomorpha

Figure 7. Example images of the four pest categories studied in this work: (**a**) Rice Leaf Roller (*Chaphalocrocis medianalis*), (**b**) Grub (*Scarabaeidae larvae*), (**c**) Peach Borer (*Conogethes punctiferalis*), and (**d**) Muscomorpha (e.g., fruit fly *Bactrocera dorsalis*).

Firstly, Grub is a type of larva, usually referring to the larval stage of beetles in the insect family Scarabaeidae. They primarily feed on plant roots, causing damage to the root system, which leads to the hindered absorption of nutrients by plants and ultimately stunts plant growth. Secondly, Peach Borer is an insect that primarily infests peach trees. They feed on the trunks and branches of peach trees, causing tunnelling damage and severely affecting the growth and fruit yield of the peach trees. Next, Muscomorpha is a group of dipteran insects, including certain agricultural pests like fruit flies and leaf flies. These insects often feed on fruits, vegetables, and other plants, causing damage through oviposition and larval feeding, resulting in crop losses and reduced yields. Lastly, the Rice Leaf Roller is a pest that poses a serious threat to rice crops. They feed on rice leaves, causing leaf rolling and affecting photosynthesis and nutrient uptake, ultimately leading to a decrease in rice yield. These pests pose a significant threat to the reproduction and growth of crops, necessitating the implementation of effective control measures to mitigate their impact.

The data used in this paper is from the publicly available Pest Identification Big Data set IP102 [23]. IP102 contains approximately 75,000 images across 102 pest categories and is the largest dataset of its kind for insect pest recognition. However, the IP102 dataset has problems such as duplicate images, mismatched pest categories, and low resolution,

which require manual screening and cleaning of the images. In this study, 100 images were selected as the training set for the dataset for the four pest species to be studied. And 20 images were selected as the test set. There are a total of 120 images.

3.2. IQA Analysis

During the image preparation stage, one crucial aspect involves quantitatively evaluating the quality of the images. This step is essential to establish standardised classification criteria, thereby eliminating reliance on the subjective judgments of the operators. To address the challenge of quantifying visual distortions in the absence of reference image data, we employ the Non-Reference Image Quality Assessment (NR-IQA) technique.

In this study, we utilise MATLAB (Version R2023a) to obtain the BRISQUE, NIQE, and PIQE values for each image. These three methods provide non-reference image quality scores, eliminating the need for an ideal reference image for comparison. They offer advantages such as computational efficiency, suitability for real-time applications, and high predictive accuracy. These methods consider factors such as sharpness, texture, noise, and contrast to evaluate the image quality and generate a comprehensive score representing the overall quality. It is important to note that the scoring ranges slightly differ among these methods, with NIQE scores ranging from 0 to 10, while both BRISQUE and PIQE scores range from 0 to 100. Lower scores indicate better image quality. Therefore, we anticipate that images of pests with higher clarity and less blurriness will yield lower scores. Figure 8 shows a comparison of the BRISQUE values for two images of pests of different quality.



(**a**) (**b**)

Figure 8. Visual comparison of two pest images with corresponding Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) scores, illustrating the metric's correlation with perceived image quality (lower score indicates higher quality). (**a**) BRISQUE = 14.75. (**b**) BRISQUE = 45.24.

This image quality assessment process establishes a reliable foundation for subsequent pest feature extraction and analysis.

3.3. Pest Image Pre-Processing

In the field of image processing and computer vision, image preprocessing plays a crucial role. In real-world scenarios, pest images are often affected by noise and blurring, which degrade the quality and usability of the images, posing challenges for subsequent image analysis and processing tasks. Therefore, applying appropriate preprocessing techniques can enhance image quality and provide a more reliable foundation for subsequent image classification and analysis tasks.

Image preprocessing involves several key steps, including enhancing image quality, noise removal, and extracting relevant information. In this section, we explain the concepts and principles involved in the image preprocessing workflow, starting from $f(x, y)$ to $h(x, y)$ to the addition of noise $n(x, y)$, resulting in the transformed image $g(x, y)$. We also

delve into frequency domain analysis, where $g(x,y)$ is transformed into $G(u,v)$ and $F(u,v)$. Finally, we discuss the Inverse Fourier Transform (IFT) operation on $F(u,v)$ to obtain the processed image $\tilde{f}(x,y)$. The workflow diagram is presented in Figure 9.

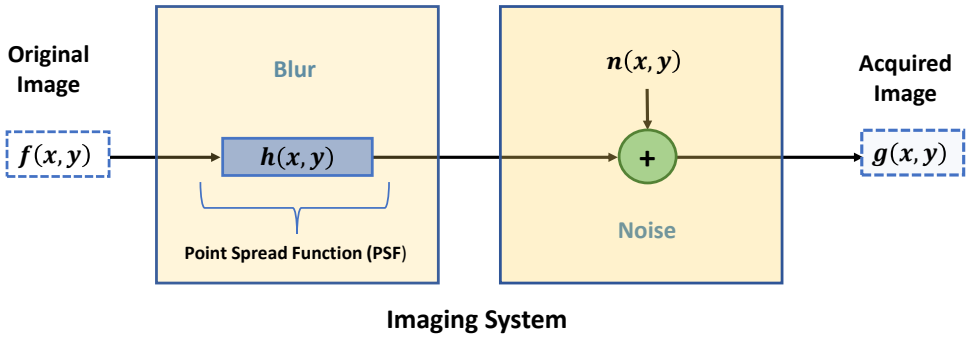


Figure 9. Block diagram of the generalised imaging system model, depicting the transformation from the original image to the acquired image via blurring with a Point Spread Function (PSF) and the addition of noise.

The pre-processing steps outlined above are aimed at improving image quality, reducing noise, and enhancing the relevant features of the pest images. These steps are essential for preparing the images before further analysis, such as segmentation, classification, or object detection. By effectively handling noise and blurring issues, the pre-processing techniques contribute to more accurate and reliable results in subsequent image-processing tasks.

3.4. Image Pre-Processing Fundamentals

This section explains the meaning of each function in the flowchart and the workflow. $f(x,y)$: The original input image, where (x,y) represents the spatial domain coordinates. This image may contain objects or features of interest. $h(x,y)$: The blur kernel or point spread function (PSF). The blurring operation simulates the effect of image blur caused by factors such as camera optics or environmental conditions. $h(x,y)$ describes the degree and nature of image blurring. Adding noise $n(x,y)$: $n(x,y)$ represents the noise present in the image. Images can be affected by adverse noise caused by sensors, electronic devices, or environmental factors. Noise can have a detrimental impact on image quality and feature extraction. $g(x,y)$: The image obtained after the blurring and noise addition processes. According to the image restoration model, $g(x,y)$ can be expressed as the convolution of $f(x,y)$ and $h(x,y)$, plus the noise component $n(x,y)$. The mathematical expression for this process is:

$$g(x,y) = f(x,y) * h(x,y) + n(x,y). \tag{1}$$

The symbol $*$ denotes 2D convolution. Next, we present the frequency domain analysis.

$G(u,v)$: $G(u,v)$ represents the two-dimensional Fourier transform (DFT) of $g(x,y)$, where (u,v) represents the frequency domain coordinates. By applying the Fourier transform to $g(x,y)$, we can convert the image from the spatial domain to the frequency domain for further analysis of its frequency domain characteristics. The mathematical expression for this step is:

$$G(u,v) = F(u,v)H(u,v) + N(u,v), \tag{2}$$

where $F(u,v)$ and $H(u,v)$ represent the frequency domain representations of $f(x,y)$ and $h(x,y)$, respectively. $N(u,v)$ represents the frequency domain representation of the noise component.

$F(u,v)$: $F(u,v)$ represents the frequency domain image obtained after frequency domain filtering and processing. Based on the principles of frequency domain filtering,

various operations can be applied to $G(u,v)$, such as filtering, correction, and enhancement, to obtain the frequency domain representation of the processed image.

Finally, the inverse Fourier transform (IFT) operation is performed. $\tilde{f}(x,y)$: $\tilde{f}(x,y)$ represents the processed image obtained by applying the inverse Fourier transform to $F(u,v)$. The inverse Fourier transform converts the image from the frequency domain back to the spatial domain, providing the spatial domain representation of the processed image. The mathematical expression for this step is:

$$\tilde{f}(x,y) = IFFT\{F(u,v)\}. \tag{3}$$

The comprehensive image preprocessing workflow outlined above encompasses fundamental knowledge and techniques in the field of image preprocessing. These methods and principles play a crucial role in improving image quality, noise reduction, and information recovery, laying a solid foundation for subsequent image analysis and processing tasks.

3.5. Image Pre-Processing Filter

The following is a list of the most common pre-processing methods that the project used for image enhancement. Each filter has its corresponding parameter setting, which has a big influence on the outcome after the pre-processing.

I. Wiener Filter

Wiener filtering is a linear filter based on the least squares criterion of optimality. The Wiener filter aims to recover the image degradation due to noise and to restore the original image as much as possible.

The core idea of the Wiener filter is to minimise the mean square error between the original image and the degradation model. The degradation model describes how an image can be superimposed with blur and noise by some degradation process, for example, the blur operation can be represented by convolution, while the noise can be modelled by adding random interference. The goal of Wiener filtering is to estimate the original image by inverse operations.

$$\hat{f}(u,v) = \frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_{n(u,v)}}{S_{f(u,v)}} \cdot F(u,v)}, \tag{4}$$

where $\hat{f}(u,v)$ is the frequency domain representation of the recovered image, $H(u,v)$ is the frequency response of the degraded function, $F(u,v)$ is the frequency domain representation of the degraded image, $S_n(u,v)$ is the noise power spectrum and $S_f(u,v)$ is the power spectrum of the original image. The steps of Wiener filtering are as follows:

1. Transform the original image and the degenerate model into the frequency domain.
2. Calculate the frequency response $H(u,v)$ of the degenerate function.
3. Estimate the noise power spectrum $S_n(u,v)$.
4. Estimate the power spectrum of the original image $S_f(u,v)$.
5. Filter the frequency domain representation $F(u,v)$ of the image using the Wiener filter formula to obtain the frequency domain representation of the recovered image.
6. Transform the frequency domain representation of the recovered image back to the spatial domain to obtain the final recovered image.

