Article

Integrated Scale Adaptive Adjustment Factor Enhanced BlendMask Method for Pineapple Processing System

Haotian Wang , Haojian Zhang * , Yukai Zhang , Jieren Deng , Chengbao Liu , Jie Tan

Posted Date: 9 August 2024

doi: 10.20944/preprints202408.0699.v1

Keywords: Instance segmentation, 3D vision, Automated processing, Machine learning, Path planning, Mechatronics

Article

Integrated Scale Adaptive Adjustment Factor Enhanced BlendMask Method for Pineapple Processing System

Haotian Wang [1,2], Haojian Zhang [2,*], Yukai Zhang [3], Jieren Deng [2], Chengbao Liu [2] and Jie Tan [2]

[1] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China
[2] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[3] Institute of Physics, Chinese Academy of Sciences, Beijing, 100080, China
* Correspondence: zhanghaojian2014@ia.ac.cn

Abstract: The study addresses the challenge of efficiently peeling pineapples, which have a distinct elliptical form, thick skin, and small eyes that are difficult to detect with conventional automated methods. This results in significant flesh waste. To improve the process, the researchers developed an integrated system combining an enhanced BlendMask method, termed SAAF-BlendMask, and a PCP method. SAAF-BlendMask improves the detection of small pineapple eyes, while PCP ensures accurate posture adjustment for precise path planning. The system uses 3D vision and deep learning technologies, achieving an average precision (AP) of 73.04% and a small object precision (APs) of 62.54% in eye detection, with a path planning success rate reaching 99%. The fully automated electromechanical system was tested on 110 real pineapples, demonstrating a reduction in flesh waste by 11.7% compared to traditional methods. This study highlights the potential of advanced machine vision and robotics in enhancing the efficiency and precision of food processing.

Keywords: instance segmentation; 3D vision; automated processing; machine learning; path planning; mechatronics

1. Introduction

In recent years, the global production of pineapples has shown a continuous upward trend. In 2020, it is estimated that approximately 40% of the global pineapple production (around 10 billion pineapples) is utilized for processing, yielding canned pineapples, pineapple juice, and other derivative products [1]. However, during processing, especially when dealing with the pineapple's distinctive elliptical shape, thick outer skin, and deeply recessed pineapple eyes, the inability to accurately locate the pineapple eyes results in imprecise automated processing. Consequently, pineapples processed by machinery resembling the Ginaca [2] exhibit suffer from pronounced pulp wastage. cylindrical cutting leads to significant wastage, as illustrated in Figure 1 (a). Manual handling using specialized tools has emerged as a common approach to address non-standard processing, offering flexibility and processing precision, as depicted in Figure 1 (b), illustrating the manual processing of pineapples. However, it requires a considerable amount of manual labor. The key aspects of automated pineapple processing include precise identification of pineapple eyes and their rapid and accurate removal.

Artificial intelligence, especially deep learning, has revolutionized the field of fruit recognition and classification for the past few years. Technologies like Convolutional Neural Networks (CNNs) have been widely used for visual fruit recognition [3–7]. For instance, Zhang et al. [8] developed a 5-layer CNN model for fruit classification. This model was applied to both the UEC-FOOD100 dataset [9] and an in-house fruit dataset. Liu et al. [10] utilized a YOLO-v3 [11] single-stage detector, enhancing it to accurately predict circular regions for tomato localization in intricate scenes. Yu et al. [12] introduced a Mask R-CNN [13] algorithm for recognizing 100 wild strawberry images. The results indicated an average recognition accuracy of 95.78% and a recall rate of 95.41%. In 2021, Wang et al. [14] proposed a modified VGG16 [15] network to count apple flowers in images. The total flower count represented the image class, and the network was trained to estimate the visible flower count without localization. Chen et al. [16] proposed another deep learning approach to perform segmentation, rather than detection, using a DCNN.

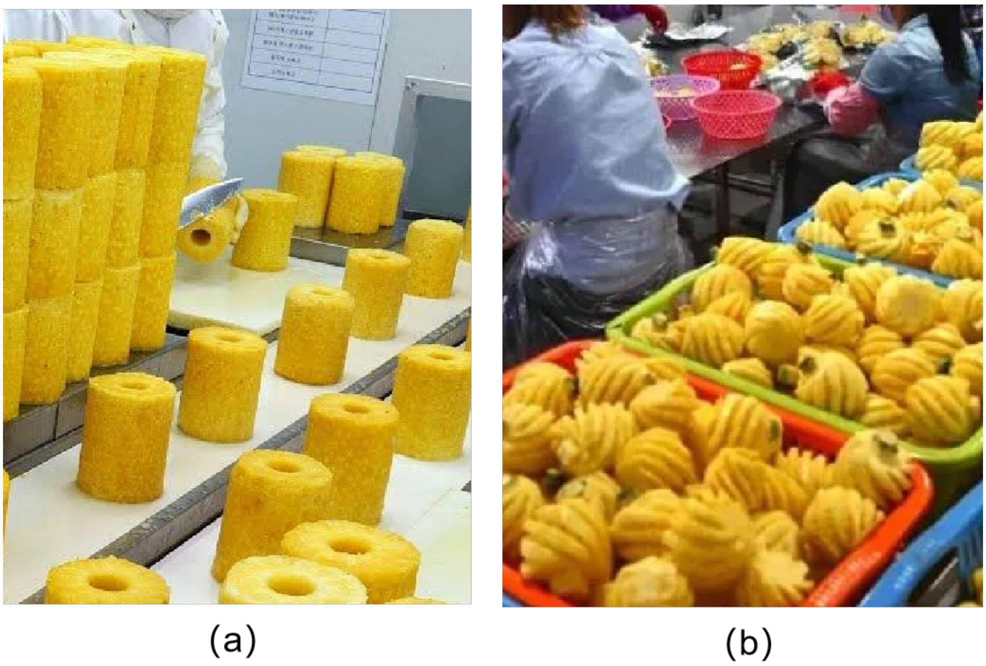

Figure 1. Pineapple processed in the traditional way. (a) Pineapple processed in an automated manner, (b) Pineapple is processed by hand.

However, there is a significant scale difference between the pineapple contour and the pineapple eyes, resulting in low detection performance for small target pineapple eyes. Moreover, during the peeling process, excessive removal of the peel results in particularly small retained pineapple eye sizes, exacerbating the difficulty of small target recognition. To address the challenge of small object detection, researchers have proposed various methods, which can be categorized as follows: data augmentation methods to address the scarcity and non-uniform distribution of small object data. Kisanta et al. [17] improved the recognition accuracy and segmentation accuracy of small objects by 7.1% and 9.7%, respectively, through oversampling and augmentation. The Mosaic data augmentation method proposed in YOLOv4 [18] randomly splices four images together by flipping, scaling, and cropping, reducing the computational load on the graphics processing unit (GPU), but potentially decreasing the model's generalization ability due to smaller object sizes. For super-resolution methods addressing the weak visual features of small objects, Haris et al. [19] proposed an end-to-end super-resolution method that uses the Faster R-CNN network to process low-resolution regions, improving small object detection performance. Bai et al. [20] proposed the multitask generative adversarial network (MTGAN), which, based on super-resolution methods, can up-sample blurry images to generate fine images corresponding to the desired scale for more accurate detection. For multi-scale fusion methods addressing the weak representation capability of single feature layers for small objects, Lin et al. [21] proposed feature pyramid networks (FPN), which integrate low-resolution strong semantic features with high-resolution weak semantic features to enhance the information of feature maps, fully exploiting the semantic information of each different scale feature map, thereby providing certain assistance for small object detection. However, during the fusion process, there may be issues with semantic gaps and noise, which to some extent hinder the further improvement of small object detection methods in terms of performance.

