Apple Fruit Detection Model Using yolo v8 and Convolutional Neural Network

P Bharath[1], Suhas P[2], Sowmya Ranjan Panda[3], Harshith B[4], Sowmya K N[5], Vinod Kumar S[6]
[1,2,3,4] *Department of AIML, Jyothy Institute of Technology Bengaluru, India*
[5,6] *Assistant Professor Department of AIML, Jyothy Institute of Technology Bengaluru, India*

***Abstract*—Precise identification of apple fruit contours and growth locations plays a vital role in enabling smart harvesting systems and accurate yield forecasting. This study introduces a hybrid edge detection framework named RED, which integrates convolutional neural networks with rough set theory. The approach begins by leveraging the Faster R-CNN model to extract individual apples from images containing multiple fruits, effectively minimizing interference from surrounding visual noise. Following this, K-means clustering is employed to further isolate the apple target within each cropped image, enhancing focus and reducing residual background effects.**

To address challenges such as variable lighting, intricate backgrounds, and fruit occlusion, the rough set method is utilized to produce edge approximations—specifically, upper and lower bounds—that help to refine the detected fruit contours. The RED model's outcomes are then benchmarked against existing edge detection techniques commonly used in similar applications.

Results from extensive experiments reveal that the RED model delivers notably improved detection accuracy and reliability. Its performance remains stable even in visually complex and poorly lit environments, demonstrating its effectiveness over traditional edge detection operators. This robust performance positions the RED framework as a promising tool for advancing automation in fruit picking and yield estimation processes

I. INTRODUCTION

1. Apple cultivation holds a central position in global fruit production, with China emerging as the dominant player—contributing over 40% of both the global planting area and total output [1]. With the rapid advancement of precision agriculture, there is a growing emphasis on automation technologies such as intelligent fruit detection and robotic harvesting. At the core of these technologies lies the need for accurate localization and boundary identification of fruits within complex orchard environments.

Detecting the precise location and contour of apples is a particularly challenging task due to environmental interference—such as overlapping branches, variable lighting, and occlusion by leaves or other fruits. Consequently, researchers have explored a variety of edge detection and object segmentation techniques to improve reliability under natural conditions.

1.1 Traditional Edge Detection Techniques

Classical image processing methods have laid the groundwork for object contour extraction. Algorithms like those introduced by Versaci et al., which utilize fuzzy divergence metrics and entropy minimization, have demonstrated effectiveness in managing uncertainty in blurry or noisy images [2]. Han et al. addressed the limitations of the conventional Sobel operator by enhancing edge localization precision, especially in low-resolution imagery [3].

Building on foundational methods, Lu et al. augmented the Canny edge detection approach by integrating local variance analysis to improve performance on thermal imagery [4]. Sekehravani et al. enhanced robustness in noisy environments by combining median filtering with Canny's multi-stage detection process [5]. Septiarini and colleagues applied a composite technique involving Canny, morphological filtering, and image reconstruction for palm fruit segmentation. Despite success, this method faced challenges in distinguishing objects under inconsistent lighting and multicolour backgrounds [6].

Jiao et al. explored edge detection for apples in natural orchards by modeling fruit positioning using geometric fitting. While effective for isolated fruits,

the method struggled in cases of significant occlusion and irregular fruit shapes, leading to inaccuracies in edge definition [7].

1.2 Artificial Intelligence-Based Detection Methods

In recent years, deep learning has significantly improved the capabilities of edge detection models. Su et al. proposed PiDiNet, a minimalistic yet effective edge detection network that combines the strengths of traditional operators with CNN-based feature extraction to achieve both speed and accuracy [8]. Wang et al. combined visual saliency modeling with adaptive thresholding (e.g., Otsu's method) to identify prominent object boundaries even in scenes with heavy visual clutter [9].

Ganesan et al. introduced a hybrid segmentation model that addresses the pitfalls of conventional optimization methods, such as mountain climbing, by pairing it with modified clustering algorithms in multiple color spaces [10]. Similarly, Xavier Soria et al. integrated HED and Xception architectures into a single edge detection pipeline that avoids the need for model pre-training, enhancing versatility across tasks [11].

1.3 Apple Fruit Detection and Segmentation Advances

Significant progress has also been made in apple-specific object detection. Wang et al. developed MS-ADS, a segmentation network that uses masked attention to isolate apples with high accuracy. However, the complexity of the model requires substantial computational resources and large datasets for effective training [12].

Tian et al. extended the YOLO-V3 framework to accommodate apple detection across various growth stages, yielding better generalization to different orchard conditions. Nonetheless, opportunities remain to improve detection reliability under heavy occlusion and varied illumination [13]. Li et al. built a U-Net variant that incorporates residual and gated convolutions, making it more suitable for tasks involving limited training samples. Even so, challenges related to occlusion and processing latency persist [14].

Zhang et al. pursued a machine learning strategy based on statistical features such as color histograms and texture metrics (e.g., GLCM). This approach, using classifiers like random forests, provided fast and accurate segmentation but struggled with adaptability in complex orchard scenes compared to deep learning models [15].

1.4 Research Motivation and Contributions

Despite the range of techniques developed, many existing methods still face difficulties in real-world orchard environments where lighting, shadows, overlapping fruits, and background clutter can degrade detection performance. Rough set theory, a mathematical framework for dealing with uncertainty and vague information, offers a promising direction but remains underutilized in edge detection tasks.

In this work, we propose an integrated fruit edge detection framework that combines the object detection capabilities of Faster-RCNN with the decision-making power of rough sets. Our approach includes the following key contributions:

Robust Fruit Detection Using Faster-RCNN

We employ Faster-RCNN to isolate individual fruits from complex image backgrounds, reducing the influence of sky, branches, and lighting variations.

Noise Reduction via Clustering and Morphological Operations

K-means clustering and morphological enhancements are applied to recover partially occluded fruit regions, ensuring more complete segmentation.

Refined Edge Extraction with Rough Set Theory

Using upper and lower approximations from rough set theory, we extract stable and continuous fruit boundaries, improving the model's robustness under varying environmental conditions.

Together, these components form a robust edge detection pipeline that improves upon traditional and deep learning methods in terms of accuracy, noise resistance, and adaptability to orchard variability**.**

2. MATERIALS AND METHODS

2.1. Data Acquisition and Preprocessing

The entire dataset included 1500 images obtained via natural environment photog- raphy and online image collection. Part of the dataset is shown in Figure 1. In this study, 1200 images were selected as the training set, and 300 images were used as the testing set. In the preprocessing stage, the label learning tool was used to annotate the dataset.

Figure 1. Partial images in the dataset.

We conducted further analysis on the content of the dataset. The statistical analysis and distribution of image resolutions in the dataset is shown in Figure 2a. The distribution of the number of targets contained within each image in the dataset is shown in Figure 2b. In this study, the Faster R-CNN model was constructed based on Keras and TensorFlow, as shown in Figure 3. We utilized Res2Net-50 as the backbone for training, and conducted training for 100 epochs. During the training process, we employed the Adam optimizer with a batch size of 16. The input image size was set to 800 × 1333, and we applied data augmentation techniques such as random rotation (-15°~15°), random flipping, and jitter, etc., during the training process. The computer processor was an Intel (R) core i7-H8750 (Intel, Santa Clara, CA, USA), with a memory of 16.00 G and a frequency of 2.20 GHz.