Wiener filtering can be effective in reducing the effects of degradation and noise on the image, but in practice, it is accurate modelling of the noise and degradation process is key,

as there may be errors in the estimation of these parameters that may lead to unsatisfactory filtering results.

II. Lucy–Richardson Filter

The Lucy–Richardson algorithm is an iterative blind deconvolution algorithm for image recovery and super-resolution reconstruction. It was originally proposed by W. H. Richardson in 1972 [41] and L. F. Lucy in 1974 [42] for the case where the degradation model is known but the point spread function (PSF) of the degradation process is unknown or incompletely known. The principle of the Lucy–Richardson algorithm is also based on the idea of the minimum mean square error criterion and maximum likelihood estimation, and aims to recover the original image by iterative inversion. The basic principles are as follows:

1. Initialisation: First, the degraded image is used as the recovered image for the initial estimation.
2. Iterative update: The recovery image is updated iteratively by alternating the steps of forward and inverse models until a predetermined number of iterations or convergence conditions are reached.
    (a) Forward model (predicted image): The current recovery image is convolved with the point spread function to obtain the predicted image.
    (b) Reverse model (updated estimation): The ratio of the degraded image to the predicted image is multiplied by the inverse convolution of the point spread function to obtain the updated estimated image.

Step 2 is repeated until a predetermined number of iterations is reached or the convergence condition is satisfied. The key idea of the algorithm is to progressively optimise the estimated image to approximate the original image by iteratively updating it and using the information about the difference between the degraded image and the predicted image. The RL algorithm is sensitive to noise, so in practice, it is often necessary to regularise or add a priori constraints during the iterative process to balance the effects of deconvolution and denoising and to avoid over-fitting the noise in the degraded image.

III. Sharpening Filter

A sharpening filter enhances image edges and fine details by emphasising high-frequency components. It applies a specific filter, like Laplacian or unsharp mask, to amplify intensity differences between neighbouring pixels. This process improves visual clarity and sharpness by making edges more pronounced. However, careful parameter adjustment is crucial to avoid artefacts and noise amplification.

IV. DampedLS Filter

The DampedLS (Damped Least Squares) filter is a filtering technique used in image processing to enhance image quality and reduce noise. It employs a mathematical optimisation algorithm that minimises the difference between the filtered image and the original image while taking into account the noise characteristics. The DampedLS filter effectively balances the trade-off between noise reduction and preservation of image details. By applying a damping factor, it suppresses the noise while preserving important image features. This filter is particularly useful in applications where noise reduction is essential without sacrificing important image information.

V. Tikhonov Filter

The Tikhonov filter, also known as the regularised least squares filter, is a popular technique used in image processing for denoising and image restoration. It is based on Tikhonov regularisation, which introduces a regularisation term into the least squares

problem to improve the stability and robustness of the solution. The Tikhonov filter strikes a balance between noise reduction and preserving image details by imposing a penalty on the solution that encourages smoothness. This regularisation term helps to suppress noise while preserving important structural information in the image.

VI. Total Variation Filter

The Total Variation (TV) filter is a powerful technique used in image processing for denoising and image restoration. It addresses the issue of preserving sharp edges and fine details while reducing noise. The TV filter exploits the concept of total variation, which measures the amount of variation or change in intensity across neighbouring pixels in an image. By minimising the total variation in an image while preserving important features, the TV filter effectively removes noise while preserving edges and details.

3.6. Pest Image Processing

In this chapter, we first try to identify and extract pest surface features from traditional visual inspection methods like Sobel, Canny, etc. Then, build a YOLO deep learning algorithm framework, construct a pest feature dataset, train the model, and finally test the training results to verify its efficiency and accuracy.

3.6.1. Traditional Extraction Algorithm for Pest Species Classification Features

Pests exhibit distinctive morphological and textural traits. Edge detection captures these features like segments, tentacles, and spots, aiding in feature extraction and identification. Widely used methods encompass the Roberts, Sobel, Prewitt, and Canny operators. The Sobel and Prewitt operators, as first-order differentials, identify edges where grayscale differences between neighbouring pixels are maximised. The Canny approach employs first-order derivatives to detect step-like edges effectively, yielding accurate edge localisation and image edge detection.

3.6.2. Pest Location Identification Algorithm Based on YOLOv5

Deep learning has emerged as a significant approach for image detection, including the automatic identification of pest locations. Accordingly, for this project, a deep learning-based target detection algorithm is chosen.

While newer versions of YOLO (e.g., YOLOv7, v8) exist, YOLOv5 was selected for this study due to its proven balance between speed, accuracy, and architectural maturity, which is well-suited for a methodological investigation focused on the front-end image processing pipeline. Its extensive documentation, robust community support, and ease of customisation for our specific pest dataset facilitated a more focused evaluation of the impact of IQA and preprocessing. Furthermore, as our primary contribution lies in the integrated workflow rather than benchmarking the absolute latest detector, YOLOv5 provides a strong and reliable baseline model. Future work will involve integrating and comparing these newer architectures.

The core issue addressed by this algorithm involves determining the position of the Bounding Box (BBox), typically specified by the coordinates of the upper-left corner and the box's length and width. Bounding Boxes enclose pests during dataset labelling. Current deep learning-based target detection methods generally fall into two categories: region-proposal approaches like R-CNN, Fast R-CNN, and Faster R-CNN, and non-regional-proposal approaches like YOLO and SSD. Region-proposal methods nominate regions of interest (ROIs) for classification. In contrast, non-regional-proposal methods directly perform tasks like BBox regression and object classification.

Additionally, the Intersection over Union (IOU) concept is frequently employed in target detection to quantify overlap between two BBoxes. This ratio, calculated as the

intersection area divided by the union area of BBoxes, indicates the overlap extent. Larger values signify higher overlap and more precise detection, as depicted in Figure 10.

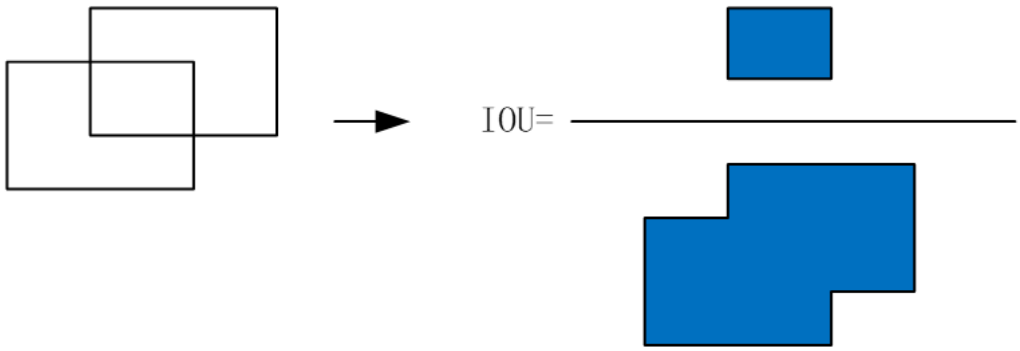


Figure 10. Schematic illustrating the Intersection over Union (IOU) calculation, used to evaluate the overlap between a predicted bounding box and the ground truth box.

1. YOLOv5 NET:
   YOLOv5, short for "You Only Look Once version 5," is an advanced single-stage object detection model introduced by Ultralytics in 2020. It represents a culmination of state-of-the-art techniques for accurate and efficient object detection in images. The architecture is built upon the classic CSPDarknet53 backbone, incorporating specialised modules like the focus module and spatial pyramid pooling (SPP) module. Additionally, the model employs PANet as a neck network and the traditional YOLO detection head for predictions.
   What sets YOLOv5 apart is its adaptability through varying model sizes: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, each increasing in depth and width. This enables tailored solutions to different computational and accuracy requirements. Notably, the YOLOv5s model is particularly suitable for real-time applications due to its reduced parameters and fast inference speed.
   The YOLOv5 architecture leverages deep learning techniques to directly predict object classes and bounding box coordinates in a single forward pass. This streamlined approach ensures rapid processing and accurate detection, making it a preferred choice for numerous computer vision applications, including robotics, surveillance, and food monitoring. The YOLOv5 model is shown in Figure 11.
   In the above diagram, BackBone is the backbone network, mainly responsible for abstracting the input images into features. The Neck layer is used between the BackBone and the final output layer to fully extract the network features and prepare the network for the output. The YOLOv5 Neck network uses the CSP2 structure, which is based on the Cross Stage Partial Network (CSPnet) design, to enhance the fusion of features. Prediction layer: This layer takes the data features extracted from the network and finally transforms them into the data format we need. SPP: a multi-scale fusion using 1×1, 5×5, 9×9 and 13×13 maximum pooling. The YOLOv5 algorithm uses a K-Means-based clustering operation to cluster the feature frame data from the pest feature dataset into anchor frames of a specific size.