Pineapple processing has also gone through different stages. Traditional pineapple processing methods involve manual operations that require skilled labor to use various tools for tasks such as peeling, coring, and removing eyes. While manual peeling and slicing tools, as described in researches [22–24] exist, they are challenging to operate and have low efficiency. To address labor challenges, automated processing methods, including mechanical peeling and coring machines, have been introduced. These methods have evolved over time and encompass various technologies. For instance, Hawaiian

Pineapple Company (HPC) hired Henry Ginaca in 1911 developing a machine(Ginaca machine) that produced pineapple cylinders at a much higher rate [2]. Ginaca machine boasts high efficiency but employs a relatively coarse processing method, resulting in significant fruit loss. [25] designed a semi-automatic peeling principle machine, with a particular focus on the stability of pineapple positioning. However, this machine still employs cylindrical blades, leading to substantial fruit loss during peeling. [26] designed a semi-automatic peeling principle machine that used a new process, namely rotary peeling, and conducted theoretical analysis along with experimental verification. Although these methods have relatively higher efficiency in peeling, they still suffer from issues like lower overall efficiency, the inability to remove eyes, and the need for manual eye removal, or imprecise processing that results in significant loss of pineapple pulp.

To date, there have no complete researches on the automatic processing system of pineapple with integrated machine vision recognition. In this study, we propose an integrated approach for pineapple eye removal, incorporating scale adaptive adjustment factor and pose correction planning method. Additionally, we design and develop an innovative 3D vision-based intelligent motor-integrated pineapple processing system. This system achieves accurate identification and efficient automation of pineapple fruit processing, providing valuable insights for the automation of agricultural product processing.

2. Materials and Methods

As shown in Figure 2, the 3D vision-based mechatronics integrated pineapple processing system consists of the pineapple clamping platform mechanism, peeling mechanism (which we constructed by referencing these work [24,27]), a 3D vision system, and eye removal mechanism. The specific steps are as follows.

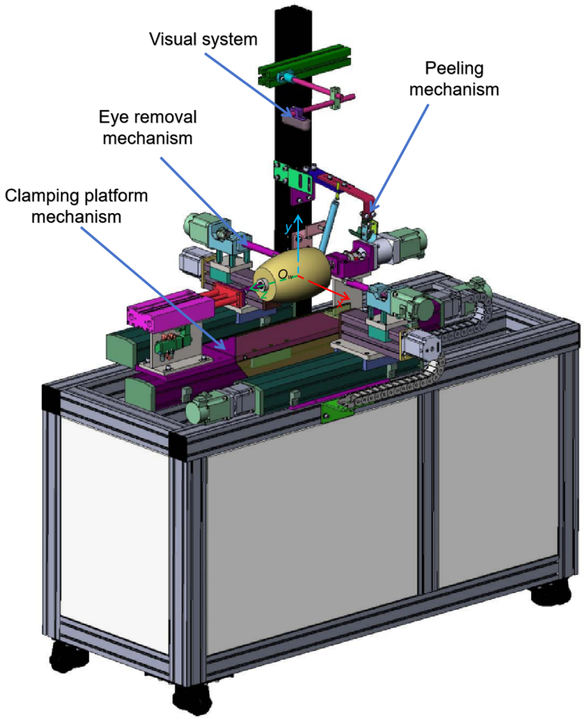

Figure 2. The processing system. Mainly consists of the clamping platform mechanism, peeling mechanism, vision system, and eye removal mechanism.

(1) *Data collecting* - employing RGB-D camera to uniformly capture six images of each peeled pineapple, comprising both RGB imagery and depth information;

(2) *Image processing* - utilizing an improved BlendMask algorithm (SAAF-BlendMask) for the identification of pineapples and pineapple eyes, followed by the reconstruction of the point cloud distribution of pineapple eyes;

(3) *Motion planning* - planning of cutting paths (PCP method) and trajectories based on the distribution of pineapple eyes and cutting the pineapple eyes following the planned trajectories.

In particular, precise removal of pineapple eyes is critical and is achieved by emulating the manual removing process, particularly in steps (2) and (3). Accurate identification of pineapple eyes and the planning of removing trajectories are of paramount importance. The premise for trajectory planning is the alignment and maintenance of the pineapple's central axis with the rotational axis, which is achieved in step (3). The following elaborate exposition will be presented.

2.1. Data Collecting

The pineapple needs to be peeled prior to image acquisition, as depicted in Figure 2. We have designed a peeling and clamping mechanism for this purpose. During the peeling process, the knife follows a spiral pattern on the pineapple's surface, effectively removing the peel. This peeling mechanism maintains a relatively consistent peel thickness, typically ranging between 6 to 9 millimeters. This minimizes fruit wastage while retaining the pineapple eyes.

We opted for an RGB-D camera, specifically the Intel RealSense D435, to establish a depth network vision system. An RGB-D camera captures RGB images with depth values, or Z-depth, which are transformed into a 3D point cloud in the camera coordinate system O_c using intrinsic parameters, and further converted to a global coordinate system O_w using extrinsic parameters, as shown in Figure 3 (b).

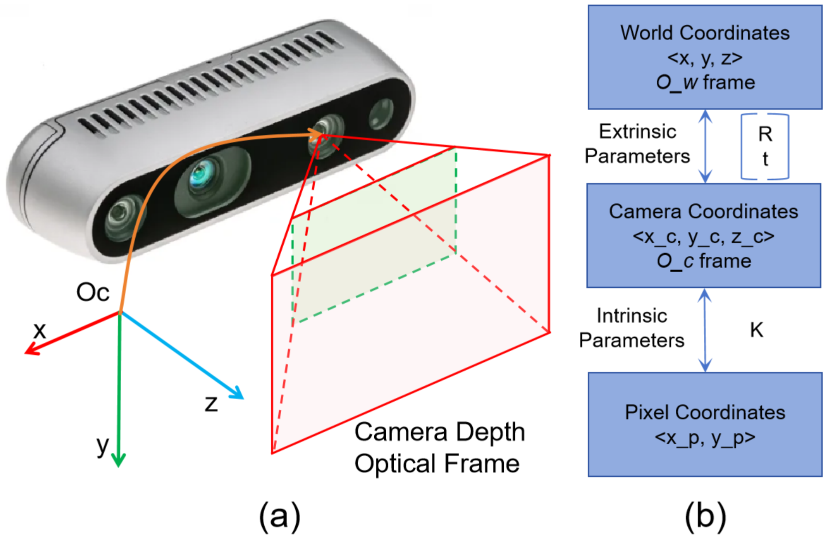

Figure 3. RGB-D camera calibration. (a) The field of view for the D435 depth camera is depicted as a frustum. The central region of the frustum represents the acceptable distance of the surface from the camera coordinate system, (b) A schematic diagram illustrates the coordinate transformation relationship between world, camera, and pixel frames. R and t denote the rotation and translation elements constituting the homogeneous transform for extrinsic parameters, while K represents the matrix embodying intrinsic parameters.

Figure 3 (a) provides a visual representation of the camera's field of view. It is noteworthy that points at greater distances from O_c exhibit higher errors (larger physical sizes corresponding to the same pixel), while points too close may become out of focus. Establishing an acceptable range is crucial to ensuring the uniform and accurate capture of points by the camera, as depicted in Figure 3 (a). Therefore, during camera installation, we aim to position the camera as close to the pineapple as possible within the effective detection range. This is to ensure that the pineapple eyes occupy a greater number of pixels.

Pineapple Image Acquisition. To enhance the effectiveness and robustness of our algorithm, we collected pineapple images from real-world scenarios.

Prior to image acquisition, the visual system was installed and calibrated with the processing equipment according to the technical specifications of the D435 camera.

In order to reconstruct the complete point cloud distribution of pineapple eyes more accurately, multiple images of each pineapple were captured from different angles. Each image had a resolution of 640×480 pixels, and images were captured at 60-degree intervals. After every capture, the pineapple was rotated by 60 degrees. After six captures, the pineapple returned to its initial angle, ensuring consistency between the reconstructed pineapple pose and its actual pose.

Then a significant number of images were collected and processed using data augmentation techniques, such as brightness adjustments, to account for varying lighting conditions in the processing environment. The collected images were further processed using the Labelme annotation software.

2.2. SAAF-BlendMask Algorithm

The BlendMask algorithm belongs to the one-stage dense instance segmentation methods. It is based on the anchor-free detection model FCOS [28] and combines instance-level information (represented by the orange dashed box in Figure 5) and finer-grained semantic information (represented by the green dashed box in Figure 5) to enhance mask prediction. It also introduces a blending module (represented by the yellow-green block in Figure 5), combining top-down and bottom-up instance segmentation methods. This effectively predicts position-sensitive instance features for dense pixel points using only one convolutional layer, preserving advantages in speed. In some cases, it outperforms Mask R-CNN in both speed and performance, meeting the real-time processing requirements of the system [29].