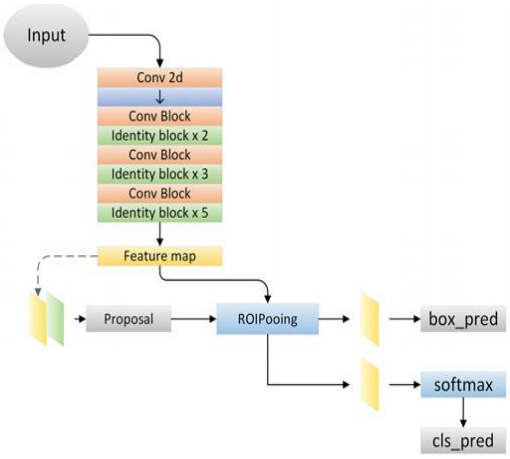


Figure 2. Faster-RCNN model architecture diagram.

In this study, a manually annotated apple image dataset was used to train the model, and complex apple images obtained from natural environments were used as the test set for accuracy testing. The results of the apple object detection are shown in Figure 4.

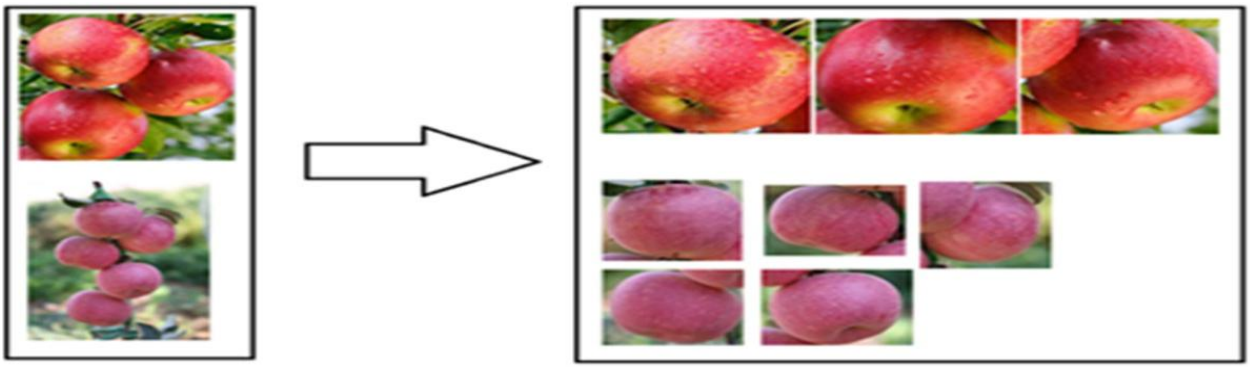


Figure 4. Display diagram of apple target detection model recognition segmentation. (a) shows the original input image and (b) shows multiple apple images for testing image segmentation.

1.1. Background Knowledge

1.1.1. Faster-RCNN

Faster-RCNN is a target detection algorithm presented by He Kaiming et al. [16], and many researchers have improved it [17–19]. The algorithm is divided into four main parts: a feature extraction network, a PNN, a pooling layer, and a classification layer.

The backbone feature extraction networks commonly used in the Faster-RCNN net- work architecture include VGG [20] and ResNet [21]. ResNet has a larger network than VGG, so it has a strong learning ability and significantly improved classification. ResNet 50 was used as the training backbone feature extraction network in this paper.

The region proposal network (RPN) [22] is an advantage of Faster-RCNN, which greatly improves the generation speed of detection frames. The method includes three steps: obtaining positive and negative classifications through SoftMax classification anchors; obtaining accurate proposals by calculating the bounding box regression offset of anchors; obtaining proposals by combining the first two steps, and removing proposals that are too small or beyond the boundary.

The classification section is the classification layer of the Faster-RCNN, which can be used to identify multiple targets and classify them. Given that this paper mainly focused on apple localization and recognition, the training category was the apple class.

1.1.2. Image Color Space Conversion

The color space [23] is the theoretical basis for color information research. This space quantifies color from people's subjective feelings into specific expressions, providing a strong basis for using computers to record and express color.

The conversion from the RGB color space to the LAB space usually starts with the transition to the XYZ color space and then further converts to the LAB space. The conversion formula is as follows [24].

RGB color space to the XYZ color space:

$$\begin{cases} X = 0.412453R + 0.35780G + 0.180423B \\ Y = 0.212671R + 0.715160G + 0.072169B \\ Z = 0.019334R + 0.119193G + 0.950227B \end{cases} \quad (1)$$

XYZ color space to LAB color space:

$$\begin{cases} L = 116 f\left(\frac{Y}{Y_n}\right) - 16 \\ a = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \\ b = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \end{cases} \quad (2)$$

where X_n, Y_n, and Z_n are the CIE XYZ tristimulus from the white point reference, each of which is X_n = 0.950456, Y_n = 1.0, and Z_n = 1.088754.

1.1.3. K-Means Clustering

The K-means clustering algorithm is an iterative clustering analysis algorithm that is suitable for unsupervised learning dataset analysis and has strong adaptability; thus, it is widely used in image segmentation [25–27].

Assume that the target image I contains N pixels, each pixel in I is represented as X_i ($1 \le i \le N$), and the three channel L, a, and b values contained in each pixel are the eigenvalues of X_i respectively. The pixel set of the same image is denoted as $D = \{X_i | X_i \in I, 1 \le i \le N\}$.

1.1.4. Dilation and Erosion of Images

Dilation and erosion are the most basic operations for determining morphology, and a reasonable combination of these operations can reduce the noise around the apple and minimize the loss of the target image.

Erosion is defined as $A \odot B = \{x|(B)_X \subseteq A\}$. For binary image A of the target after clustering, convolutional template B is used for erosion treatment. The convolutional calculation is performed between templates B and A to obtain the minimum pixel value in the coverage area of B in A, and this minimum value is used to replace the pixel value of the reference point.

Dilation is defined as $A \oplus B = \{x|(B)_x \cap A \neq \emptyset\}$. For the target image C, a convolutional template D is used for dilation processing. A convolution calculation is performed
between template D and image C. The AND operation is performed on each pixel in the scanned image. Scan each pixel in the image and perform an AND operation. If the results are all 0, the target pixel is 0. Finally, the maximum pixel value of the D coverage area in C is obtained, and this maximum value is used to replace the pixel value of the reference point.

1.1.5. Rough Set

The rough set (RS) theory [28–30] is a mathematical tool for characterizing undefined and uncertain information. This theory can be used to analyze and process various incomplete information such as imprecision, contradiction, and incompleteness effectively and obtain implicit knowledge and rules [31]. RS theory was developed by the Polish scientist Z Pawlak in 1982. The related concepts and definitions of RS are as follows.

Discourse domain U: given a finite nonempty set, the classification knowledge is embedded in the set.

Knowledge: the ability to classify objects. The object here refers to any entity, namely, the discourse domain, which is any subset family of U.

Knowledge base: the classification family on U is called the knowledge base.

Knowledge equivalence: ind(P) = ind(Q), indicating that P is equivalent to Q. P and Q are two equivalence relation families defined on the set U.