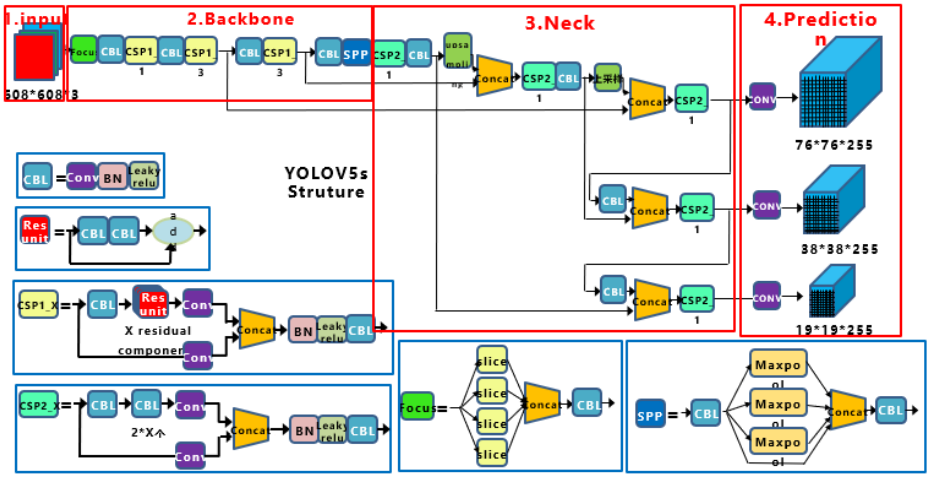


Figure 11. Architectural overview of the YOLOv5 object detection model, comprising the CSP-Darknet53 Backbone, the Path Aggregation Network (PANet) Neck, and the YOLO detection Head, including the Spatial Pyramid Pooling (SPP) module.

2. Statistical anchor based on K-means algorithm:

The K-Means algorithm is an unsupervised clustering algorithm, and this paper uses K-Means to cluster the anchor frames of pest species. Firstly, the number k of anchor frames to be generated is defined, then k sample pest frames are randomly selected as the initial anchor frames, the similarity between each sample frame and the anchor frame is calculated, the magnitude of similarity classifies all defective frames, and the class mean is used as the new anchor frame. In the K-Means algorithm, it is assumed that there are m samples in the training data set $X_1, X_2, \ldots, X_m$, where each sample X_i is an n-dimensional vector. At this point, the samples can be represented as an $m \times n$ matrix:

$$X_{m\times n} = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}_{m\times n}. \tag{5}$$

Assume that there are k classes of $C_1, C_2, \ldots, C_k$. In the K-Means algorithm, the Euclidean distance is used to calculate the similarity between each defective frame and the anchor frame and to classify that defective frame sample into the most similar class. The number of new clustering centres k is obtained by recalculating the divided data set. The process is repeated until the centre of mass no longer changes. The goal of the K-Means algorithm is to divide each sample X_i into the most similar categories, and to recalculate the cluster centres C_k using the samples in each category:

$$C_K{}' = \frac{\sum_{x^{(i)}\in C_k} X^{(i)}}{(X^i \epsilon C_k)}, \tag{6}$$

where molecule represents the sum of the eigenvectors of all samples in category C_k and the denominator represents the number of samples in category C_k. The K-Means algorithm is calculated when the samples are classified by similarity, when the category is no longer changed, and the anchor frames are no longer updated, indicating that all defect frames of the compound skin part surface defect dataset have been clustered into representative anchor frames. In contrast, YOLOv5 updates the width and height data of the anchor frames in real-time during the training process,

performing a reverse operation to update the network parameters and anchor frame data based on the difference between each predicted defect frame and the anchor frame. In this way, the anchor box data is adaptively adjusted according to the results of each training session, and its accuracy is higher.

3. Data set creation:

(a) Data Enhancement:
Data augmentation was applied exclusively to the training set to prevent data leakage and ensure a fair evaluation on the original test set. In general, for deep learning models to be applied to practical agricultural pest detection, many images of pest samples need to be obtained. And since the amount of data currently acquired is limited. Therefore, the pest dataset needs to be augmented with data augmentation. The enhancement of our pest dataset can do two tasks: (1) increase the number of pest samples to improve the accuracy of the model detection, and (2) ensure the stability of the model detection will not be reduced when there is noise interference at the detection site. The following data augmentation techniques were used to improve model robustness and simulate real-world conditions:

i. Geometric Transformations: Mirroring, rotation ($\pm 30°$), and panning (up to 20% of image width/height) were used to add viewpoint invariance. The effect of mirroring and rotating 30 degrees counter-clockwise is shown in Figure 12.
To improve the neural network's ability to locate defects anywhere in the image, we employ a translation-based data augmentation. The target defect is shifted along the X and/or Y axes over a black background, as shown in Figure 13. This technique, proven effective in prior work, trains the network to recognise targets in all image regions.

ii. Photometric Transformations: Brightness ($\pm 20\%$) and contrast ($\pm 30\%$) adjustments were applied to simulate varying lighting conditions.

iii. Occlusion and Scaling: Random occlusion (cutout of up to 10% of the image area) was used to simulate obstructions by leaves or debris. Scale variation (zooming between $0.8\times$ and $1.2\times$) was incorporated to account for different camera-to-subject distances.

iv. Noise Addition: Salt-and-pepper and Gaussian noise were added to improve model resilience to sensor noise and poor acquisition conditions. One noise pattern is pretzel noise, where there are a random number of black and white noise points of random size in the image, and another is Gaussian noise, which is usually sensor noise caused by poor lighting conditions or high temperatures. The effect of adding noise is shown in Figure 14.

(b) Data labelling:
The defective dataset was annotated using the deep learning dataset annotation tool LabelImg, as shown in Figure 15.
There are two forms of target annotation for the dataset: the first is to use a rectangular box to completely encompass the target to be identified; the accuracy of the rectangular box determines how well the deep learning model learns. The second is to depict the outline details of the detected target, generating a mask image that represents the shape of the pest. As shown above, this topic uses rectangular boxes to frame the defect and annotate the defect type, the pest class, rectangular box coordinates, rectangular box length and width data, and the corresponding image to save to a file. Finally, the annotated dataset

is disrupted and fed into the model in a random order for learning, and the dataset is divided into a training set and a test set in a ratio of 5:1.



Figure 12. Example of data augmentation through geometric transformations: (**Left**) horizontally mirrored image, (**Right**) image rotated 30 degrees counter-clockwise.



Figure 13. Example of translation-based data augmentation, where the target pest is shifted within the image frame over a black background to improve model robustness to object location.



Figure 14. Examples of images with synthetic noise added to improve model resilience: (**Left**) Salt-and-pepper noise, (**Right**) Gaussian noise.

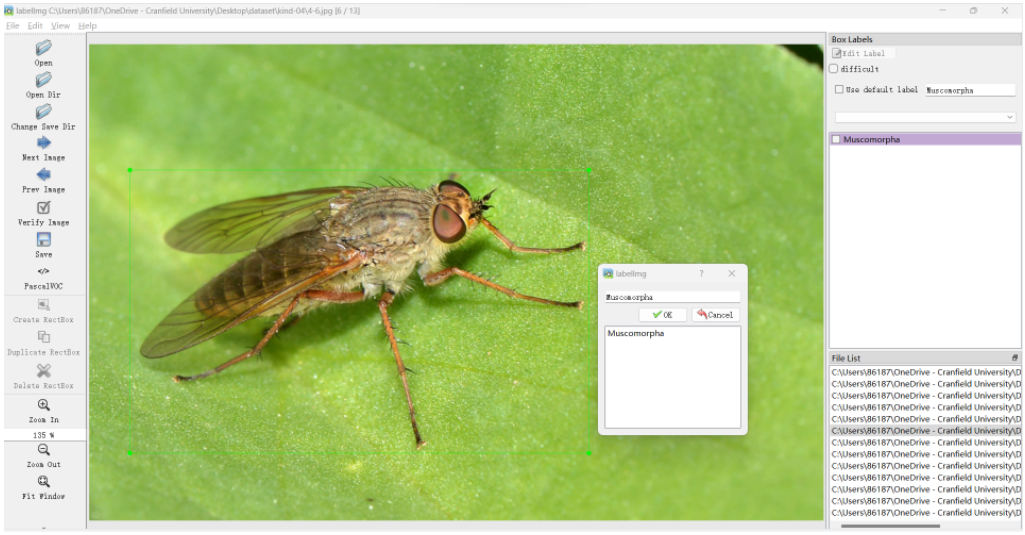


Figure 15. Screenshot of the Labelling tool interface used for annotating pest bounding boxes in the dataset.

3.7. Model Training Configuration

The YOLOv5 model was trained using the PyTorch framework (v1.12.0) and the Ultralytics YOLOv5 repository. Training was conducted on a computing environment with an NVIDIA Tesla V100 GPU and an Intel Xeon CPU. The model was trained for 300 epochs with a batch size of 16. The Stochastic Gradient Descent (SGD) optimiser was employed with an initial learning rate of 0.01, a momentum of 0.937, and a weight decay (L2 regularisation) factor of 0.0005. A cosine annealing learning rate scheduler was used to adjust the learning rate during training. To prevent overfitting, early stopping was implemented with a patience of 50 epochs, monitoring the mean Average Precision (mAP@0.5) on the validation set. The dataset was split into training and test sets in a 5:1 ratio, and all augmentations were applied exclusively to the training set.

4. Experimental Results and Discussion

The results after applying the presented methodology to the Pest images are as follows and can be divided into three subsections: image enhancement, image segmentation, and object detection. This section includes an assessment of the results' accuracy as well as an analysis of the outcome values.

4.1. Image Enhancement

Some image quality problems identified in the initial dataset were: blurriness and noise. 30 images were selected randomly from the original dataset.

BRISQUE, PIQE, and NIQE values can all help evaluate and quantify the initial image quality. Figure 16 below shows the three kinds of values of these 30 original images.

In the case of BRISQUE, for example, images below 30 are already high quality in their own right. So, 12 images with BRISQUE values higher than 30 were selected for image enhancement with 6 different filters. The partial results of the six filter applications are shown in Tables 1–3.