During reconstruction, each pineapple is reconstructed from the point cloud extracted from six images, with overlapping point clouds between adjacent images. Therefore, precision in point cloud extraction is particularly crucial. In the task of pineapple eye recognition and segmentation, segmentation networks can more accurately extract point cloud information relevant to pineapple eyes compared to traditional object detection methods (which typically use bounding boxes to label targets). Hence, we introduce the real-time instance segmentation algorithm BlendMask as the baseline.

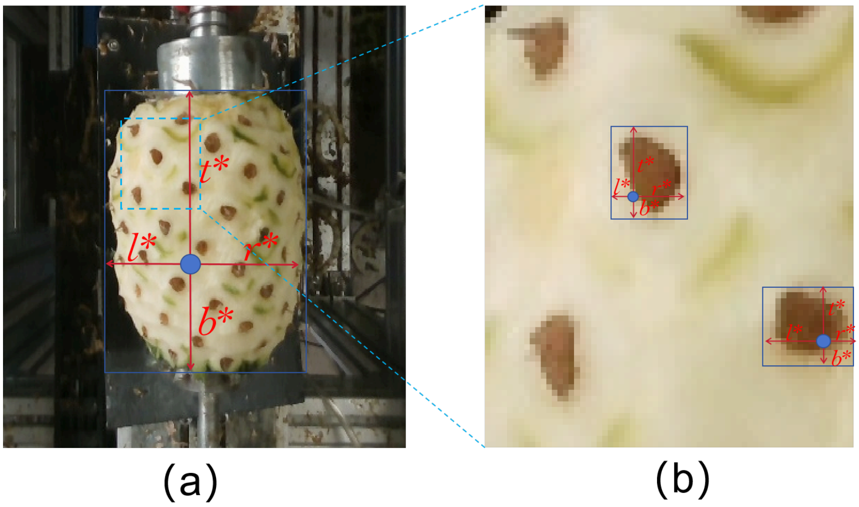

Figure 4. The performance of FCOS. In the diagram, the rectangles represent the ground-truth (GT) boxes predicted by FCOS, the blue dots represent anchor points, and l^*, t^*, r^*, and b^* denote the distances from the anchor points to the left, top, right, and bottom sides of the boxes, respectively

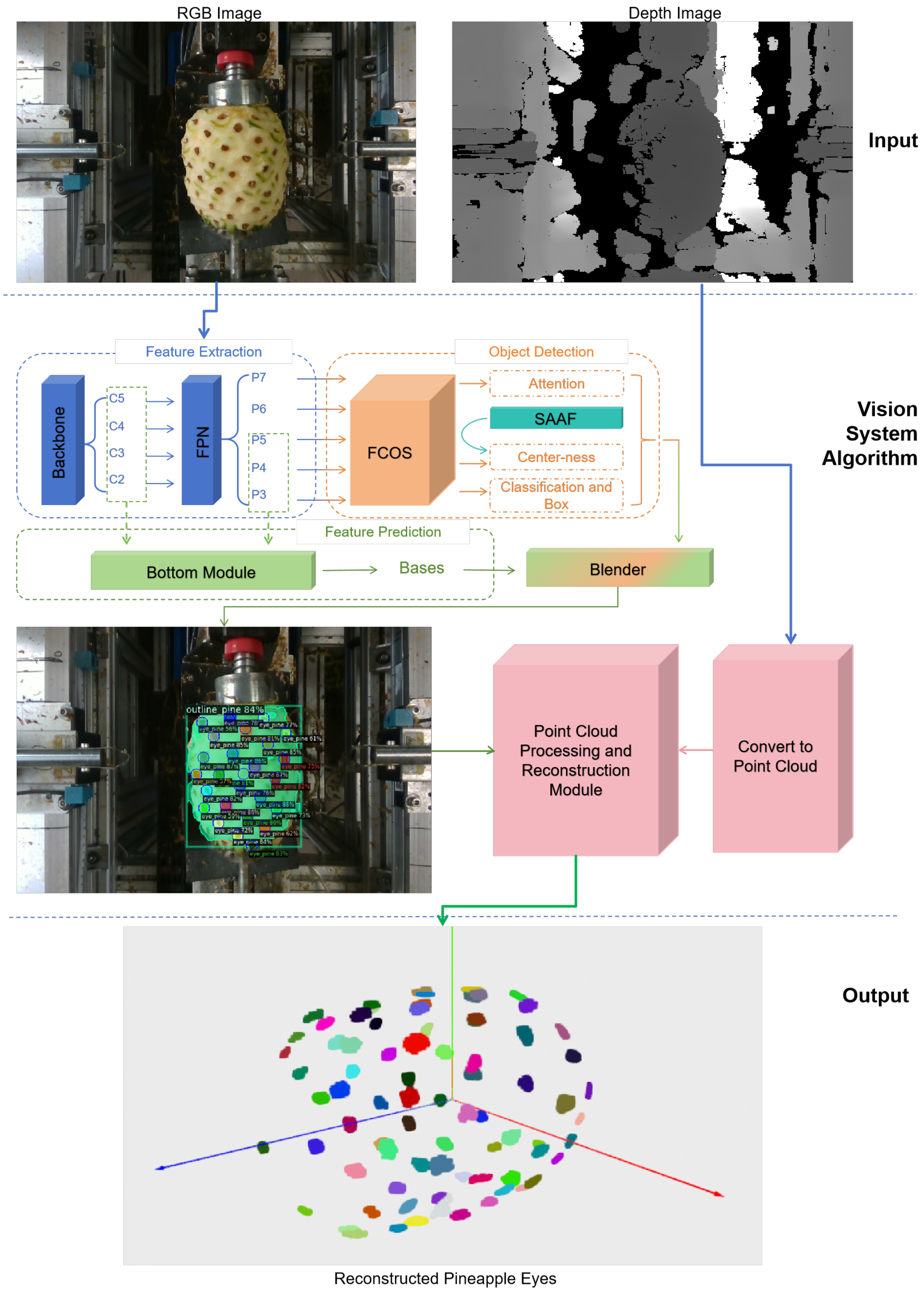

Figure 5. The overall structure of vision system algorithm. It comprises two main components: instance segmentation and 3D point cloud reconstruction. The instance segmentation, based on the improved BlendMask [29], is further divided into four major modules: the feature extraction module (denoted by the blue dashed box), object detection module (represented by the orange dashed box), feature prediction module (indicated by the green dashed box), and fusion module (depicted by the orange-green block). This constructed network facilitates the rapid segmentation of pineapple and pineapple eyes' outlines, thereby enhancing the recognition rate of small-target pineapple eyes. The point cloud reconstruction section includes depth image processing, point cloud filtering, and pineapple eye reconstruction.

To enhance the recognition performance of small-scale pineapple eyes and mitigate the influence of pineapple contour variations and eye size discrepancies, we devised a scale-adaptive balancing factor to refine the BlendMask algorithm [29]. The overall architecture of the vision system algorithm, as depicted in Figure 5, comprises two main components: instance segmentation and 3D point cloud reconstruction.

This algorithm employs a pyramid structure network (FPN), efficiently extracting feature maps at different resolutions, addressing the issue of target overlap. To address the challenge posed by the imbalance in size between pineapples and pineapple eyes in this application scenario, we proposed the Scale Adaptive Adjustment Factor (SAAF) to enhance the BlendMask loss function, thereby improving the recognition accuracy of small target pineapple eyes.