Definition 1. *For a given finite nonempty set U, R is an equivalent relationship in U, also known as R's knowledge about U.*

For a rough set, whether an object x belongs to set X can be divided into the following
situations: 1 x definitely does not belong to X; 2 x definitely belongs to X; 3 x may or
may not belong to X. Based on the above three situations, use $[x]_R$ to represent the set of all objects that are indistinguishable from x. Provide definitions for the upper approximation,
lower approximation, negative field, and boundary of set X.

$$\begin{aligned} L(X) &= \{x \in U : [X]_R \subseteq X\} \\ U(X) &= \{x \in U : [X]_R \cap X \neq \emptyset\} \\ Bn(X) &= U(X) - L(X) \\ Neg(X) &= \{x \in U : x \in U(X)\} \end{aligned} \quad (3)$$

The upper and lower approximation graphs of rough set are shown in Figure 5. The curved section in Figure 1 represents the true boundary of the identified object. The internal
area of this boundary is the lower approximation L (X) of the object, which represents the smallest definable set (positive field) that may exist in the physical position of the object. The green area that intersects with the true boundary is called the boundary domain BN (X) of the object. The area outside the boundary domain is the negative domain Neg (X) of the object, which means that the physical location of the object must not be within that area. Negating the set Neg (X) yields the upper approximation U (X) of the object, which represents the maximum definable set in which the physical position of the object may exist.

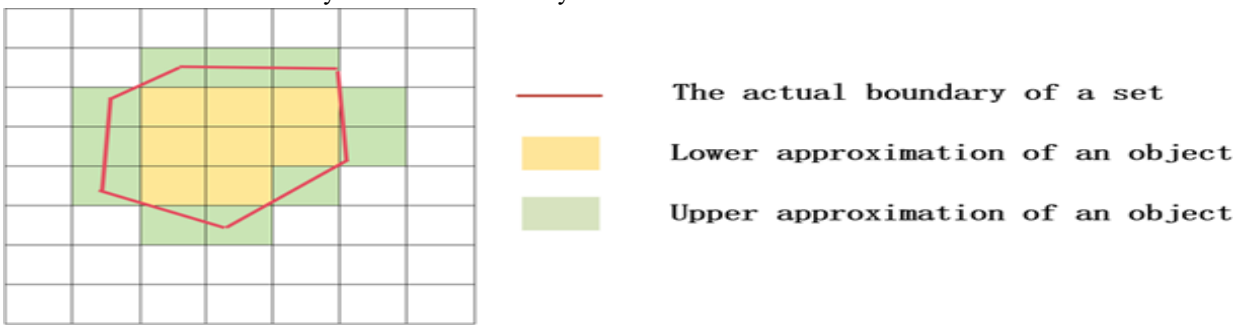


Figure 5. The upper and lower approximation graphs of rough set.

1.2. Overall Architecture of Our Edge Detection Model (RED)

This model focuses on accurate detection of the edges of apple fruit targets. Therefore, in the process of image processing, attention should be given to removing as much noise as possible and minimizing the loss of target features. The basic architecture diagram of RED is shown in Figure 6.

Figure 6. The basic architecture diagram of RED.

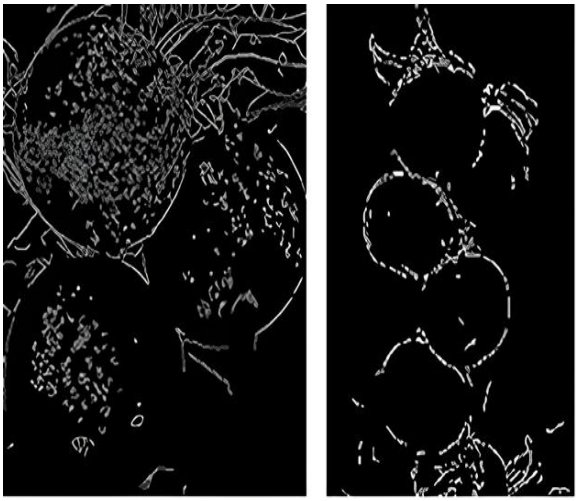
The overall algorithm consists of four modules: (**a**) object detection module, (**b**) semantic segmentation module, the red circles marked in the seg- mented image (right subplot of b) indicate the holes and noise that need to be addressed after segmentation, which will be handled in module c, (**c**) morphological image processing module, and (**d**) edge detection and image stitching module. The initial input comprises multiple images of apples in natural environments, and the final output is the corresponding edge detection result image.

First, the Faster-RCNN algorithm was used to detect the position of apples in natural environment apple tree images. Then, the RGB image was converted into a LAB color

space, and the features of the apples were extracted through K-means clustering. Next, morphological methods such as corrosion, expansion, and void filling were used to process the edges and internal noise, and the edge image of a single apple was extracted using a rough set to obtain the upper and lower approximations of the edges. Finally, the edge images of all the individual apples were merged to form a complete edge image that includes all the apples.

1.3. Detection Module

Most of the images in the dataset contain multiple apples, as well as irrelevant features, such as sky background, branches, and leaves. Direct edge detection resulted in a large amount of noise in the results (as shown in Figure 7). In response to this issue, the Faster- RCNN was used to perform individual apple detection and segmentation operations on each image, removing irrelevant features and transforming the multi-apple edge detection problem into a single apple edge detection problem.



(a) (b)

Figure 7. Effect of edge detection with a large amount of noise. Both (**a**) and (**b**) are the results of edge detection directly on the initial natural environment image, and the outcomes appear to be unsatisfactory.

1.4. Refinement Module

The refinement module was the key part of this study. The output of Faster-RCNN was used as the input of the edge detection model, and significant targets were obtained through spatial transformation, clustering segmentation, and image morphology processing after segmentation.

1.4.1. Apple Image Segmentation Based on K-Means Clustering

After being cut by the Faster-RCNN model, the image still contained irrelevant features such as leaves and branches, resulting in noise and voids when directly performing edge detection (as shown in Figure 8). For this problem, the K-means clustering method was used to cluster and segment the image to be processed, extract the target part of the image, and remove irrelevant features around the target. By conducting experiments, it was determined that setting the value of k for K-means clustering to 2 resulted in the best clustering performance.

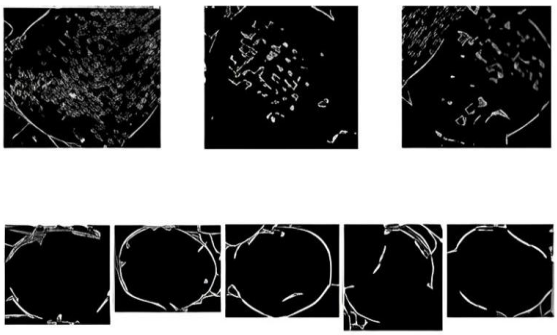
(a)

Figure 8. Effect of edge detection without clustering. (a) shows the raw images extracted from the object detection module without further processing. (b) shows the result obtained by directly applying edge detection to (a), displaying issues such as speckle noise, discontinuous edges, and lack of clarity.

The original images were all in RGB format, as shown in Figure 8a. The RGB color space is based on three basic colors, R (red: red), G (green: green), and B (blue: blue), without separating brightness information from chromaticity information. Clustering and segmenting images in RGB color space can result in a significant loss of target edge pixel information, leading to a significant gap between the edge detection effect of the target and the actual effect as shown in Figure 8b.

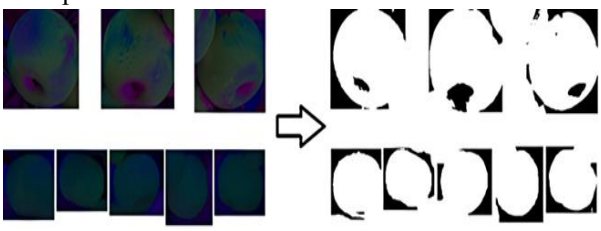
It was necessary to convert the image from the RGB space to the LAB space to minimize the pixel information loss of the target. The LAB space effectively separated brightness information and chromaticity information, where L represents brightness and a and b represent color channels.