It is worth mentioning that when applying the filter to each image, the corresponding parameters need to be adjusted. Not every image can share the same parameters of the same filter. Different filters produce different results for different images. Similarly, the same filter with different parameters will produce very different results. Therefore, after many experiments, we have developed the filter and parameter settings suitable for each image. The BRISQUE values before and after processing are shown in Table 4.

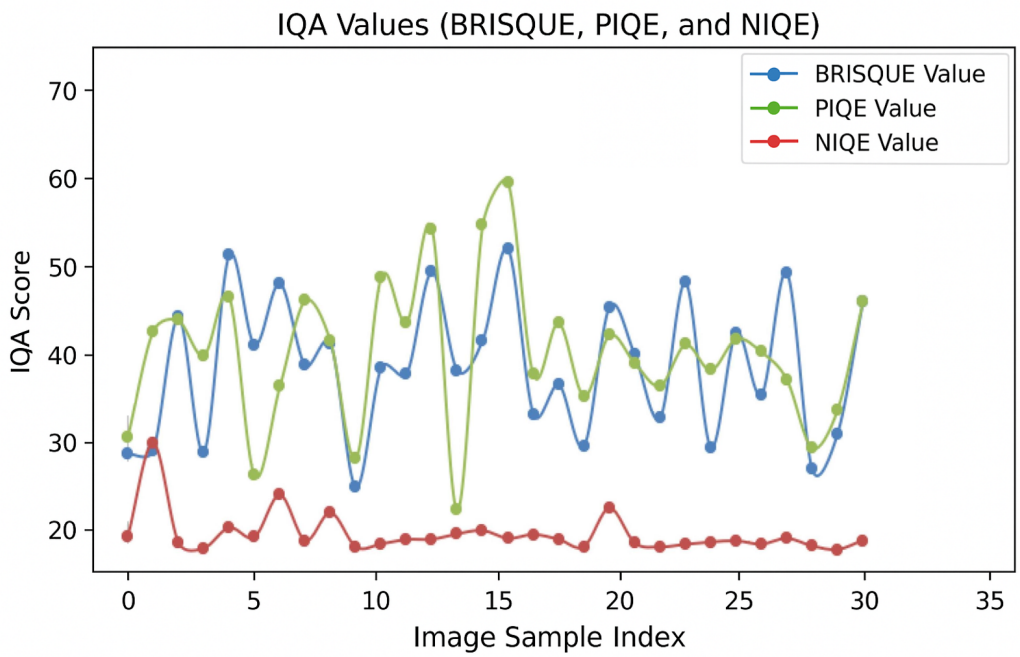


Figure 16. IQA Values (BRISQUE, PIQE, and NIQE) for 30 randomly selected original images.

Table 1. Image enhancement results comparing the original images with outputs from the Total Variation and Wiener filters, alongside their corresponding BRISQUE scores.

Image	Original Image	Total Variation	Wiener
0-11			
	BRISQUE: 38.26	BRISQUE: 30.21	BRISQUE: 19.58
0-13			
	BRISQUE: 51.14	BRISQUE: 22.24	BRISQUE: 25.42
0-14			
	BRISQUE: 36.84	BRISQUE: 23.9	BRISQUE: 11.18
0-15			
	BRISQUE:43.46	BRISQUE:43.45	BRISQUE:43.42

Table 2. Image enhancement results comparing the original images with outputs from the Richardson-Lucy and Damped Least Squares (DampedLS) filters, alongside their corresponding BRISQUE scores.

Image	Original Image	Richardson-Lucy	DampedLS
0-3			
	BRISQUE: 45.24	BRISQUE: 11.39	BRISQUE: 33.78
1-1			
	BRISQUE: 32.45	BRISQUE: 15.29	BRISQUE: 11.45
1-9			
	BRISQUE: 35.81	BRISQUE: 18.93	BRISQUE: 17.79
4-10			
	BRISQUE: 45.08	BRISQUE: 38.66	BRISQUE: 39.89

Table 3. Image enhancement results comparing the original images with outputs from the Tikhonov and Sharpening filters, alongside their corresponding BRISQUE scores.

Image	Original Image	Tikhonov	Sharpen
1-1			
	BRISQUE: 32.45	BRISQUE: 10.87	BRISQUE: 16.95
4-12			
	BRISQUE: 35.65	BRISQUE: 33.99	BRISQUE: 26.77

Table 3. *Cont.*

Image	Original Image	Tikhonov	Sharpen
4-13			
	BRISQUE: 40.56	BRISQUE: 26.02	BRISQUE: 37.8
leaf04			
	BRISQUE: 44.11	BRISQUE: 18.85	BRISQUE: 31.4

Table 4. Comprehensive comparison of BRISQUE values for all 12 test images before (Original) and after processing with six different filters (Wiener, Lucy–Richardson (RL), Damped-LS, Tikhonov, Sharpen, Total Variation (TV)).

Image	Original	Wiener	RL	Damped-LS	Tikhonov	Sharpen	TV
0-11	38.26	19.58	29.33	19.35	27.74	29.46	30.21
0-13	51.14	25.42	26.85	26.26	24.24	37.1	22.24
0-14	36.84	11.18	13.24	18.27	13.44	19.07	23.9
0-15	43.46	43.42	43.43	43.44	43.46	43.46	43.45
0-17	34.22	43.7	37.37	43.31	34.67	34.14	31.14
0-3	45.24	32.07	11.39	33.78	24.16	42.7	32.68
1-1	32.45	11.81	15.29	11.45	10.87	16.95	23.33
1-9	35.81	18.89	18.93	17.79	20.9	22.41	17.72
4-10	45.08	37.84	38.66	39.89	36.69	45	32.63
4-12	35.65	28.16	33.49	29.4	33.99	26.77	20.32
4-13	40.56	26.35	21.76	24.03	26.02	37.8	26.42
leaf.04	44.11	28.47	26.52	30.39	18.85	31.4	27.81

Finally, based on the BRISQUE value, a filter and corresponding parameters are selected for each image that ultimately suit them. So that they have the lowest BRISQUE value, i.e., the highest image quality. In order to make the results more obvious, the filters and parameters with the lowest BRISQUE values after processing were selected for each image, and a line graph comparing them with the original data is shown in Figure 17.

To demonstrate the validity of the results, image quality was again assessed using PIQE VALUE. Again, the lower the value, the better the image quality. The results of the evaluation are shown in Figure 18.

Since BRISQUE mainly focuses on analysing the spatial domain information of an image, including the sharpness, contrast, and block variation. It is an evaluation metric based on statistical properties. PIQE mainly focuses on the image's perceptual quality and assesses the image's overall visual quality by analysing the image's colour, texture, distortion, etc. It is an evaluation metric based on perceptual properties. So the evaluation

results of PIQE are slightly different from BRISQUE. It is found by calculating its mean value. The mean value of PIQE for raw image data quality is 34.94, and the mean value of the process is 30.38. This means that the image enhancement approach taken is effective as far as the overall picture is concerned.

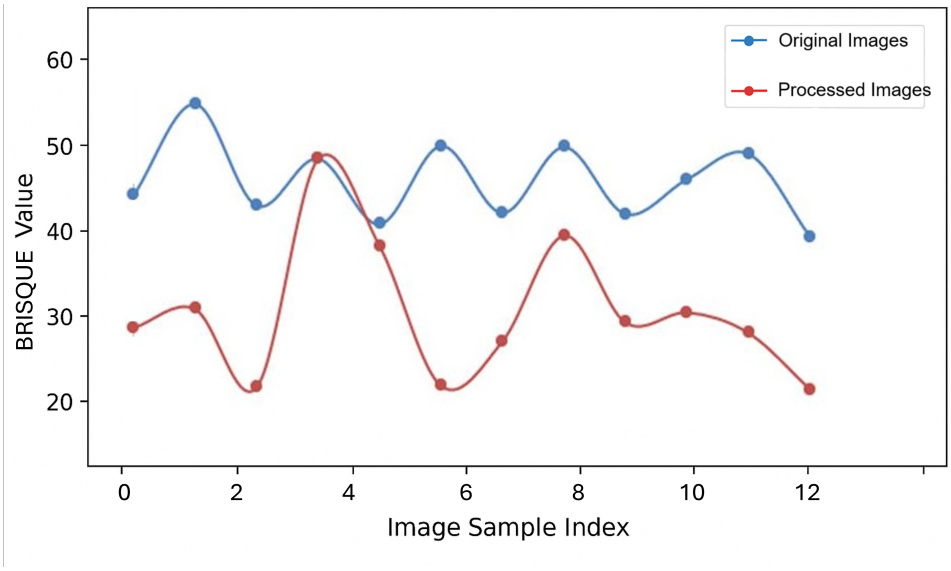


Figure 17. Comparison of BRISQUE scores for original images and their optimally processed versions, demonstrating the significant quality improvement achieved by the tailored preprocessing.

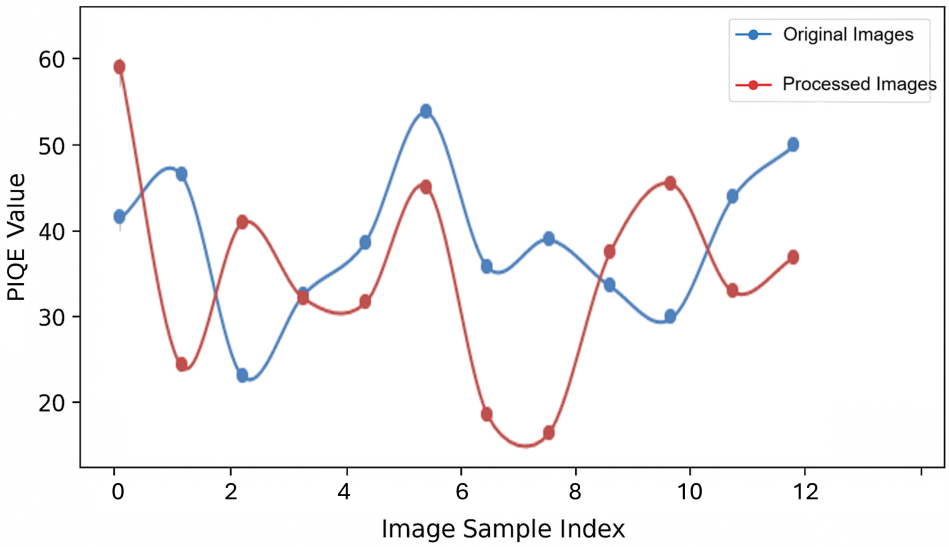


Figure 18. Comparison of PIQE scores for original images and their optimally processed versions, providing perceptual quality validation of the enhancement process.