In the FCOS architecture's head module, pixel-wise target detection is performed using different hierarchical feature maps to predict targets, regressing 4D vectors $T^* = (l^*, b^*, r^*, t^*)$ for the target bounding boxes to improve accuracy, as shown in Figure 4. To enhance accuracy further, the concept of centerness [28] is introduced to eliminate low-quality predicted boxes that are far from the center.

$$centerness^* = \sqrt{\frac{min(l^*, r^*)}{max(l^*, r^*)} * \frac{min(t^*, b^*)}{max(t^*, b^*)}}. \tag{1}$$

To better address the issue of large-scale differences between pineapple eyes and pineapples leading to overlap, we conducted a detailed statistical analysis. The results indicated that the average resolution of pineapple eyes is about 20x20 pixels, while pineapples have an average resolution of about 200x250 pixels. Therefore, we adapted FCOS's feature maps with five different sizes, denoted as $P3$, $P4$, $P5$, $P6$ and $P7$, each with corresponding strides of 8, 16, 32, 64, and 128. We limited the regression distances differently for each feature map to achieve this purpose. Specifically, each feature map Pi was assigned lower and upper bounds m_{i-1} and mi as constraints for regression distances. For each coordinate point (x, y) on feature map Pi, if $max(l, t, r, b) > m_i$ or $min(l, t, r, b) < m_{i-1}$, it was treated as a negative sample on that feature layer, and regression was no longer performed. $m2$, $m3$, $m4$, $m5$, $m6$ are set to 0, 64, 128, 256, and +∞, respectively. This implies that feature map $P3$ handles target regression values within the [0, 64] range, while feature map $P7$ is responsible for targets with regression values greater than 256.

In the previous analysis, although FCOS employs hierarchical feature maps to predict targets of different scales, the practical situation is that, on average, the number of anchor points in pineapple contour significantly exceeds the number of anchor points on pineapple eyes. This imbalance leads to differences in center-ness weight between anchor points on large pineapple targets and small pineapple eyes. Specifically, the center-ness weight of small pineapple eyes tends to be lower, thus, they are often treated as low-quality regression boxes and suppressed. However, in the processing pipeline of this study, the peeling operation results in significant differences in the size of retained pineapple eyes, including very small pineapple eyes being quite common. Since the system's goal is to accurately remove each residual pineapple eye, identifying these commonly occurring small pineapple eyes becomes crucial. To address the issue of weight imbalance caused by differences in target scales, we introduced the SAAF module to enhance centerness. This enhancement aims to ensure that small pineapple eyes are equally well-recognized and processed, thereby improving the overall performance of the system.

The inputs of the SAAF module consist of ground-truth and all positive samples, while the output is the adjustment factor. Initially, we compute the average area μ_{sgt} of the ground-truth,

$$\mu_{sgt} = \sum_{i}^{N_{gt}} s_{gti} / N_{gt}, \tag{2}$$

where, s_{gti} represents the area of the $i-th$ ground truth, N_{gt} represents the number of ground truths. corresponding to the area $s_{sa(x,y)}$ of each positive sample (at the (x,y) point) from the ground-truth.

$$s_{sa(x,y)} = (l^* + r^*) * (b^* + t^*). \tag{3}$$

We determine the centerness of each positive sample and calculate a factor $\omega_{s(x,y)}$ for each sample's centerness.

$$\omega_{s(x,y)} = \sqrt{\frac{\mu_{sgt}}{s_{sa(x,y)}}}, \tag{4}$$

$$\mu_{\omega s(x,y)} = \sum_{x,y} \omega_{s(x,y)} / N_s, \tag{5}$$

this is achieved by comparing against a threshold $\mu_{\omega s(x,y)}$, N_s represents the number of positive samples for each ground-truth.

$$k_{x,y} = \begin{cases} \omega_{s(x,y)} * m, (\omega_{s(x,y)} > \mu_{\omega s(x,y)}) \\ 1 * n, (\omega_{s(x,y)} < \mu_{\omega s(x,y)}) \end{cases}, \tag{6}$$

where m and n represent hyperparameters, with a default value of 1. So, each positive sample's weight is represented as $s_{x,y} = k_{x,y} * centerness^*$, which means it's adaptively adjusted based on the ground-truth scale. After this adjustment, the loss function still consists of three parts: classification loss, regression loss, and centerness loss, combined.

$$\begin{aligned} & L(\{p_{x,y}\}, \{t_{x,y}\}, \{s_{x,y}\}) \\ = & \frac{1}{N_s} \sum_{x,y} L_{cls}(p_{x,y}, c^*_{x,y}) \\ & + \frac{1}{N_s} \sum_{x,y} 1_{\{s^*_{x,y}>0, c^*_{x,y}>0\}} L_{reg}(t_{x,y}, t^*_{x,y}) \\ & + \frac{1}{N_s} \sum_{x,y} 1_{\{c^*_{x,y}>0\}} L_{ctr}(s_{x,y}, s^*_{x,y}), \end{aligned} \tag{7}$$

where, $p_{x,y}$ represents the scores for each class predicted at the (x,y) point in the feature map. $t_{x,y}$ represents the predicted target bounding box information at the (x,y) point in the feature map. $s_{x,y}$ represents the centerness weight predicted at the (x,y) point in the feature map. N_s represents the number of positive samples. $c^*_{x,y}$ represents the true class labels at the (x,y) point in the feature map. $1_{s^*_{x,y}>0}$ is 1 when the (x,y) point in the feature map is matched as a positive sample, otherwise 0. $1_{c^*_{x,y}>0}$ is 1 when the adjusted weight at the (x,y) point is greater than 0, otherwise 0. $t^*_{x,y}$ represents the true target bounding box information corresponding to the (x,y) point in the feature map. $s^*_{x,y}$ represents the weight adjusted adaptively through the (x,y) point in the feature map.

Next, we transformed the depth images into point clouds and integrated them with segmented pineapple eye regions to generate corresponding pineapple eye point cloud data. We successfully stitched together point clouds for all pineapple eyes as shown in Figure 5, achieving a high-quality reconstruction of the entire pineapple. To achieve the effect of artificially removing pineapple eyes, as shown in Figure 1 (b), the next step is cutting path planning.

Next, we transformed the depth images into point clouds and integrated them with segmented pineapple eye regions to generate corresponding pineapple eye point cloud data. To facilitate understanding and computation, we establish the origin O_w of the world coordinate system on the rotation axis of the pineapple, with the Z-axis aligned along the rotation axis and the Y-axis oriented vertically upward, as illustrated in Figure 2. Assuming the camera's extrinsic matrix is denoted as ${}^{w}T_c$, and the point cloud obtained from image segmentation after each capture is represented as P_{ci} for $i \in \{1, \ldots, 6\}$. The world coordinate system's point cloud P_{wi} is given by:

$$P_{wi} =^{w} T_c P_{ci}. \tag{8}$$

After each capture, the pineapple undergoes a 60° rotation around the Z-axis, with the corresponding rotation transformation matrix denoted as T_{60}. Consequently, the previously obtained world coordinate system's point cloud P_{wi} undergoes a rotation and is updated accordingly:

$$P_{wi} = T_{w60} P_{wi}. \tag{9}$$

This rotation operation is performed each time a new set of data is collected.

Through the above operations, we successfully stitched together point clouds for all pineapple eyes as shown in Figure 5, achieving a high-quality reconstruction of the entire pineapple. However, to achieve the effect of artificially removing pineapple eyes, as shown in Figure 1 (b), cutting path planning is necessary.

2.3. Pose Correction Planning (PCP) Method

After reviewing the literature [30,31], we have found that the distribution of pineapple eyes on each pineapple follows 8 helical lines. Using a specialized tool [22], we can cut along each helical line, thereby removing all pineapple eyes after cutting along the 8 helical lines.