The input image examples of the edge detection model are shown in Figure 9. The spatial transformation results for K-means clustering and cutting are shown in Figure 10.



(b)

Figure 9. Edge detection model input image examples.

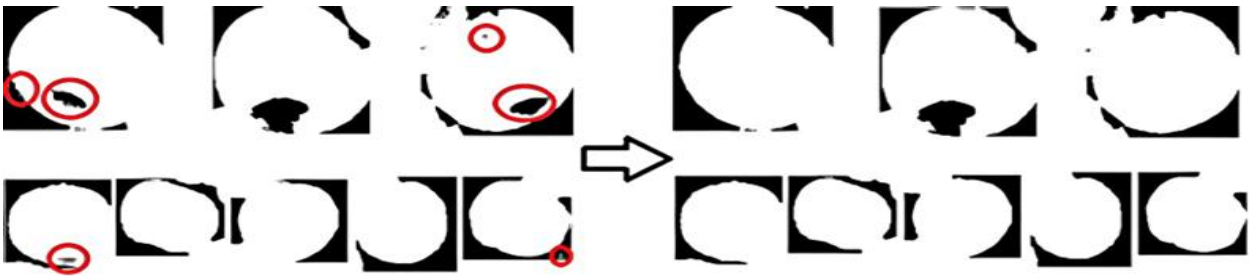


(a) (b)

Figure 10. K-means clustering for image segmentation. (a) shows the effect of converting single apple images into a Lab color space. (b) shows the results after applying K-means clustering to (a). The upper part of (b) contains voids due to the depression at the bottom of the fruit, while the lower part (b) only contains a few small voids, resulting in a relatively complete overall clustering effect.

1.4.2. Image Denoising

The clustering results in Figure 10 show that there were voids and edge noise effects in the image. This study adopted the void-filling algorithm to address the problem of voids. For edge noise, the erosion method was used to process the image edges, but simple erosion processing lost the pixel information of the target edge. Therefore, the dilation method was used to process the corroded image. The results are shown in Figure 11.



(a) (b)

Figure 11. Example diagram image processing effect after clustering and segmentation. (a) shows the effect of labeling defects such as voids after clustering. The red circle denotes the hole that exists after the segmentation. (b) shows the effect of morphological processing such as dilation, erosion, and filling in voids on the images, effectively removing most of the existing defects.

The images were filled with voids, eroded, and dilated, although after processing in this stage, the voids in the image were filled, edge noise was eliminated, and there was basically no loss of pixel information at the apple edge.

1.5. Edge Detection Module

The rough set method was the core method of this study. Currently, there are many traditional edge detection operators, such as Canny [32–34], Laplacian [35,36], Prewitt [37,38], and Holistically-Nested(HED) [39–41]. Many research papers have shown that traditional operators also have better edge detection effects.

In this paper, a structural operator for upper and lower approximation operations was defined based on rough set theory. The process of traversing objects using the structural operator convolution method to obtain the upper approximation of the image is as follows.

(1) Assume that the target object image X, the RS structure operator Y is defined, and the initial U(X) is empty.

(2) First, the structural operator Y is placed in the region corresponding to the size in the upper left corner of X, and the elements X [i, j] in object X are overlapped with those in

Y. The X [i, j] and Y [i, j] bitwise AND operations are computed. If the detection point is inside the object, it is considered to belong to U (X), and there is a high possibility of points belonging to the object around it. U (X) is added, and the sliding structural operator Y continues until the traversal is completed and the process stops. Finally, the maximum pixel value of the object coverage area can be obtained.

(3) The U (X) obtained after traversal is the upper approximation of the edge of the target object.

The process of traversing objects using the structural operator convolution method to obtain the lower approximation of the image is as follows.

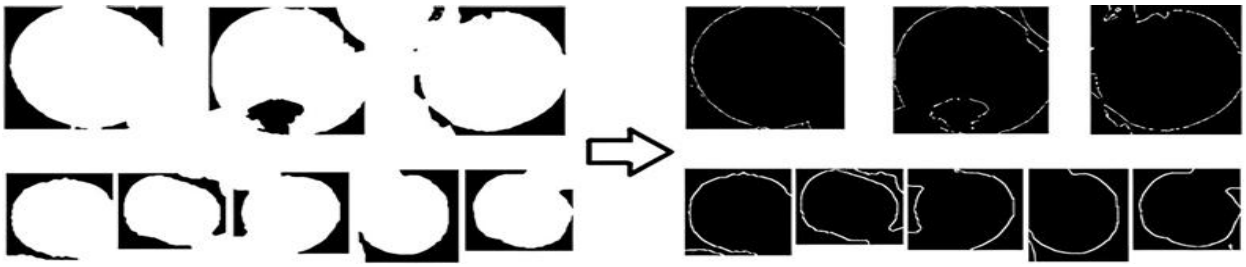
(1) Assuming the target object image X, define the rough set structure operator Y, and the initial U (X) is empty.

(2) The structural operator is first placed in the area corresponding to the size in the upper left corner, and the elements [i, j] in the objects in that area are aligned with the elements in Y. Compute [i, j] and [i, j] bitwise OR operations. If the detection point is inside the object, it is considered to belong to $L(X')$. If the detection point belongs to

an edge point, then there are points with different pixel values around it. There is a high possibility of edge points around it, so they are added to $L(X')$. The structural operator Y' is continuously slid until X' has been traversed before stopping. Finally, the minimum pixel value of the object coverage area is obtained.

(3) $L(X')$ obtained after traversal is the lower approximation of the edge of the target object.

In this study, rough set was used to obtain the upper and lower approximation edges of the target. Referring to the method for computing Bn(X) as per Formula (3), we have successfully obtained the salient contours of the target. The edge detection effect is shown in Figure 12.

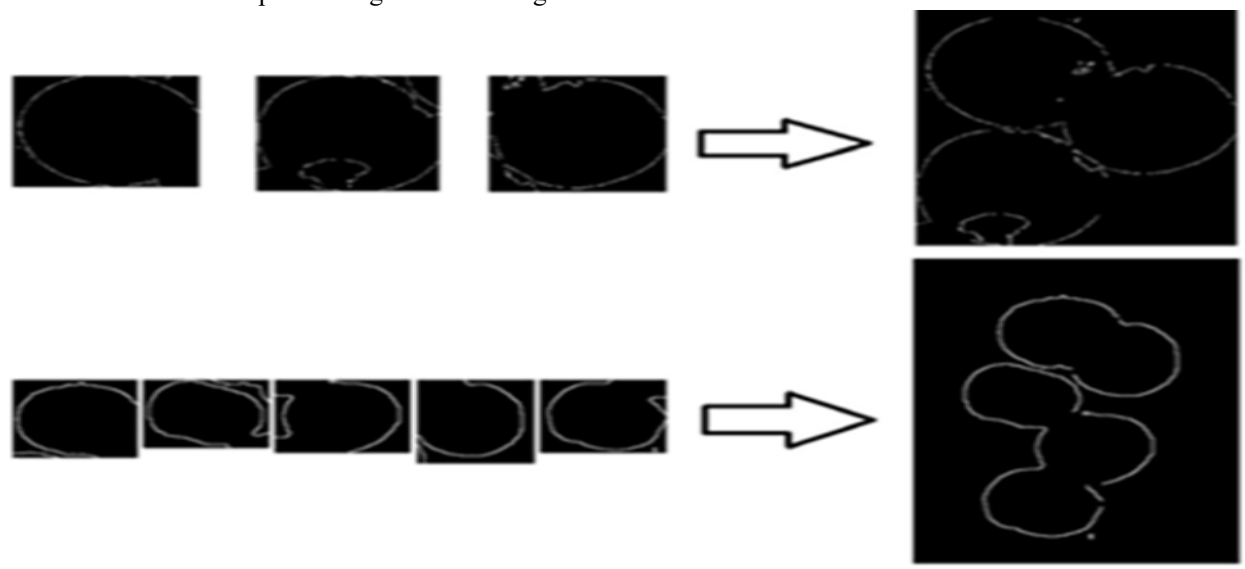


(a) (b)

Figure 12. Obtaining apple edge saliency profile images using rough set. (**a**) shows the resulting images after morphological processing. (**b**) shows the effect of edge detection based on the rough set method.