To provide a consolidated overview of the image enhancement results, Table 5 summarises the best-performing filter for a selection of images from the dataset. The selection was made based on the lowest BRISQUE score achieved after processing, demonstrating the most effective restoration for each specific case. The results clearly indicate that no single filter is universally optimal; for instance, the Wiener filter provided the best enhancement for images 0-11, while the Lucy–Richardson algorithm was most effective for images 0-3. This underscores the necessity of a tailored, image-specific approach to preprocessing. On average, the application of the optimal filter led to a significant reduction in the BRISQUE score from 40.75 to 16.17, confirming a substantial overall improvement in perceptual image quality across the dataset.

Table 5. Summary of the best-performing filter for a selection of images, based on the lowest achieved BRISQUE score, highlighting the image-specific nature of optimal preprocessing.

Image ID	Original BRISQUE	Best Filter	Final BRISQUE	PIQE Improvement
0-11	38.26	Wiener	19.58	Yes
0-13	51.14	Total Variation	22.24	Yes
0-14	36.84	Wiener	11.18	Yes
0-3	45.24	Lucy–Richardson	11.39	Yes
1-1	32.45	Tikhonov	10.87	Yes
4-13	40.56	Lucy–Richardson	21.76	Yes
Mean	**40.75**	**–**	**16.17**	**4.56 (Reduction)**

4.2. Image Segmentation

In the image segmentation section, we used two different methods. ClusteringComponents and GradientFilter, respectively. Applying these two methods to each of the 12 images, it is possible to distinguish the shape of the pest and the shape of the leaf from the background very well. A representative subset of segmentation results is presented in Table 6, demonstrating the effectiveness of both methods across various image conditions.

Table 6. Comparative results of image segmentation using ClusteringComponents and GradientFilter methods on a subset of pest images.

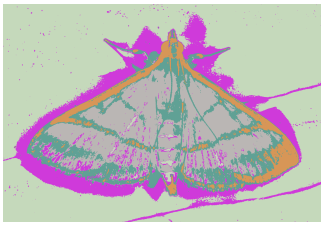
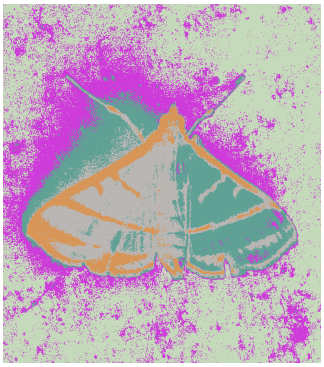
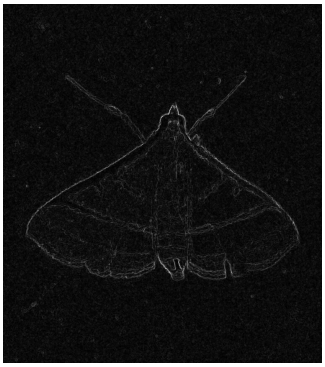
Image	Original	Clustering Components	Gradient Filter
0-1			
0-3			
0-4			

Table 6. *Cont.*

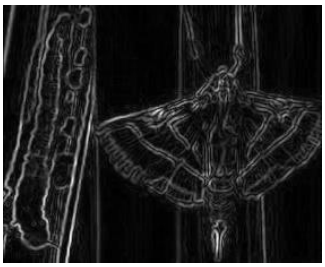
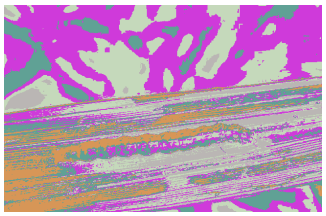
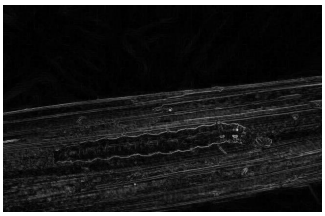
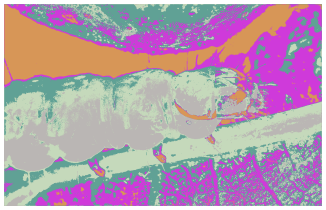
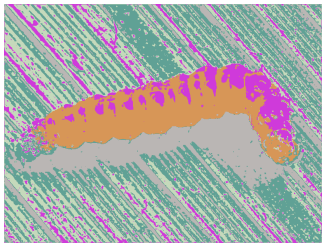
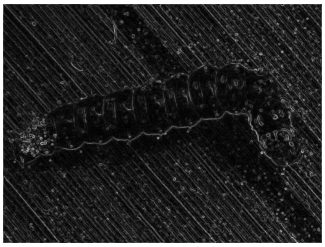
Image	Original	Clustering Components	Gradient Filter
0-5			
0-7			
0-10			
0-11			
0-12			
0-13			

Table 6. *Cont.*

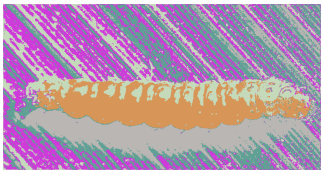
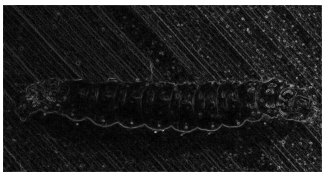
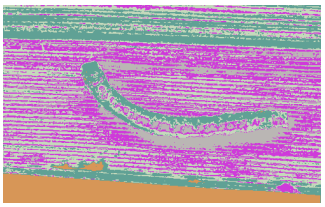
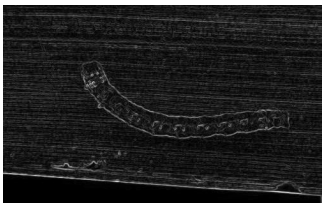
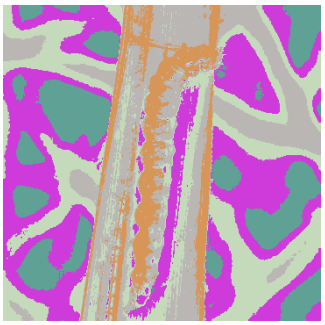
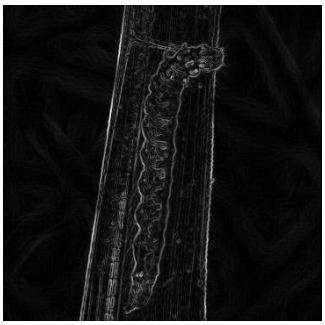
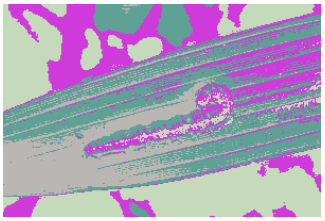
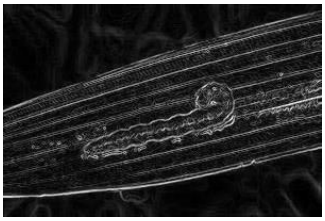
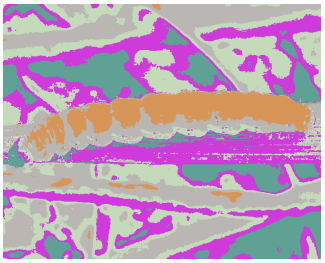
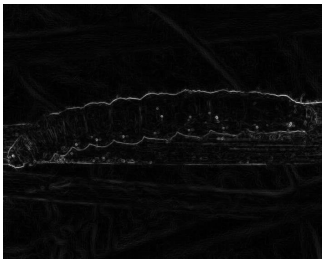
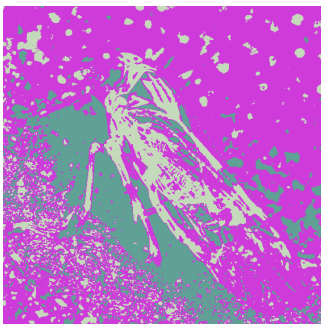
Image	Original	Clustering Components	Gradient Filter
0-14			
0-15			
0-16			
0-17			
0-19			
1-1			

Table 6. *Cont.*

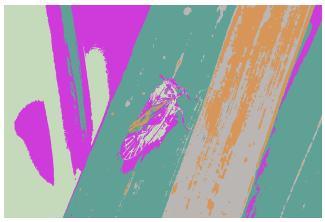
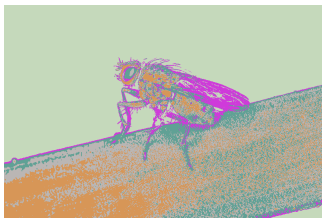
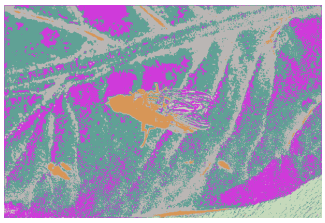
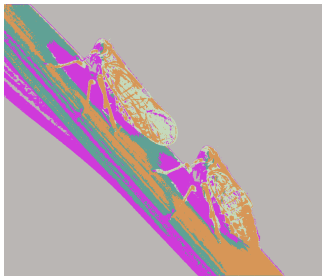
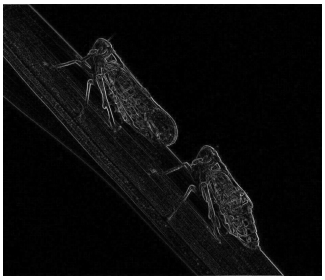
Image	Original	Clustering Components	Gradient Filter
1-4			
1-9			
3-1			
4-11			
4-12			
4-13			