However, during the pineapple loading process, it cannot be ensured that the rotation axis of the pineapple aligns with its own central axis. As illustrated in Figure 6 , the red dashed line represents the rotational axis of the pineapple (i.e., the Z-axis of the mechanism coordinate system constructed in Section III, Subsection A), while the green dashed line denotes the central axis of the pineapple. The challenges and focal points lie in how a computer can accurately identify the 8 helical lines of pineapple eyes and quickly and precisely remove them. To address these issues, we developed the Pose Correction Planning (PCP) method, as outlined in Algorithm 1. The key steps are as follows:

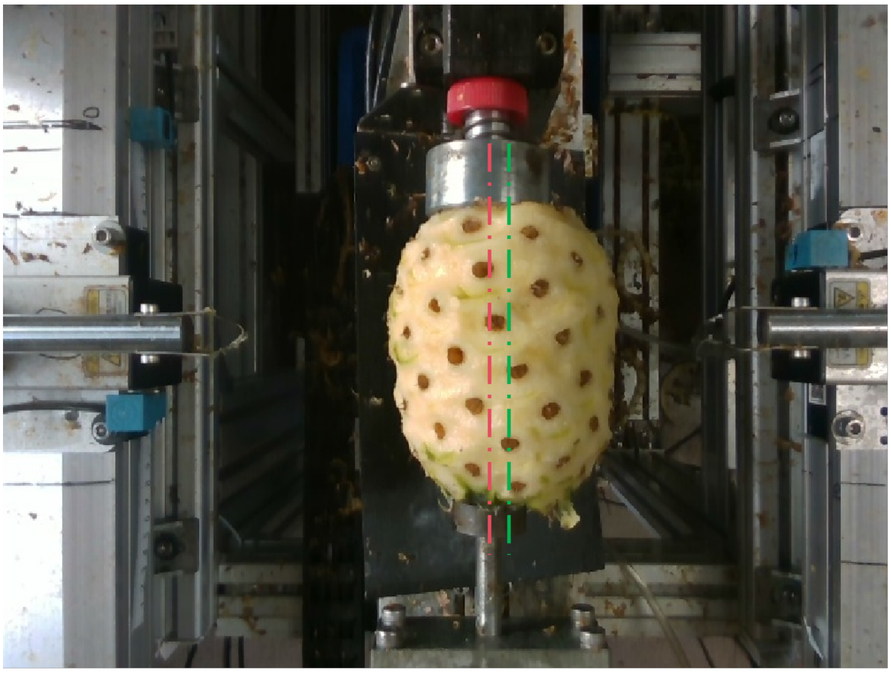

Figure 6. The clamping state of the pineapple after feeding and peeling.

```
Algorithm 1: PCP(Pose Correction Planning) Method
  Data: $S_{pe}$: reconstructed pineapple eyes point cloud $S_{pp}$: reconstructed pineapple point cloud
  Result: $D_l$: optimized cutting paths, a dictionary composed of 8 lists
  voxel subsampling to the pineapple point cloud: $S_{pps} = VoxelSample(S_{pp})$;
  $O_p$: pineapple's coordinate system;
  perform Principal Component Analysis (PCA) on $S_{pps}$ to obtain the central symmetric axis
    and corresponding pineapple's coordinate system: $O_p = myPCA(S_{pps})$;
  $P_{ei}$: the $i$-th pineapple eye;
  $P_{eci}$: the $i$-th pineapple eye center;
  $S_{pps}$: voxel subsampling of $S_{pp}$;
  foreach $P_{ei}$ in $S_{pe}$ do
      radius filter of each pineapple eye: $S_{pe}$ = RadiusFilter($S_{pe}$);
      compute centroid coordinates of each pineapple eye: $P_{eci} = Centroid(P_{ei})$;
  end
  ${}^{p}T_w$: transformation matrix from the world coordinate system to the pineapple coordinate
    system;
  $P_{peci}$: coordinates of the centers of each pineapple eye in the pineapple coordinate system;
  $P_{peci} = {}^{p}T_w P_{eci}$;
  $P_{peci(\theta-z)}$ : pineapple eye centroids in the $\theta - z$ coordinate system;
  project to obtain the distribution of pineapple eye centroids in the $\theta - z$ coordinate system:
    $P_{peci(\theta-z)} = MyProject(P_{peci})$;
  $S_{pj}$: the set of pineapple eye in $j$-th spiral line, $j \in \{1, \ldots, 8\}$;
  apply a clustering algorithm to determine the categories of pineapple eye centroids along the
    eight spiral lines: $(S_{p1},S_{p2},S_{p3},S_{p4},S_{p5},S_{p6},S_{p7},S_{p8}) = MyClusterAL(P_{peci(\theta-z)})$;
  sort the centroids on each line based on the Z-coordinate to obtain eight sequences of
    pineapple eye centroids, representing machining and cutting paths: $D_l =$
    $MySort(S_{p1},S_{p2},S_{p3},S_{p4},S_{p5},S_{p6},S_{p7},S_{p8})$;
  return $D_l$
```

1) Uniform sampling of the pineapple surface point cloud (S_p) yields a consistent and well-distributed point cloud (S_{ps}) of the pineapple surface.
2) Use Principal Component Analysis ($myPCA(\cdot)$) to determine the pose or coordinate system (O_p) of the pineapple.
3) Compute the centroid (P_{wi}) of the pineapple eye surface point cloud obtained with previous sections of the paper.
4) Calculate the coordinates (P_{pi}) of the point cloud centroid in the pineapple coordinate system.
5) The distribution of point cloud centroids (P_{pi}) is transformed from the pineapple coordinate system to the radian-height coordinate system ($C_{pi(\theta-z)}$), where the radian is the horizontal axis ($\theta - axis$) and the $z - axis$ is the vertical axis, as shown in Figure 7. It can be seen that the distribution of the pineapple eyes in this coordinate system is quite regular, with the pineapple eyes on the same spiral line essentially aligned in a straight line. Therefore, we thought of using clustering to distinguish the pineapple eyes on each spiral line.

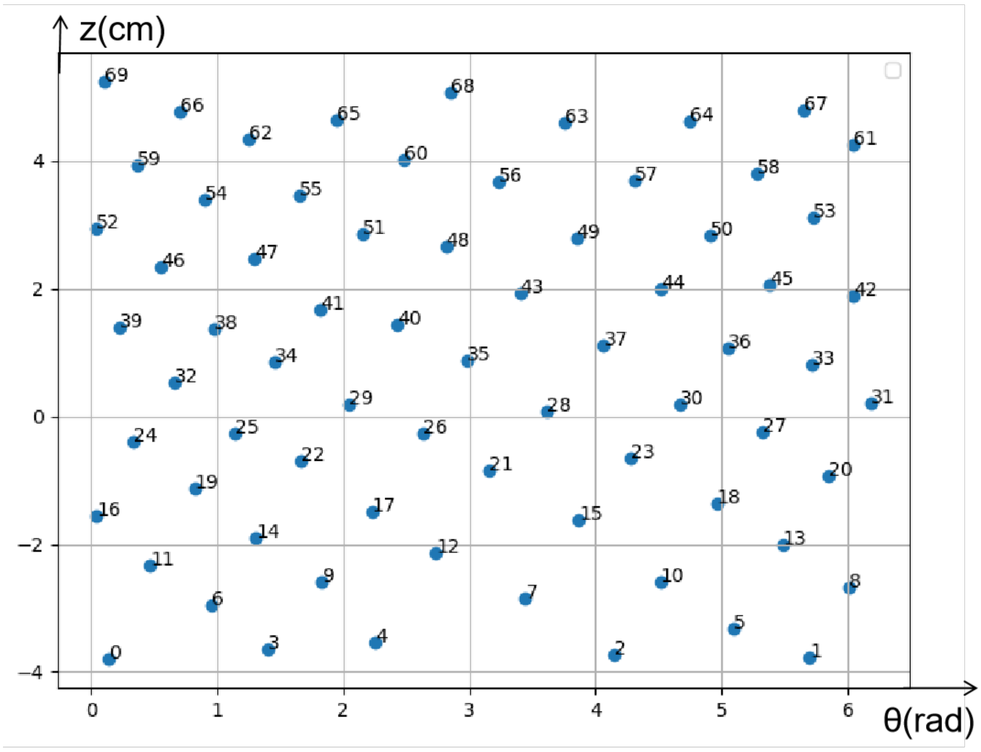

Figure 7. Display of pineapple eyes in the $\theta - z$ coordinate system.