1.6. Edge Consolidation

Considering the excessive noise caused by multiple apples, an object detection algorithm based on the Faster-RCNN was adopted to segment the images. The segmented multiple images were input into the edge detection model to obtain multiple single apple edge result images, as shown in Figure 13a. The position information of each segmented image obtained via the object detection algorithm in the original image was used to draw multiple edge images of the same size to restore multiple edge images to a multi-apple image, as shown in Figure 13b.



(a) (b)

Figure 13. Effect of apple edge image merge. (a) shows the result images after edge detection. (b) shows the merging of all edge images into the corresponding edge image of the original image based on the recorded coordinates of each apple image.

1.7. Evaluation Metrics

The results of this paper were evaluated via four indicators: precision (P), recall rate (R), Dice (D), and Jaccard (J).

Accuracy was an important indicator in the performance evaluation of classifiers and was used to measure the accuracy of classifiers for samples with positive predictions. It can be expressed as the following formula:

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

where TP represents the number of samples correctly predicted as positive examples, and FP represents the number of samples incorrectly predicted as positive examples.

The recall rate was another core classifier performance evaluation indicator and was used to measure the recall rate of the classifier for actual positive samples. It can be expressed as the following formula:

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

where TP represents the number of samples correctly predicted as positive examples, and FN represents the number of samples incorrectly predicted as negative examples.

The Dice coefficient (Sørensen-Dice Dice coefficient) and Jaccard index are two indica- tors for measuring similarity, and the calculation formulas are as follows [42]:

$$Dice = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (6)$$

$$Jaccard = \frac{TP}{TP + FP + FN} \quad (7)$$

where TP represents the number of samples correctly predicted as positive examples, and FP represents the number of samples incorrectly predicted as positive examples, and FN represents the number of samples incorrectly predicted as negative examples.

The aforementioned metrics were all calculated using a confusion matrix and are commonly used for object detection tasks. We constructed the confusion matrix for edge detection results and actual edge results using the following method.

We employed two N × N filters (in the experiment, we used 3 × 3, the filter size can be adjusted reasonably based on the actual error tolerance) to slide simultaneously over the result image (filter B) and the ground truth edge image (filter A). There were four possible scenarios:

True Positive (TP): both filters detect an edge pixel.

False Positive (FP): only Filter B detects an edge pixel, but Filter A does not. False Negative (FN): only Filter A detects an edge pixel, but Filter B does not. True Negative (TN): neither filter detects an edge pixel.

Additionally, we evaluated the segmentation results of the images involved using two metrics: mask_mAP and area relative error. The calculation method for area relative error is as follows:

$$\Delta E_s = |S_1 - S_2| \quad (8)$$

$$r_e = \frac{\Delta E_s}{S_1} \quad (9)$$

where, S1 is the number of pixels in the actual apple, S2 is the number of pixels in the apple contour.

3. RESULTS AND DISCUSSION

In this study, comparative experiments were conducted using Canny, Laplacian, Pewitt, and Holistically-Nested object detection operators and the rough set edge detection algorithm based on the object detection proposed in this paper. Comparative analyses were conducted for three aspects: illumination influence, complex background influence, and dense occlusions influence.

3.1. Analysis of Detection Results Using the Segment Anything Model (SAM)

The segment anything model (SAM) aims to segment objects of interest in images based on user-provided cues. Its strength lies in having learned the concept of objects, enabling it to segment any object. Since our work involves object segmentation, it was interesting and necessary to use SAM for the segmentation of our data in the experimental section and to discuss the results.

We conducted experiments with different apple fruit images captured in natural environments using SAM. Figure 14a is a simple example image containing only three easily recognizable apples. The original image and the segmentation results are shown in Figure 14.

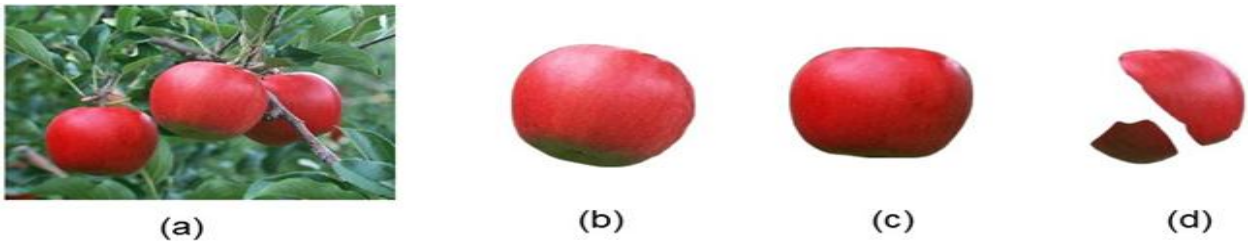


Figure 14. The original image and the segmentation results of the sparse fruit image using SAM. (**a**) represents the original image, while (**b**–**d**) display the results after SAM segmentation.

As shown in Figure 14b,c, we can observe that the segmentation results for unob- structed objects were

nearly perfect, with minimal edge loss. However, for objects that were occluded, such as occlusion between fruits or obscured by branches, as shown in Figure 14d, the segmentation results were relatively poor, with obvious missing parts in the occluded areas.

Figure 15a is a complex image example with densely packed apples and numerous occlusion factors. The original image and segmentation results are shown in Figure 15.

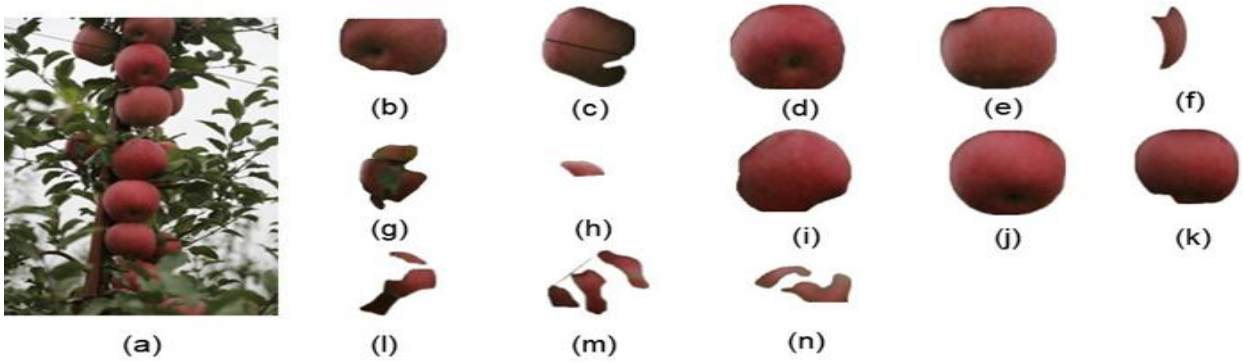


Figure 15. The original image and the segmentation results of the dense fruit image using SAM. (**a**) represents the original image, while (**b**–**n**) display the results after SAM segmentation.