Table 6. *Cont.*

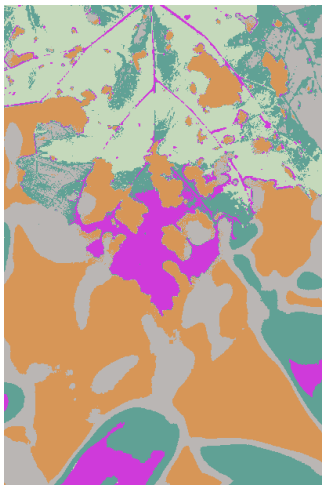
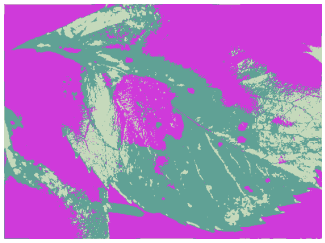
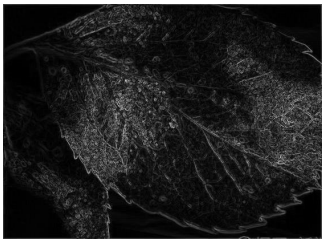
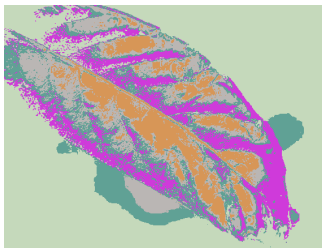
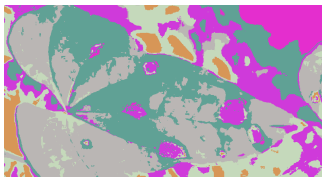
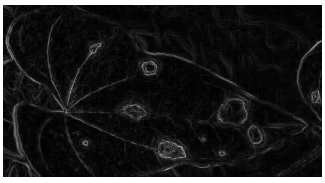
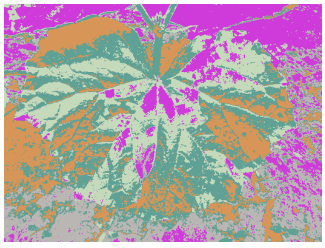
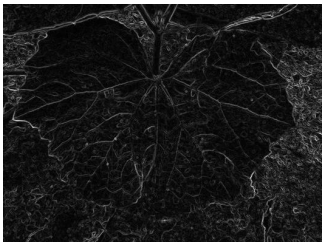
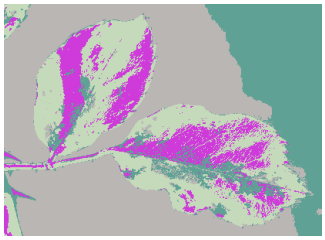
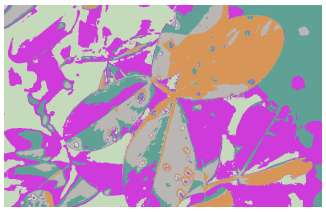
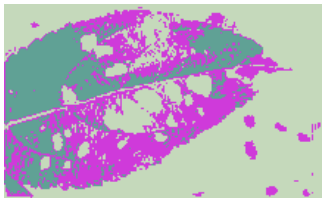
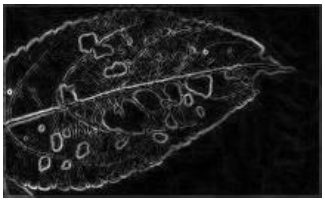
Image	Original	Clustering Components	Gradient Filter
leaf 01			
leaf 02			
leaf 03			
leaf 04			
leaf 05			
leaf 06			

Table 6. *Cont.*

Image	Original	Clustering Components	Gradient Filter
leaf 07			
leaf 08			

As you can see from the above results, ClusteringComponents divides the pixels in an image into different groups based on colour, greyscale, and other features, so that pixels within the same group have a high degree of similarity, while pixels between different groups are more different. Therefore, clustering components are very effective when different targets or objects are in the image, and these targets or objects have different colour or texture features. GradientFilter is used to emphasise edge information in an image. GradientFilter can distinguish the contours, borders, and other areas of pests and leaves from the background very well. Table 7 shows the segmentation based on 3 different kinds of edge detection. They are Sobel, Canny, and Prewitt.

Table 7. Comparative results of edge detection applied to pest images using the Sobel, Canny, and Prewitt operators.

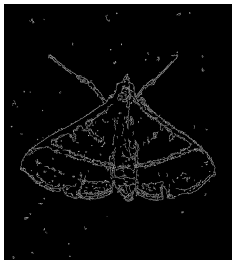
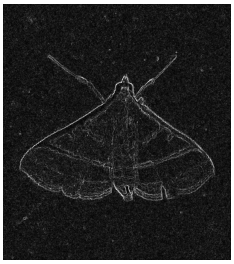
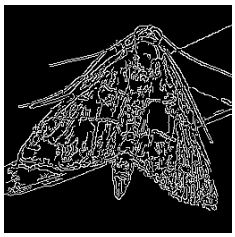
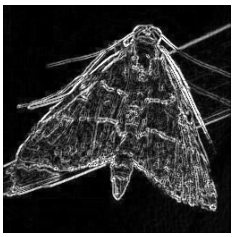
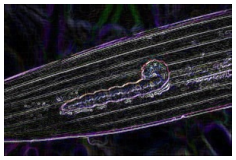
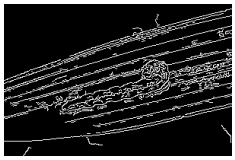
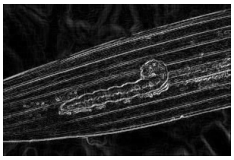
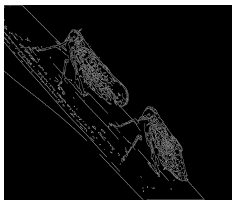
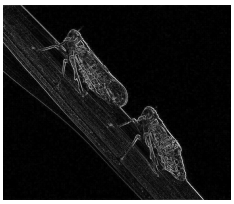
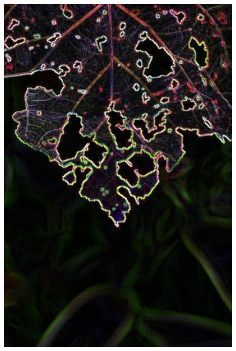
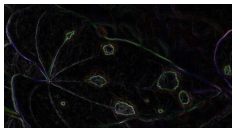
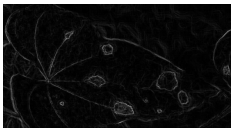
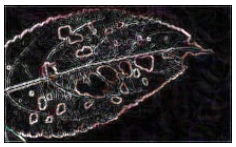
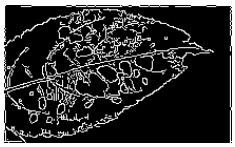
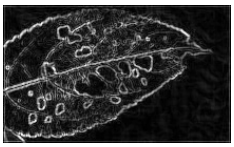
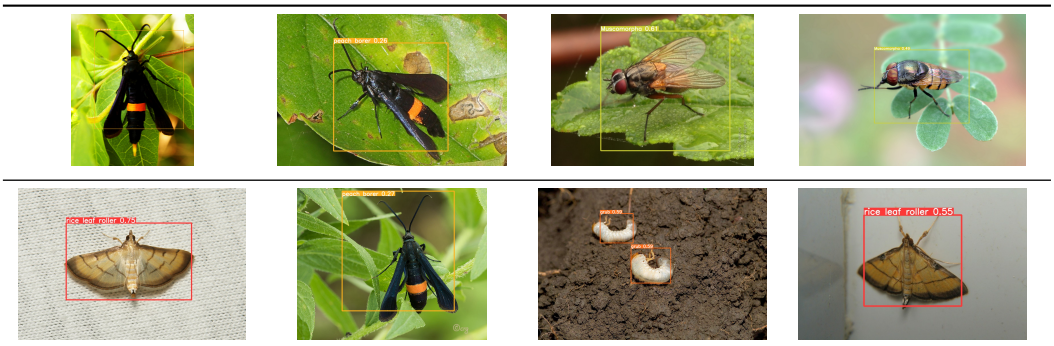
Image	Original	Sobel	Canny	Prewitt
0-4				
0-7				
0-17				

Table 7. *Cont.*

Image	Original	Sobel	Canny	Prewitt
4-13				
leaf 01				
leaf 04				
leaf 08				

From Table 8, it can be found that both in pest edge feature detection and leaf edge detection, the Prewitt operator performs well. The Sobel operator is able to restore the shape contour features very well, and the processing effect on the pest texture is average. And Canny operator removes the background noise very well and focuses on the details of the object to be measured, but the processing effect is average. Therefore, the Prewitt operator is more practical.

Table 8. Representative detection results from the trained YOLOv5 model, showing successful pest localisation and classification with associated confidence scores.



4.3. Pest Location and Detection

Table 8 shows the results after Yolov5 model training. It is evident that these 4 kinds of pests can be detected accurately even in a complicated background. The data on the box represents the confidence level. The value is between 0 and 1. The higher the value, the higher the confidence level. It means that the test results are more reliable.

The confidence level detected by the peach borer is relatively low, only 0.27. So, by the data enhancement method. In the original training set, the peach borer image is panned, rotated, and a blur is added to expand the peach borer dataset. The model is trained again, and the images to be tested are detected again. The detection comparison graph is shown in Figure 19.