6) Since the distribution radian of the pineapple eyes range from 0 to 2π, we first need to shift the pineapple eyes on the same spiral line before clustering. Assuming the slope of the straight line formed by the distribution of pineapple eyes is k, and the original coordinates of pineapple eyes are $C_{pi}(\theta_i, z_i)$, the shifted coordinates $C_i'(\theta_i', z_i')$ are:

$$\theta_i' = \begin{cases} \theta_i - 2\pi & \text{if } \theta_i - \frac{z_i - \min(z_i)}{k} > 2\pi \\ \theta_i + 2\pi & \text{if } \theta_i - \frac{z_i - \min(z_i)}{k} < 0 \\ \theta_i & \text{otherwise.} \end{cases} \tag{10}$$

Then we rotate the obtained C_i' by α and calculate the $cost$ value, α ranges from $\frac{\pi}{6}$ to $\frac{\pi}{3}$. The relationship between α and k is as follows:

$$k = -1.0 \cdot \tan(\alpha). \tag{11}$$

The coordinates after rotation, NC_i', are:

$$NC_i' = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot C_i'. \tag{12}$$

With each rotation, we perform clustering ($MyClusterAL(\cdot)$) to obtain 8 groups matrices NC_j' ($j \in \{1, \ldots, 8\}$). Each element $NC_{ij}'(\theta_{ij}, z_{ij})$ represents a pineapple eye, where j is the index of the spiral line (ranging from 1 to 8), and i is the number of pineapple eyes on each spiral line (ranging from 1 to Num_j). If $\theta_{ij} = 0$, then $NC_{ij}'(\theta_{ij}, z_{ij})$ can be represented as $NC_{ij}'(z_{ij})$. The cost function (L_{cost}) is designed to evaluate the quality of a given scheme by penalizing the variance of different variables. Specifically:

$$L_{cost} = k_1 Var(Num_j) + k_2 \frac{1}{8} \sum_{j=1}^{8} Var\left(\| NC'_{ij}(z) - \bar{NC}'_j(z) \| \right)$$
$$+ k_3 \frac{1}{8} \sum_{j=1}^{8} Var\left(\| NC'_{[i+1]j} - NC'_{[i]j} \| \right), (i \in (1,2,\ldots,Num_j), j \in (1,2,\ldots,8)), \quad (13)$$

where, k_1, k_2 and k_3 are hyperparameters that adjust the weight of each term in the overall cost function. $Var(\cdot)$ represents the variance calculation. The calculation of $\bar{NC}'_j(z)$ is as follows:

$$\bar{NC}'_j(z) = \frac{1}{Num_j} \sum_{i=1}^{Num_j} NC'_{ij}(z). \quad (14)$$

7) In the mechanism coordinate system, sort the pineapple eyes on each helical line according to their Z-axis values to obtain an ordered set of pineapple eyes for each helical line. Connect the pineapple eyes on each helical line with line segments.

To enhance processing efficiency, we also designed a cooperative dual-blade eye removal mechanism tailored to the specific characteristics, as illustrated in Figure 8. This mechanism comprises symmetric left and right eye-cutting devices, each equipped with three degrees of freedom, including the eye cutter, eye cutter rotary motor, eye cutter feed motor, eye cutter translation motor, and other essential components. Collaborating with the clamping platform's pineapple rotation axis, they simultaneously cut two lines, repeat the process twice, and successfully remove the pineapple eyes along all eight spirals.

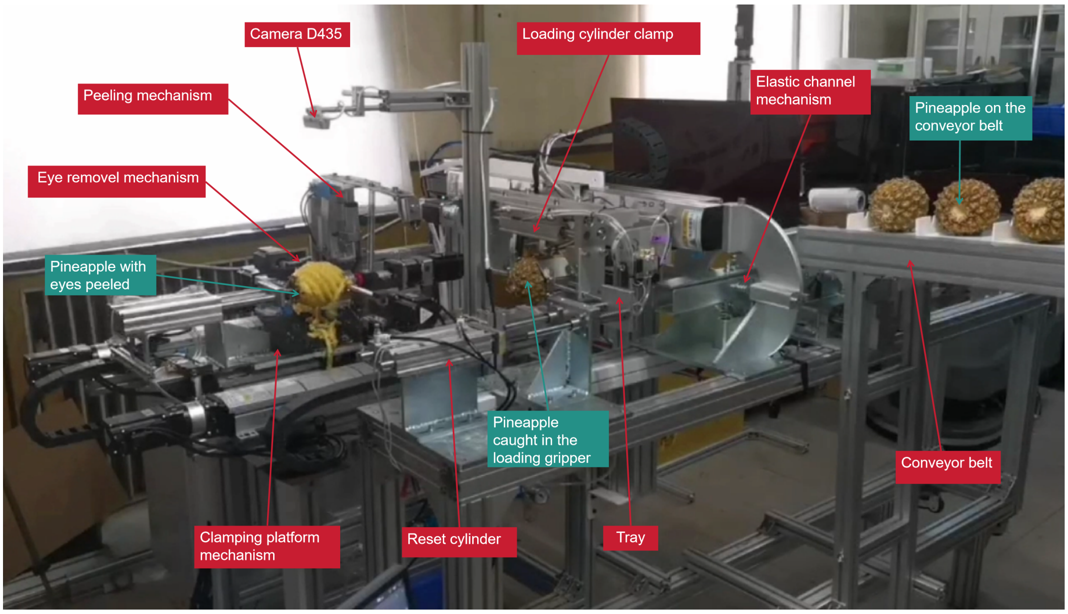
Camera D435
Loading cylinder clamp
Peeling mechanism
Elastic channel mechanism
Pineapple on the conveyor belt
Eye removel mechanism
Pineapple with eyes peeled
Pineapple caught in the loading gripper
Conveyor belt
Clamping platform mechanism
Reset cylinder
Tray

Figure 8. The vision-based pineapple automated processing system we constructed
.

3. Results and Discussion

To validate the effectiveness of the automated processing system, we constructed a prototype of the pineapple automated processing system, as shown in Figure 8. It is comprised automatic feeding mechanism (including conveyor feeding and pose correction), automatic loading mechanism, peeling mechanism, RGB-D camera (D435), eye removal mechanism, and a computer (operating system: Ubuntu 20.04). This computer is equipped with 64GB of RAM, an Intel Core i9-10940X processor, and

doi:10.20944/preprints202408.0699.v1

an NVIDIA GTX3090 graphics processor. The camera has an RGB resolution of 640*480 pixels and operates at a frame rate of 30 fps. Different modules work in a predefined sequence, enabling the entire automated processing process. In the experiments, we conducted separate tests and statistical analyses of each module and then verified the feasibility and stability of the entire automated processing system.

3.1. Result Analysis of Pineapple Eyes Detection

In the system we constructed, the camera was installed directly above the pineapple at a distance of 35mm from the rotation axis and calibrated with the mechanical setup. The computer is responsible for orchestrating the system's overall control, which encompasses the regulation of individual components or subsystems, the implementation of deep vision algorithms, as well as the deployment of cutting path and trajectory planning algorithms. This visual system is a core component of our automated processing system, responsible for identifying the position and shape of pineapple eyes.

In the actual processing environment, we processed 100 pineapples using the automatic peeling mechanism, resulting in a total of 600 pineapple images collected under varying lighting conditions. To increase data diversity and enhance the robustness of the deep vision network, we applied data augmentation techniques. Ultimately, we obtained a dataset comprising 3000 pineapple images. We used the Labelme software to perform precise annotations of pineapples and pineapple eyes. The dataset was divided into training, validation, and testing sets in an 8:1:1 ratio, which were used for training, validating, and testing the deep vision network and point cloud reconstruction.

In the experiments, we compared the performance of various deep learning models, including Mask-RCNN, BlendMask, and our enhanced SAAF-BlendMask algorithm, using the pineapple dataset established earlier. Additionally, extensive experiments were conducted to explore different values of the hyperparameters m and n. The experimental results, as depicted in Table 1 and Table 2, highlight the exceptional performance of our enhanced algorithm. Firstly, it is evident that the inference time of the BlendMask algorithm and our improved BlendMask algorithm are closely comparable, significantly outperforming the Mask-RCNN algorithm by approximately 62% in inference time. Moreover, compared to the Mask-RCNN algorithm, our enhanced BlendMask algorithm demonstrated a higher AP50. As shown in Table 2, when $m = 1$ and $n = 30$, the AP for Mask-RCNN improved by nearly 0.2 percentage points, and for BlendMask it improved by nearly **2.78** percentage points, reaching **73.04%**. For the recognition metric focused on small objects, APs, the enhanced BlendMask algorithm outperformed BlendMask by nearly **6** percentage points , reaching **62.54%**. Overall, these findings underscore the efficacy of our enhancements over existing algorithms.