From Figure 15, it can be observed that Figure 15d,e,j are relatively complete, although there was some loss of edge pixels. However, other objects experienced varying degrees of pixel loss due to occlusion by leaves or mutual occlusion between fruits, especially in segments Figure 15f–h,l–n.

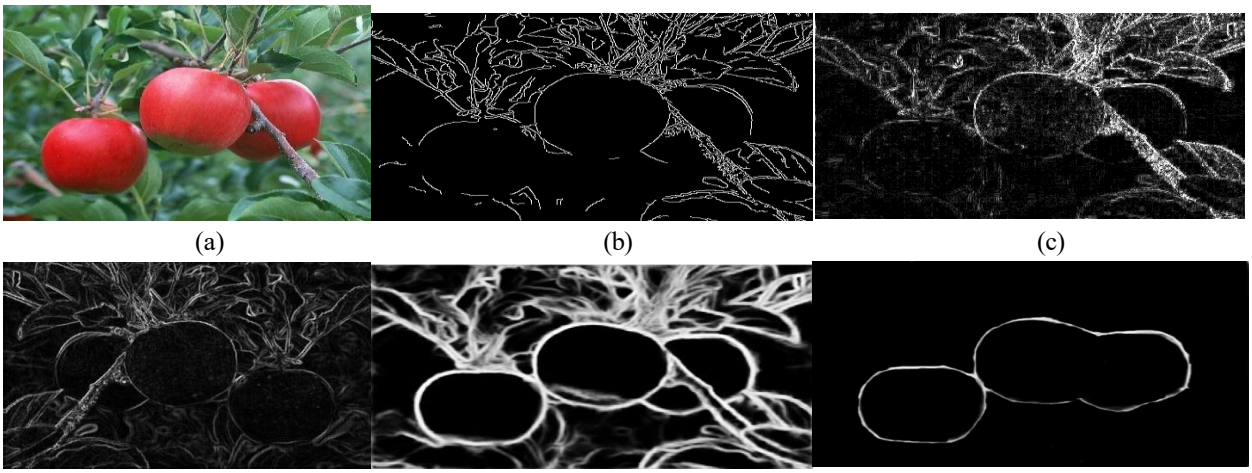
Overall, SAM can achieve remarkable segmentation results without the need for specific scene data training. However, because SAM requires achieving high-quality segmentation for every object, it cannot focus on preserving complete apples in the groundwork for edge detection. In other words, the underlying concept behind SAM is that branches, leaves, and fruits were all segmentation targets. Because SAM achieves precise segmentation for each part, there was a loss of fruit obscured by branches and leaves, which deviated from our goal of preserving the apples as completely as possible.

3.2. Analysis of the Detection Results on the Effect of Illumination

To compare the detection results of different algorithms under different illumination conditions, the Canny, Laplacian, Prewitt, and Holistically-Nested operators and the method proposed in this paper were used for detection experiments; shows the original image for detection, and the upper end of the apple in the legend has a reflective problem due to the influence of illumination. The results in Figure 16 show that the edges detected via the Laplacian and Prewitt operators retained more noise, and the edge shape was thicker. The edge continuity of the apple image detected via the Canny operator was not strong, and the edge information of the illumination part was lost, which was greatly affected by noise. The edge detection results obtained via the Holistically-Nested method surpassed those of the previous operators; however, it still faced challenges in handling interferences such as branches and leaves. Under the influence of illumination, the edge detection model that we improved resulted in relatively complete edge information detected within the apple image. Compared with the Canny operator, Laplacian operator, Prewitt operator, and Holistically-Nested operator methods in the comparative experiments, it can be observed from Figure 16 that the apple edge detection information obtained using our method performed better in terms of completeness and conciseness. However, there exists a problem of discontinuous edges due to occlusion leading to covered edge locations.

3.3. Analysis of Detection Results on the Effect of Complex Background

Considering the noise caused by the complex background of the apple in actual environmental images, edge detection experiments were conducted using the above algorithm sequentially on the apple in a complex background. The results are shown in Figure 17.



(a) (b) (c)

(d) (e) (f)

Figure 17. Comparison of the effects of each method under the influence of a complex background.

(**a**) shows the original image before detection; (**b**) shows the apple edges detected via the Canny operator; (**c**) shows the apple edges detected via the Laplacian operator; (**d**) shows the apple edges detected via the Prewitt operator; (**e**) shows the apple edges detected via the Holistically-Nested operator; and (**f**) shows the apple edges detected via the algorithm in this paper.

The results in Figure 17 show that operators such as Canny, Laplacian, and Prewitt did not eliminate edge information from backgrounds such as leaves and branches. The Holistically-Nested method provided a clear depiction of the edges of fruits, but it similarly did not remove the influence of branches or leaves, as shown in Figure 17e.

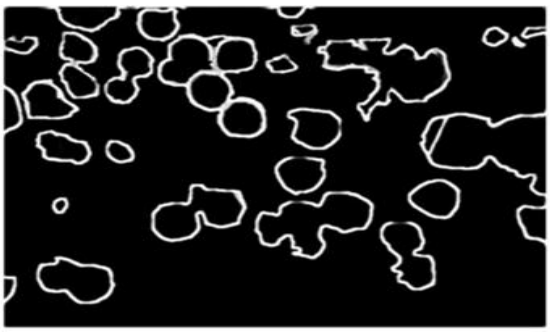
Using our method, as shown in Figure 17f, we effectively removed irrelevant objects such as branches and leaves. In the detection results, there was no point-like noise interference within the fruit. It is worth noting that the fruits on the right did not exhibit edge loss or discontinuity due to branch occlusion. This was attributed to the previous morphological processing of images and rough set, which mitigated interference and filled in defects.

3.4. Analysis of Results on the Effect of Dense Occlusions

For the situation where apples were densely distributed in the images, comparative detection experiments were conducted based on the above algorithms, and the results are shown in Figure 18.

According to Figure 18b, under a dense apple distribution, the Canny operator cannot effectively distinguish noise and detect targets, resulting in excessive noise and discontinuous edges being detected due to the dense apple distribution. Figure 18c shows that the Laplacian operator was highly coherent for edge detection between dense apples, but it did not have good noise resistance and was too sensitive to spots on apples, resulting in considerable point-like noise inside the target. Figure 18d shows that the Prewitt operator did not have good noise resistance and that there were many points, such as noise inside the target. Figure 18d indicates that the Prewitt operator did not have good noise resistance and that there was a lot of point noise inside the target. Figure 18e shows the detection results obtained using the Holistically-Nested method, which outperformed previous operators in terms of the clarity of edge delineation and the degree of internal speckle noise removal. However, it tended to capture extraneous edges influenced by tree branches and leaves. Figure 18f shows that our proposed model had significant noise resistance and had a significant effect on edge detection of dense apples. It also effectively eliminated speckle noise inside apples. However, the issue of discontinuous coverage in the detected regions due to fruit occlusion is also an aspect that requires improvement We further conducted experiments on images with more complex content, and the results are shown in Figure 19.

(a)



(b)

Figure 19. The edge detection experiments on images with more complex content. (a) shows the original image before detection. (b) shows the processed edge detection results.