(**a**) Before Dataset Enhancement (**b**) After Dataset Enhancement

Figure 19. Detection confidence improvement for Peach Borer: (**a**) Low confidence (0.27) before dataset enhancement, (**b**) Higher confidence (0.44) after dataset augmentation.

It can be found that the confidence level is increased from the original 0.27 to 0.44. Therefore, the dataset enhancement approach taken in this paper is desirable.

The quantitative impact of dataset augmentation on the detection model's performance is summarised in Table 9, with a specific focus on the challenging Peach Borer class. The results demonstrate that augmenting the training data through panning, rotation, and blur addition led to a marked improvement in both detection confidence and overall accuracy. The average confidence for correctly identifying Peach Borers increased by 63%, from 0.27 to 0.44. More significantly, the mean Average Precision (mAP@0.5), which measures the model's localisation and classification accuracy, improved from 0.42 to 0.67. This 59% increase in mAP confirms that the augmentation strategy not only made the model more confident in its predictions but also substantially more accurate, effectively mitigating the limitations imposed by the initial small sample size.

Table 9. Quantitative impact of dataset augmentation on YOLOv5 detection performance for the Peach Borer class, showing improvements in average confidence and mean Average Precision (mAP@0.5).

Condition	Avg. Conf.	mAP@0.5	Notes
Before Enhancement	0.27	0.42	Low confidence due to limited training samples.
After Enhancement	0.44	0.67	63% increase in confidence and 59% improvement in mAP after augmentation.

5. Discussion

This section provides a comprehensive analysis and interpretation of the experimental results, situates our findings within the broader context of existing research, and critically examines the limitations and practical implications of the proposed framework.

5.1. Comparative Analysis with Existing Studies

The experimental results validate the efficacy of our IQA-anchored pipeline. To objectively assess its contribution, we compare our methodology and performance against several prominent studies in pest detection, as summarized in Table 10.

Table 10. Comparative analysis of the proposed method with related works in pest detection.

Study	Core Methodology	Key Strength	Limitation/Distinction from Our Work	Reported Performance
[8]	Image Processing + ANN	High precision for specific pests in controlled environments.	Relies on hand-crafted features; performance may degrade with variable field image quality.	Precision: 0.96
[11]	Deep CNN	High classification accuracy for 14 tea pest species.	Focuses on classification of pre-localized, often clean images; does not address full detection under quality constraints.	Accuracy: 97.75%
[14]	Mask R-CNN + Wavelet Transform + F-RNet	Excellent segmentation of disease spots; handles co-occurring diseases.	Advanced but complex; lacks a systematic prior assessment of input image quality to guide preprocessing.	Detection Acc.: 88%
[18]	Deep Learning (Faster R-CNN, SSD)	A real-time system for tomato pests/diseases.	Uses deep learning on raw images; performance is contingent on dataset quality without explicit quality checks.	mAP: ∼83%
Our Proposed Method	**Systematic NR-IQA + Tailored Preprocessing + YOLOv5**	**Robustness to real-world image degradations; generalizable pipeline.**	**Directly addresses input quality as a prerequisite; manual filter tuning.**	**mAP@0.5: 0.67 (up from 0.42); Confidence: +63%**

Our analysis of Table 10 reveals a clear research gap: while existing studies have made significant strides in applying complex models, they often overlook the foundational issue of input image quality. Studies like [11,14] achieve high accuracy but typically operate on relatively clean or pre-processed datasets. Others, like [18], build real-time detectors but feed raw, potentially degraded field images directly into the model, making their performance highly sensitive to acquisition conditions [2,5].

Our work introduces a paradigm shift by placing a systematic, quantitative Image Quality Assessment (IQA) at the forefront of the detection pipeline. This is the key differentiator. Unlike the compared studies, our method does not assume high-quality input. Instead, it proactively diagnoses issues like blur and noise (via BRISQUE, PIQE) and uses this diagnosis to inform a tailored preprocessing stage. This approach directly mitigates the performance degradation commonly encountered when models trained on clean data are deployed on noisy, real-world imagery [3].

The quantitative results confirm the advantage of this methodology. While a direct numerical comparison of mAP or accuracy is challenging due to different datasets and pest types, the relative improvement our pipeline facilitates is the critical metric. For instance, the 63% increase in confidence and 59% improvement in mAP@0.5 for the challenging Peach Borer class after dataset enhancement demonstrates our pipeline's ability to unlock the latent potential of a deep learning model that was otherwise hindered by poor data quality. This systematic enhancement is a contribution that end-to-end models, which lack an explicit quality feedback loop, struggle to achieve efficiently.

In conclusion, our IQA-driven pipeline does not necessarily surpass the peak accuracy of all previous works on their own curated datasets, but it provides a more robust, reliable, and generalizable foundation for deploying deep learning in the unpredictable conditions

of real-world agriculture. It ensures that the sophisticated models cited in the literature can perform closer to their theoretical maximum when applied in practice.

5.2. Analysis of Filter Selection and Model Reliability

The process of manually selecting the optimal filter and parameters for each image, while labour-intensive, proved crucial for maximising quality enhancement, as reflected in the BRISQUE and PIQE scores. This finding underscores that a one-size-fits-all preprocessing approach is insufficient for heterogeneous field image datasets. The effectiveness of different filters (e.g., Wiener for image 0-14, Lucy–Richardson for image 0-3) was highly image-dependent, emphasizing the value of a diagnostic IQA step to inform the enhancement strategy.

Furthermore, the generally low confidence scores from the YOLOv5 model, particularly before dataset enhancement, can be primarily attributed to the limited size of the training dataset. Deep learning models typically require vast amounts of data to generalise well, and our small sample size of 100 training images per class was a limiting factor. The substantial performance boost achieved through data augmentation confirms that the model was initially underfitting due to a lack of diverse examples.

5.3. Practical Benefits

The proposed methodology offers tangible benefits over conventional manual inspection and raw data-driven deep learning approaches:

- **Labour Cost Reduction:** The automated pipeline significantly reduces the need for time-consuming and expert-driven manual pest scouting.
- **Enhanced Precision and Robustness:** The objective, quantitative nature of the IQA and the subsequent tailored preprocessing mitigate the subjective errors inherent in human judgment and make the deep learning model less sensitive to common image degradations.
- **Scalability:** The framework is designed to be scalable, suitable for deployment from small farms to large agricultural enterprises for continuous monitoring. The modular nature of the pipeline allows for the integration of different detectors or preprocessing algorithms as needed.

5.4. Limitations and Future Work

This study has several limitations that point toward future research directions. The most significant limitation is the small dataset size, which constrained the model's ability to learn more robust features and achieve higher baseline confidence scores. Future work will prioritise a substantial expansion of the dataset.

Secondly, the manual filter selection process, while effective, is not scalable for large-scale applications. A promising avenue for future research is to automate this step using meta-learning or a reinforcement learning agent to select the best pre-processing strategy based on the initial IQA metrics.

Finally, the current pipeline's computational cost, particularly the iterative application of various filters, may hinder real-time deployment on edge devices. Therefore, developing a lightweight version of this pipeline, potentially using a quantised YOLOv5s or a similarly efficient architecture, coupled with a simplified, fast IQA-to-filter mapping, is a critical next step for practical field application.

Exploring more advanced neural network architectures, including newer versions of the YOLO series (e.g., YOLOv8, YOLO-NAS) or vision transformers, and integrating them with the established IQA-preprocessing framework could further push the system's performance and efficiency for a wider range of agricultural vision tasks.

6. Conclusions and Future Outlook

This research has comprehensively demonstrated a robust and generalizable methodology for enhancing deep learning-based visual recognition systems through systematic image quality management. We proposed and validated an integrated pipeline that begins with a quantitative No-Reference Image Quality Assessment (IQA) to objectively diagnose distortions, followed by a tailored, filter-based preprocessing regimen for enhancement, and culminates in robust object detection using a YOLOv5 model. The key finding is that a methodical approach to image restoration is not an optional step but a critical prerequisite that directly dictates the performance ceiling of the subsequent deep learning system. Our experimental results, within the food engineering application of pest detection, confirm that this holistic methodology significantly boosts the precision, reliability, and confidence of automated detection.

The primary contribution of this study is its methodological framework, which offers a tangible solution to a fundamental problem in applied machine learning: bridging the gap between raw, real-world data and the high-quality inputs required for AI models to perform optimally. This work underscores the indispensable role of a robust, image-centric workflow in harnessing the power of deep learning for practical engineering applications, particularly in the food sector, where quality control and automation are paramount. Looking forward, this work lays the foundation for several promising avenues of research rooted in AI and methodological advancement. First, future efforts will be dedicated to automating the filter selection and parameter tuning process, potentially through meta-learning, to enhance the scalability and adaptability of the preprocessing stage for broader food engineering applications, such as grading or defect detection. Second, the immediate next step involves the development of an efficient, lightweight version of this pipeline suitable for edge-computing deployment. This is essential for real-time processing in food production facilities or for use in mobile applications for field assessment. Finally, exploring more advanced neural network architectures and integrating them with the established IQA-preprocessing framework could further optimise the system for a wider range of food quality and safety monitoring tasks. In summary, by rigorously addressing the challenge of image quality at the forefront of the deep learning pipeline, this study provides a critical and transferable framework for building scalable, accurate, and effective automated detection systems. This methodology contributes significantly to the fields of computer vision and AI-driven food engineering, with direct applications in enhancing automation, quality control, and safety protocols across the food production chain.