Table 1. EVALUATION RESULTS FOR SEGMENTATION (SEGM) WITH SAAF-BLENDMASK.

m	n	AP	AP50	AP75	APs	APl	infer time(ms)
1	20	72.34	94.17	82.31	62.05	83.19	0.020311
10	20	72.33	94.48	80.68	60.72	83.94	0.020599
20	20	74.27	94.95	82.47	62.44	86.11	0.020481
30	20	73.37	94.96	82.71	61.82	84.92	0.01985
1	30	73.04	95.44	83.62	62.54	83.56	0.020537
10	30	72.10	92.25	82.87	62.35	82.25	0.020342
20	30	70.83	92.78	81.74	62.69	79.00	0.020744
30	30	70.57	91.86	81.29	62.56	78.24	0.020971

Table 2. EVALUATION RESULTS FOR SEGMENTATION (SEGM) WITH DIFFERENT METHODS.

Methods	AP	AP50	AP75	APs	APl	infer time(ms)
Mask-RCNN	72.85	92.69	81.77	62.29	83.42	0.053034
BlendMask	70.26	94.41	77.59	56.55	83.96	0.019867
SAAF-BlendMask(Ours)	**73.04**	**95.44**	**83.62**	**62.54**	**83.56**	0.020537

Following pineapple eye segmentation on the RGB images and alignment with the depth images, we extract the corresponding pineapple eye regions from the depth images and convert them into point clouds. During the processing workflow, we capture six images uniformly for each pineapple. We repeat the aforementioned procedure, performing point cloud fusion, denoising, and filtering operations to reconstruct a complete pineapple eye point cloud in the equipment coordinate system, as depicted in Figure 5.

3.2. Result Analysis of Pineapple Eyes Cutting Path Planning

To validate the effectiveness of the PCP method, we conducted a comparative experiment using 100 pineapples. After peeling the pineapples with a machine and reconstructing the pineapple eye point clouds, we planned the cutting path. Initially, the PCP method was not applied, and the results, as shown in Figure 9 (a) and (b), depict the distribution of pineapple eyes in the machine coordinate system. Subsequently, the same pineapples underwent cutting path planning using the PCP method, with the pineapple eyes distributed along eight helical lines in the pineapple's coordinate system, as illustrated in Figure 9 (c) and (d). It is clear that the path planned using the PCP method perfectly forms eight spiral lines, achieving the desired effect. This method is more suitable for dual-blade coordinated cutting. The results clearly indicate that path planning with the PCP method is more reliable. The experimental outcomes reveal that 75 pineapples were successfully planned without the PCP method, whereas 99 pineapples were successfully planned with the PCP method.

Following the distribution of pineapple eyes along each spiral, we proceed to plan the cutting trajectories. As outlined in the design of the eye removal apparatus in the preceding sections, we understand that during eye removal, the left and right symmetric eye removal mechanisms collectively exhibit six axes that coordinate with the pineapple's rotation axis. Using the pineapple rotation axis as the reference, we employ cubic spline interpolation three times for the remaining six axes to obtain the motion trajectories or interpolation data for the eye removal.

During the eye removal process, the computer dynamically controls the synchronized operation of the symmetrical eye removal mechanisms in real-time based on interpolation data, achieving pineapple eye removal in a manner that emulates human dexterity (as shown in Figure 1 (b)). In the experiments, we selected 110 pineapples for cutting, divided into four groups based on weight. Initially, we conducted a cutting experiment using our method. After weighing, the peeled pineapples were further processed to achieve the same peeling and eye removal results as traditional methods. The experimental data are presented in Table 3. Our method of pineapple peeling has demonstrated a significant advantage over the traditional method in terms of minimizing waste. Specifically, the total peeling waste using our method was 40.78 kg, compared to 46.18 kg with the traditional method. This indicates that our method preserved approximately 5.40 kg more pineapple flesh, representing a reduction in waste of about **11.7%**.

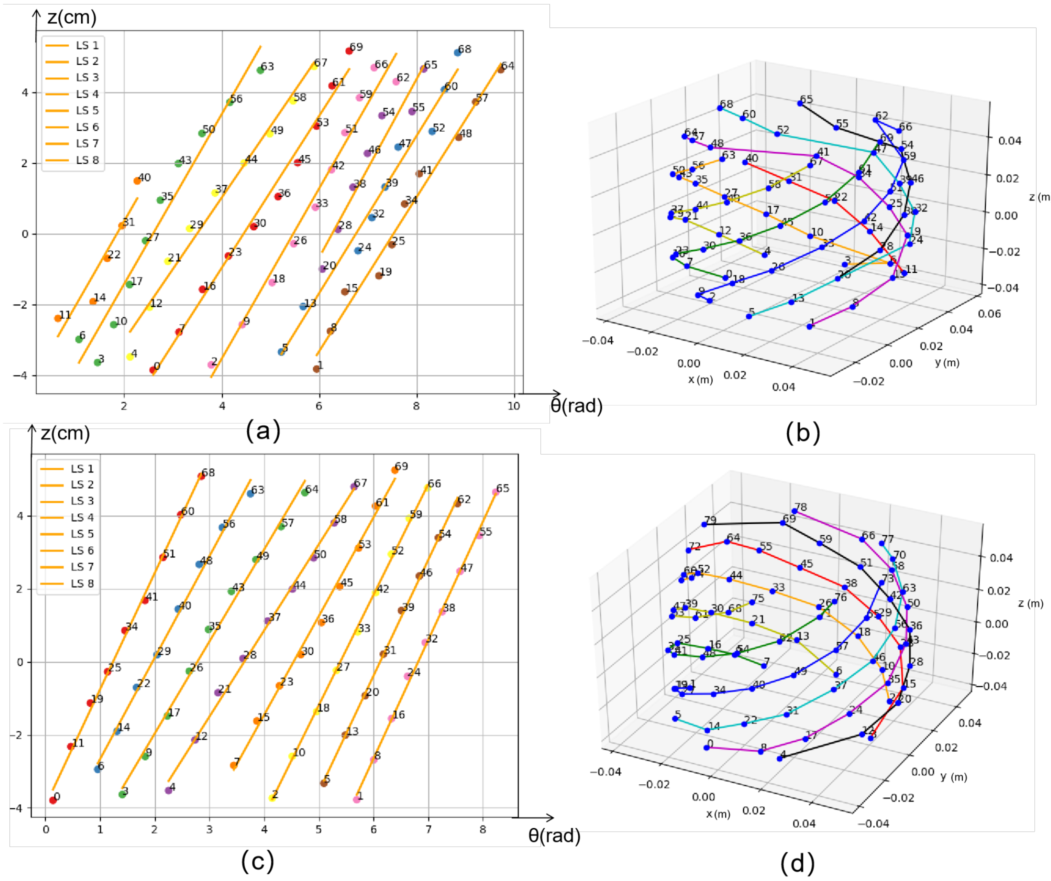

Figure 9. Pineapple eye cutting path planning results. The cutting path consists of 8 lines covering the pineapple's surface, and each point represents the centroid of the pineapple eye point cloud with numbering. (a) Path planning results for pineapple eye cutting in the $\theta - z$ coordinate system without using the posture correction algorithm. (b) Path planning results for pineapple eye cutting in the Cartesian coordinate system without using the posture correction algorithm. (c) Path planning results for pineapple eye cutting in the $\theta - z$ coordinate system with using the posture correction algorithm. (d) Path planning results for pineapple eye cutting in the Cartesian coordinate system with using the posture correction algorithm.

Table 3. Data on Pineapple Cutting and Peeling with DIFFERENT METHODS

Weight Group (kg)	< 1.2	1.2 - 1.7	1.7 - 2.2	> 2.2	Total
Number of Pineapples	22	62	19	7	110
Total Weight (kg)	22.65	92.51	37.41	17.16	169.73
Peeling Waste with Our Method (kg)	5.68	22.34	8.79	3.97	40.78
Peeling Waste with Traditional Method (kg)	6.34	24.92	10.14	4.78	46.18

* This table shows the data on pineapple cutting and peeling using different methods. The waste data is calculated based on different peeling techniques.

3.3. Comparison of Different Pineapple Processing Methods

We also conducted extensive research in the specific domain of automated pineapple processing. Table 4 compares some of the most popular reported processing methods. Clearly, through comparison, the proposed approach demonstrates more advantages than all methods. The column Integrated System addresses the question of whether a fully automated electromechanical integrated system is provided to address this issue. This paper also compares the use of machine learning model integration. It can be inferred that the proposed system not only fairly addresses the initial problem but also leverages and promotes the use of cutting-edge technology to ensure its relevance in the near future.