3.5. Figure 19a shows the original image for detection, which contains a significant number of apple targets. The increased content resulted in lower resolution of the apple targets and more complex depth relationships among them. From Figure 19b, it can be observed that the targets located at the surface of the image are depicted with relatively clear edges. The edge detection results for the apple targets located deeper within the image exhibited varying degrees of incompleteness. Overall, the presence of multiple layers of occlusion and resolution issues both affect the effectiveness of edge detection. In real complex large-scale scenes, the performance of this method for edge detection was not ideal

Ablation Study

3.5.1. Validation of the Effectiveness of the Key Modules

We carried out an ablation experiment to verify the effectiveness of each key module. The purpose of the object detection module was to locate and identify objects of interest in the image.

Table 1 shows that when this module worked independently, background noise, illumination changes, or other factors may have interfered. We further refined the task when we introduced the clustering segmentation module so that the system could better distinguish between target and nontarget regions. The purpose of clustering was to group pixels or features based on certain similarity indicators, such as color, texture, or shape, to provide clearer target boundaries. Finally, the morphological processing module provided a way to process images to further enhance or refine the shape and structure of the target. The introduction of morphological operations such as corrosion and expansion significantly improved the elimination of noise, filled in blank areas in the target, and highlighted the main features of the target. From Table 1, it can be concluded that both the recall rate and F1-score increased by 7.3%, and the precision value increased by 7.2% after introducing the cluster segmentation module. After introducing the erosion and dilation module, the precision value increased by 7.7%, the recall rate increased by 4.0%, and the F1-score increased by 5.7%.

Table 1. Module ablation experiment. The ✓ indicates that this module was added to the ablation experiment.

Object Detection Module	Cluster Segmenta-tion Module	Erosion and Dilation Module	P/%	R/%	F/%
✓			73.4	82.3	77.6
✓	✓		80.6	89.6	84.9
✓	✓	✓	88.3	93.6	90.8

3.5.2. Clustering Segmentation K-Value Experiment

Table 2 shows that too many categories can lead to segmenting fruit regions, resulting in errors. However, if there are too few cluster centers, the fruits cannot be completely separated from the background. This was because the elements in the image were roughly divided into three categories: fruit, background, and noise. In our experiment, the

algorithm achieved the best performance when the number of cluster centers was fixed at 3 as it achieved 88.3% accuracy, 93.6% recall, and 90.9% F1-score.

Table 2. Comparison data of cluster values of clustering modules.

K	P/%	R/%	F/%
2	81.3	83.6	82.4
3	88.3	93.6	90.9
4	79.2	86.5	82.7

3.5.3. Erosion and Dilation Experiments

To fully verify the influence of the number of iterations of erosion and dilation, we further compared the models with different numbers of iterations. Table 3 shows that the F1-score will decrease when the number of iterations is too large or too small. In particular, having too many control points will seriously degrade the performance. When the number of control iterations is too large, the integrity of the image will greatly decrease. The best performance was achieved when the number of iterations was approximately
three. Therefore, in our experiment, the number of iterations for corrosion expansion was fixed at three.

Table 3. Comparison data of erosion and dilation iterations.

Iterations	P/%	R/%	F/%
1	86.2	91.4	88.7
2	87.9	92.3	90
3	88.3	93.6	90.9
5	78.1	82.8	80.4
10	63.1	66.2	64.6

Overall Analysis of the Experimental Results

Using our method, we conducted experiments in three scenarios: illumination, com- plex backgrounds, and dense occlusions. In the illumination scenario, the images contained the least number of apples, while in the dense occlusion's scenario, the images contained the highest number of apples. An illustration of the segmentation module is shown in Figure 20.

We evaluated the image segmentation performance in the three experimental scenarios, as shown in Table 4. From Table 4, it can be observed that as the number of targets increases, the Mask_mAP value gradually decreases. This indicated that with the increase in the number of apples, the segmentation performance was affected to some extent, but remained within a relatively ideal range.

Table 4. Image segmentation metrics under different scenarios.

Condition	Mask_mAP/%	Area Relative Error/%
Illumination	94.1	4.67
Complex Backgrounds	92.7	5.99
Dense Occlusions	90.6	6.14

The area relative error values exhibited a small range of variation and small numerical values, indicating that regardless of changes in influencing factors or an increase in the number of apples, the final edge detection results were essentially unaffected and remained close to the original position.

Common deep learning segmentation methods such as U-Net [43], SegNet [44], and Mask-RCNN [45] require the creation of training data for model training. Our segmentation method exclusively utilizes K-

Means for pixel-level segmentation, representing an unsuper-vised approach. This method provides a feasible solution for effectively describing apples in situations where annotated data is scarce. This is particularly important in practical applications because in many real-world scenarios, acquiring a large amount of accurately annotated data is both time-consuming and costly. Of course, for more complex scenarios and when abundant accurately annotated data is available, choosing deep learning-based methods is more effective.

We compared the edge detection results of the proposed model with those of the Canny operator, Laplacian operator, Prewitt operator, and Holistically-Nested operator in terms of three aspects: illumination effect, complex background effect, and dense occlusion effect, as shown in Table 5.

Table 5. Comparison data of various methods.

Algorithm	Illumination Effect				Complex Backgrounds Effect				Dense Occlusions Effect			
	P/%	R/%	D/%	J/%	P/%	R/%	D/%	J/%	P/%	R/%	D/%	J/%
Canny	84.6	87.2	85.8	75.3	63.7	92.6	75.5	60.6	87.9	79.9	83.7	72.0
Laplacian	74.3	86.2	79.7	66.4	40.1	88.5	55.2	38.1	82.2	95.1	88.1	78.8
Prewitt	80.1	90.8	85.1	74.1	78.8	88.3	83.2	71.3	72.5	90.4	80.4	67.3
Hed	84.8	91.1	87.8	78.3	87.7	92.4	89.9	81.8	86.3	90.5	88.3	79.1
Ours	82.6	94.4	88.1	78.7	88.3	93.6	90.9	83.3	83.8	97.6	90.2	82.1

Table 5 shows that the model proposed in this paper exhibited better adaptability in various aspects, such as the effects of illumination, complex backgrounds, and dense occlusions. In terms of illumination effect experiments, the precision value, recall rate, dice, and Jaccard of our algorithm were 82.6%, 94.4%, 88.1%, and 78.7%, except that the precision value was slightly lower than the Holistically-Nested operator. Other evaluation indexes were better than the Canny operator, Laplacian operator, Prewitt operator, and Holistically-Nested operator of the contrast experiments. In terms of complex background effect experiments, the precision value, recall rate, dice, and Jaccard of our algorithm were 88.3%, 93.6%, 90.9%, and 83.3%, and the detection effect was superior to the Canny operator, Laplacian operator, Prewitt operator, and Holistically-Nested operator. In terms of dense occlusions effect experiments, the precision value, recall rate, dice, and Jaccard of our algorithm were 83.8%, 97.6%, 90.2%, and 82.1%, except that the precision value was slightly lower than Cann

The analysis of the experimental results revealed that the model in this paper achieved noise reduction by gradually removing the environmental noise around the target fruit. Our proposed rough set edge detection method can better utilize pixel information with fuzzy fruit edges and unclear classification, more effectively extract the salient contours of fruits, and overall demonstrate stronger robustness.

4. CONCLUSIONS

Based on the application background of target edge detection, a detection method combining rough set and convolutional neural network was proposed in this paper. This method obtains the target edge image by gradually extracting the target features of the original image and minimizing the loss of target features through graphics-related methods.