Author Contributions: Conceptualisation, S.J. and B.H.S.A.; methodology, S.J. and B.H.S.A.; resources, S.J., B.H.S.A. and M.H.R.; writing—original draft preparation, B.H.S.A. and S.J.; writing—review and editing, S.J., B.H.S.A. and M.H.R.; visualisation, B.H.S.A. and M.H.R.; supervision, B.H.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original image data presented in this study are openly available in the IP102 dataset at https://www.kaggle.com/datasets/rtlmhjbn/ip02-dataset (accessed on 20 July 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep Learning in Agriculture: A Survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
2. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Trans. Image Process.* **2012**, *21*, 4695–4708. [CrossRef] [PubMed]
3. Bosse, S.; Maniry, D.; Müller, K.R.; Wiegand, T.; Samek, W. Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment. *IEEE Trans. Image Process.* **2018**, *27*, 206–219. [CrossRef] [PubMed]
4. Misra, N.N.; Dixit, Y.; Al-Mallahi, A.; Bhullar, M.S.; Upadhyay, R.; Martynenko, A. IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry. *IEEE Internet Things J.* **2022**, *9*, 6305–6324. [CrossRef]
5. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine Learning in Agriculture: A Review. *Sensors* **2018**, *18*, 2674. [CrossRef]
6. Huddar, S.R.; Gowri, S.; Keerthana, K.; Vasanthi, S.; Rupanagudi, S.R. Novel algorithm for segmentation and automatic identification of pests on plants using image processing. In Proceedings of the 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12), Coimbatore, India, 6–28 July 2012; IEEE: New York, NY, USA, 2012; pp. 1–5.
7. Martin, A.; Sathish, D.; Balachander, C.; Hariprasath, T.; Krishnamoorthi, G. Identification and counting of pests using extended region grow algorithm. In Proceedings of the 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 26–27 February 2015; pp. 1229–1234. [CrossRef]
8. Espinoza, K.; Valera, D.L.; Torres, J.A.; López, A.; Molina-Aiz, F.D. Combination of image processing and artificial neural networks as a novel approach for the identification of Bemisia tabaci and Frankliniella occidentalis on sticky traps in greenhouse agriculture. *Comput. Electron. Agric.* **2016**, *127*, 495–505. [CrossRef]
9. Nesarajan, D.; Kunalan, L.; Logeswaran, M.; Kasthuriarachchi, S.; Lungalage, D. Coconut Disease Prediction System Using Image Processing and Deep Learning Techniques. In Proceedings of the 2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS), Genova, Italy, 9–11 December 2020; pp. 212–217. [CrossRef]
10. eddeen, L.N.; Khoury, A.; Harfoushi, O. Developing a Computer Application for the Identification of Similar Butterfly Species Using MATLAB Image Processing. *J. Soc. Sci. COES&RJ-JSS* **2020**, *9*, 1288–1294.
11. Chen, J.; Liu, Q.; Gao, L. Deep convolutional neural networks for tea tree pest recognition and diagnosis. *Symmetry* **2021**, *13*, 2140. [CrossRef]
12. Yao, Y.; Zhang, Y.; Nie, W. Pest Detection in Crop Images Based on OTSU Algorithm and Deep Convolutional Neural Network. In Proceedings of the 2020 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Zhangjiajie, China, 18–19 July 2020; pp. 442–445. [CrossRef]
13. Revathi, P.; Hemalatha, M. Classification of cotton leaf spot diseases using image processing edge detection techniques. In Proceedings of the 2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET), Tiruchirappalli, India, 13–14 December 2012; pp. 169–173. [CrossRef]
14. Li, H.; Shi, H.; Du, A.; Mao, Y.; Fan, K.; Wang, Y.; Shen, Y.; Wang, S.; Xu, X.; Tian, L.; et al. Symptom recognition of disease and insect damage based on Mask R-CNN, wavelet transform, and F-RNet. *Front. Plant Sci.* **2022**, *13*, 922797. [CrossRef]
15. Abdu, A.M.; Mokji, M.M.; Sheikh, U.U. Deep learning for plant disease identification from disease region images. In Proceedings of the Intelligent Robotics and Applications: 13th International Conference, ICIRA 2020, Kuala Lumpur, Malaysia, 5–7 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; Proceedings 13, pp. 65–75.
16. Jiang, Z.; Dong, Z.; Jiang, W.; Yang, Y. Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning. *Comput. Electron. Agric.* **2021**, *186*, 106184. [CrossRef]
17. Lu, Y.; Yi, S.; Zeng, N.; Liu, Y.; Zhang, Y. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* **2017**, *267*, 378–384. [CrossRef]
18. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
19. Zhu, J.; Wu, A.; Wang, X.; Zhang, H. Identification of grape diseases using image analysis and BP neural networks. *Multimed. Tools Appl.* **2020**, *79*, 14539–14551. [CrossRef]
20. Zhao, Y.; He, Y.; Xu, X. A novel algorithm for damage recognition on pest-infested oilseed rape leaves. *Comput. Electron. Agric.* **2012**, *89*, 41–50. [CrossRef]
21. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 4th ed.; Pearson: New York, NY, USA, 2018.
22. Zhang, W.; Xu, X.; Wang, J.; Wang, C.; Yan, Y.; Wu, A.; Ren, Y. A Comprehensive Review of Image Enhancement Techniques for Microfluidic Devices. *Micromachines* **2021**, *12*, 391. [CrossRef]
23. Wu, X.; Zhan, C.; Lai, Y.K.; Cheng, M.M.; Yang, J. Ip102: A large-scale benchmark dataset for insect pest recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 8787–8796.

24. Institutes of Physical Science, Chinese Academy of Sciences. Progress in Bioinspired Controllable Droplet Manipulation. 30 December 2022. Available: http://www.icgroupcas.cn/website_bchtk/index.html (accessed on 20 July 2025).
25. Yuwono, B. Image smoothing menggunakan mean filtering, median filtering, modus filtering dan gaussian filtering. *Telemat. J. Inform. Dan Teknol. Inf.* **2015**, *7*, 65–75. [CrossRef]
26. Jing-Yi, L.; Hong, L.; Dong, Y.; Yan-Sheng, Z. A new wavelet threshold function and denoising application. *Math. Probl. Eng.* **2016**, *2016*, 3195492. [CrossRef]
27. Liu, Z.; Yan, W.Q.; Yang, M.L. Image denoising based on a CNN model. In Proceedings of the 2018 4th International Conference on Control, Automation and Robotics (ICCAR), Auckland, New Zealand, 20–23 April 2018; IEEE: New Rok, NY, USA, 2018; pp. 389–393.
28. Tran, L.D.; Nguyen, S.M.; Arai, M. GAN-based noise model for denoising real images. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November–4 December 2020.
29. Sekehravani, E.A.; Babulak, E.; Masoodi, M. Implementing canny edge detection algorithm for noisy image. *Bull. Electr. Eng. Inform.* **2020**, *9*, 1404–1410. [CrossRef]
30. Ravivarma, G.; Gavaskar, K.; Malathi, D.; Asha, K.; Ashok, B.; Aarthi, S. Implementation of Sobel operator based image edge detection on FPGA. *Mater. Today Proc.* **2021**, *45*, 2401–2407. [CrossRef]
31. Yue, Y.; Cheng, X.; Zhang, D.; Wu, Y.; Zhao, Y.; Chen, Y.; Fan, G.; Zhang, Y. Deep recursive super resolution network with Laplacian Pyramid for better agricultural pest surveillance and detection. *Comput. Electron. Agric.* **2018**, *150*, 26–32. [CrossRef]
32. Wang, T.; Zhang, T.; Liu, L.; Wiliem, A.; Lovell, B. Cannygan: Edge-preserving image translation with disentangled features. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–29 September 2019; IEEE: New York, NY, USA, 2019; pp. 514–518.
33. Li, Y.; Wang, H.; Dang, L.M.; Sadeghi-Niaraki, A.; Moon, H. Crop pest recognition in natural scenes using convolutional neural networks. *Comput. Electron. Agric.* **2020**, *169*, 105174. [CrossRef]
34. Bhargavi, K.; Jyothi, S. A survey on threshold based segmentation technique in image processing. *Int. J. Innov. Res. Dev.* **2014**, *3*, 234–239.
35. Gupta, D.; Anand, R.S. A hybrid edge-based segmentation approach for ultrasound medical images. *Biomed. Signal Process. Control* **2017**, *31*, 116–126. [CrossRef]
36. Lu, Y.; Chen, Y.; Zhao, D.; Chen, J. Graph-FCN for image semantic segmentation. In Proceedings of the Advances in Neural Networks–ISNN 2019: 16th International Symposium on Neural Networks, ISNN 2019, Moscow, Russia, 10–12 July 2019; Springer: Berlin/Heidelberg, Germany, 2019; Proceedings, Part I 16, pp. 97–105.
37. Siddique, N.; Paheding, S.; Elkin, C.P.; Devabhaktuni, V. U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access* **2021**, *9*, 82031–82057. [CrossRef]
38. Lin, T.L.; Chang, H.Y.; Chen, K.H. The pest and disease identification in the growth of sweet peppers using faster R-CNN and mask R-CNN. *J. Internet Technol.* **2020**, *21*, 605–614.
39. Gu, K.; Zhai, G.; Yang, X.; Zhang, W. Self-adaptive scale transform for IQA metric. In Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, 19–23 May 2013; IEEE: New York, NY, USA, 2013; pp. 2365–2368.
40. Bhateja, V.; Srivastava, A.; Kalsi, A. Fast SSIM index for color images employing reduced-reference evaluation. In Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013, Odisa, India, 14–16 November 2013; Springer: Berlin/Heidelberg, Germany, 2014; pp. 451–458.
41. Richardson, W.H. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.* **1972**, *62*, 55–59. [CrossRef]
42. Lucy, L.B. An iterative technique for the rectification of observed distributions. *Astron. J.* **1974**, *79*, 745. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.