In response to the problems of multiple apples in the original image, which contain too many irrelevant features and severe mutual influence between apples, a Faster-RCNN model was constructed using convolutional neural network knowledge in deep learning to segment multiple apples one by one and simplify the multi-apple edge detection problem into a single apple edge detection problem.

For edge detection of a single segmented apple image, the branches and leaves around the target still have a serious impact. According to various related

studies, the K-means clustering method was used to segment the target from the background and achieved further noise reduction. In the processed results, the target image still suffered from feature loss and a small amount of edge noise. To address this, we used cavity filling and graphic erosion and dilation methods to minimize the loss of target features while maximizing noise removal.

In terms of the core algorithm of edge detection, the rough set method was introduced to better characterize the edge image of the target in response to the uncertainty and imprecision of image edges. The experimental results showed that, considering the effects of illumination, complex backgrounds, and dense occlusions, the values of dice were 88.1%, 90.9%, and 90.2%, respectively, which were significantly higher than those of the Canny operator, Laplacian operator, Prewitt operator, and Holistically-Nested operator participating in the contrast experiment. Meanwhile, the values of Jaccard were 78.7%, 83.3%, and 82.1%, respectively, which were higher than those of the Canny operator, Laplacian operator, Prewitt operator, and Holistically-Nested operator participating in the contrast experiment.

This paper studied the positioning and edge detection of apples. The research objects of this paper were apple target location and edge detection in the natural environment, and the accuracy of apple detection was effectively improved, which provides a valuable reference for intelligent harvesting, growth analysis, and yield prediction. However, there are still some areas that require improvement. We have designed a series of processes to overcome issues such as branch occlusion, fruit occlusion, and shadow interference. Although these processes largely remove non-target pixels, the final edge detection results revealed that occlusion between fruits can lead to discontinuous edge delineation. Continuous edge delineation is crucial for capturing depth information between targets. In future work, we need to improve the issue of discontinuous edges, such as by introducing depth information to fill in the final discontinuous edges.

REFERENCES

[1] Chen, R.; Wang, J.; Li, Y.; Song, Y.; Huang, M.; Feng, P.; Qu, Z.; Liu, L. Quantifying the impact of frost damage during flowering on apple yield in Shaanxi province, China. Eur. J. Agron. 2023, 142, 126642. [CrossRef]

[2] Versaci, M.; Morabito, F.C. Image Edge Detection: A New Approach Based on Fuzzy Entropy and Fuzzy Divergence. Int. J. Fuzzy Syst. 2021, 23, 918–936. [CrossRef]

[3] Joo, J.; Han, L.; Tian, Y.; Qi, Q. Research on edge detection algorithm based on improved sobel operator. MATEC Web Conf. 2020, 309, 03031. [CrossRef]

[4] Lu, Y.; Duanmu, L.; Zhai, Z.; Wang, Z. Application and improvement of Canny edge-detection algorithm for exterior wall hollowing detection using infrared thermal images. Energy Build. 2022, 274, 112421. [CrossRef]

[5] Akbari Sekehravani, E.; Babulak, E.; Masoodi, M. Implementing canny edge detection algorithm for noisy image. Bull. Electr. Eng. Inform. 2020, 9, 1404–1410. [CrossRef]

[6] Septiarini, A.; Hamdani, H.; Hatta, H.R.; Anwar, K.J.S.H. Automatic image segmentation of oil palm fruits by applying the contour-based approach. Sci. Hortic. 2020, 261, 108939. [CrossRef]

[7] Jiao, Y.; Luo, R.; Li, Q.; Deng, X.; Yin, X.; Ruan, C.; Jia, W. Detection and Localization of Overlapped Fruits Application in an Apple Harvesting Robot. Electronics 2020, 9, 1023. [CrossRef]

[8] Su, Z.; Liu, W.; Yu, Z.; Hu, D.; Liao, Q.; Tian, Q.; Pietikäinen, M.; Liu, L. Pixel difference networks for efficient edge detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5117–5127.

[9] Wang, D.; He, D.; Song, H.; Liu, C.; Xiong, H. Combining SUN-based visual attention model and saliency contour detection algorithm for apple image segmentation. Multimed. Tools Appl. 2019, 78, 17391–17411. [CrossRef]

[10] Ganesan, P.; Sathish, B.; Sajiv, G. Automatic segmentation of fruits in CIELuv color space image using hill climbing optimization and fuzzy C-Means clustering. In Proceedings of the 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), Coimbatore, India, 29 February—1 March 2016; pp. 1–6.

[11] Poma, X.S.; Riba, E.; Sappa, A. Dense extreme inception network: Towards a robust cnn model

for edge detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 13–19 June 2020; pp. 1923–1932.

[12] Wang, D.; He, D. Apple Detection and Instance Segmentation in Natural Environments Using an Improved Mask Scoring R-CNN Model. Front. Plant Sci. 2022, 13, 1016470. [CrossRef]

[13] Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple Detection during Different Growth Stages in Orchards Using the Improved YOLO-V3 Model. Comput. Electron. Agric. 2019, 157, 417–426. [CrossRef]

[14] Li, Q.; Jia, W.; Sun, M.; Hou, S.; Zheng, Y. A Novel Green Apple Segmentation Algorithm Based on Ensemble U-Net under Complex Orchard Environment. Comput. Electron. Agric. 2021, 180, 105900. [CrossRef]

[15] Zhang, C.; Zou, K.; Pan, Y. A Method of Apple Image Segmentation Based on Color-Texture Fusion Feature and Machine Learning.Agronomy 2020, 10, 972. [CrossRef]

[16] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28.

[17] Bao, J.; Wei, S.; Lv, J.; Zhang, W. Optimized faster-RCNN in real-time facial expression classification. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Chennai, India, 16–17 September 2020; p. 012148.

[18] Yu, X.; Yuan, Y.J.J.C. Hand Gesture Recognition Based on Faster-RCNN Deep Learning. J. Comput. 2019, 14, 101–110. [CrossRef]

[19] Zhang, K.; Shen, H.J.A.S. Solder joint defect detection in the connectors using improved faster-rcnn algorithm. Appl. Sci. 2021, 11, 576. [CrossRef]

[20] Teng, B.; Zhao, H.; Jia, P.; Yuan, J.; Tian, C. Research on ceramic sanitary ware defect detection method based on improved VGG network. J. Phys. 2020, 1650, 022084. [CrossRef]

[21] Sarwinda, D.; Paradisa, R.H.; Bustamam, A.; Anggia, P. Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer. Procedia Comput. Sci. 2021, 179, 423–431. [CrossRef]

[22] Guo, Y.; Zhang, J.; Su, P.; Hou, G.H.; Deng, F.Y. The Study of Locating Diseased Leaves Based on RPN in Complex Environment. J. Phys. 2020, 1651, 012089. [CrossRef]

[23] Retter, T.L.; Webster, M.A.J.C.B. Color Vision: Decoding Color Space. Curr. Biol. 2021, 31, R122–R124. [CrossRef] [PubMed]

[24] Pardede, J.; Husada, M.G.; Hermana, A.N.; Rumapea, S.A. Fruit ripeness based on RGB, HSV, HSL, L ab color feature using SVM. In Proceedings of the 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM), North Sumatera, Indonesia, 28–29 November 2019; pp. 1–5.

[25] Khan, I.; Luo, Z.; Shaikh, A.K.; Hedjam, R. Ensemble clustering using extended fuzzy k-means for cancer data analysis. Expert Syst. Appl. 2021, 172, 114622. [CrossRef]