Table 4. COMPARATIVE ANALYSIS OF THE PROPOSED FRAMEWORK WITH THE MOST MAINSTREAM TECHNIQUES.

Methods	IS	LLC	LFW	EP	AIM	EM
Full Manual Processing [22,24]	×	×	×	×	×	✓
Automatic Processing [32]	✓	✓	×	✓	×	✓
Semi-auto Peeling[25]	×	×	×	✓	×	×
Spiral Peeling Processing [26]	×	×	✓	×	×	×
Visual Recognition Pineapple Eyes[33]	×	×	×	×	✓	×
Ours	✓	✓	✓	✓	✓	✓

* Abbreviations: IS (Integrated System), LLC (Low Labor Costs), LFW (Low Flesh Wastage), EP (Efficient Processing), AIM (AI-based Method), EM (Ensemble Methods).

4. Conclusion

In conclusion, the developed algorithms, notably the Scale Adaptive Adjustment Factor enhanced BlendMask algorithm and the PCP method, have demonstrated remarkable efficacy in addressing the challenges of precision pineapple processing. Through meticulous experimentation and analysis, these algorithms have showcased superior performance compared to conventional methods, particularly in pineapple eye recognition and path planning accuracy.

The successful integration of these advanced algorithms into the automated processing system represents a significant milestone in pineapple processing technology. The developed system, equipped with 3D vision and deep learning capabilities, offers a comprehensive solution for improving processing efficiency and reducing fruit wastage in industrial settings.

The experimental results validate the effectiveness of the system, highlighting its robustness, reliability, and scalability in real-world applications. By streamlining processing workflows and minimizing fruit wastage, the system not only enhances productivity but also contributes to resource conservation and sustainability in the food industry.

In conclusion, the combination of innovative algorithms and advanced system development represents a transformative approach to pineapple processing. By harnessing the power of cutting-edge technologies, this research paves the way for more efficient, sustainable, and economically viable agricultural product processing systems.

Author Contributions: Conceptualization, H.W., H.Z., and J.D.; methodology, H.W., H.Z., and J.D.; software, H.Z., C.L., and Y.Z.; validation, H.W., J.D. and J.T.; formal analysis, Y.Z., and J.T.; investigation, H.W.; resources, J.T.; data curation, Y.Z., and C.L.; writing—original draft preparation, H.W.; writing—review and editing, H.Z., and C.L.; visualization, Y.Z., and J.D.; supervision, C.L.; project administration, J.T.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Funding: This research was funded by the Chinese Academy of Sciences through the STS Program, grant number STS-HP-202207.

Data Availability Statement: All data are presented in this article in the form of figures and tables.

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

References

1. Mohsin, A.; Jabeen, A.; Majid, D.; Allai, F.M.; Dar, A.H.; Gulzar, B.; Makroo, H.A. Pineapple. *Antioxidants in fruits: Properties and health benefits*, **2020**, 379-396.
2. Rohrbach, K.G.; Leal, F.; d'Eeckenbrugge G.C. History, distribution and world production. *In The pineapple: botany, production and uses*, **2003**, 1-12.
3. Jia, W.; Zhang, Z.; Shao, W.; Hou, S.; Ji, Z.; Liu, G.; Yin, X. Foveamask: A fast and accurate deep learning model for green fruit instance segmentation, *Computers and Electronics in Agriculture*, **2021**, *191*, 106488.
4. Kasinathan, T.; Singaraju, D.; Uyyala, S.R. Insect classification and detection in field crops using modern machine learning techniques, *Information Processing in Agriculture*, **2021**, *8*, 446–457.
5. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep learning–method overview and review of use for fruit detection and yield estimation, *Computers and electronics in agriculture*, **2019**, *162*, 219–234.
6. Indira, D.; Goddu, J.; Indraja, B.; Challa, V.M.L.; Manasa, B.A. review on fruit recognition and feature evaluation using cnn, *Materials Today: Proceedings*, **2023**, *80*, 3438–3443.
7. Shakya, D.S. Analysis of artificial intelligence based image classifica- tion techniques, *Journal of Innovative Image Processing*, **2020**, *2*, 44–54.
8. Zhang, W.; Zhao, D.; Gong, W.; Li, Z.; Lu, Q.; Yang, S. Food image recognition with convolutional neural networks, in Proceedings of the 2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing and 2015 IEEE 12th International Conference on Autonomic and Trusted Computing and 2015 IEEE 15th International Conference on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom),Beijing, China, 10-14 August 2015; pp.690–693.
9. Matsuda, Y.; Hoashi, H.; Yanai, K. Recognition of multiple-food images by detecting candidate regions, in Proceedings of the 2012 IEEE International Conference on Multimedia and Expo, Melbourne, Australia, 9-13 July 2012; pp.25–30.
10. Liu, G.; Nouaze, J.C.; Touko Mbouembe, P.L.; Kim, J.H. Yolo-tomato: A robust algorithm for tomato detection based on yolov3, *Sensors*, **2020**, *20*, 2145.
11. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement, *arXiv preprint arXiv:1804.02767*, **2018**.
12. Yu, Y.; Zhang K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on mask-rcnn, *Computers and Electronics in Agriculture*, **2019**, *163*, 104846.
13. He, K.; Gkioxari, G.; Doll´ar, P.; Girshick, R. Mask r-cnn, in Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22-29 October 2017; pp.2961–2969.
14. Wang, X.A.; Tang, J; Whitty, M. Deepphenology: Estimation of apple flower phenology distributions based on deep learning, *Computers and Electronics in Agriculture*, **2021**, *185*, 106123.
15. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, **2014**.
16. Chen, S.W.; Shivakumar, S.S.; Dcunha, S.; Das, J.; Okon, E.; Qu C.; Taylor, C.J.; Kumar, V. Counting apples and oranges with deep learning: A data-driven approach, *IEEE Robotics and Automation Letters*, **2017**, *2*, 781–788.
17. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for small object detection," *arXiv preprint arXiv:1902.07296*, **2019**.
18. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection, *arXiv preprint arXiv:2004.10934*, **2020**.
19. Haris, M.; Shakhnarovich, G.; Ukita, N. Task-driven super resolution: Object detection in low-resolution images, in Proceeding of the 28th International Conference on Neural Information Processing, Sanur, Bali, Indonesia, 8–12 December 2021; pp.387–395.
20. Bai, Y.; Zhang, Y.; Ding, M.; Ghanem, B. Sod-mtgan: Small object detection via multi-task generative adversarial network, in Proceedings of the European Conference on Computer Vision, Munich, Germany, 8-14 September 2018; pp.206–221.
21. Lin, T.-Y.; Doll´ar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21-26 July 2017; pp.2117–2125.
22. Singh, V.; Verma, D.K.; Singh, G. Development of pineapple peeler- cum-slicer, *Pop. Kheti*, **2013**, *1*, 21–24.

23. Kumar, P.; Chakrabarti, D.; Patel, T.; Chowdhuri, A. Work-related pains among the workers associated with pineapple peeling in small fruit processing units of north east india, *International Journal of Industrial Ergonomics*, **2016**, *53*, 124–129.
24. Anjali, A.; Anjitha, P.; Neethu, K.; Arunkrishnan, A.; Vahid, P.A.; Mathew, S. M. Development and performance evaluation of a semi- automatic pineapple peeling machine, *International Journal of Current Microbiology and Applied Sciences*, **2019**, *8*, 325–332.
25. Jongyingcharoen, J.S.; Cheevitsopon, E. Design and development of continuous pineapple-peeling machine," Agriculture and Natural Resources, **2022**, *56*, 979–986.
26. Siriwardhana, P.G.A.L.; Wijewardane, D.C. Machine for the pineapple peeling process, *Journal of Engineering and Technology of the Open University of Sri Lanka (JET-OUSL)*, **2018**, *6*, 1-15.
27. Kumar, P.; Chakrabartiand, D. Design of pineapple peeling equipment, in Proceedings of the IEEE 6th International Conference on Advanced Production and Industrial Engineering, Delhi, India, 18-19 June 2021; pp.545–556.
28. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one- stage object detection, in Proceedings of the International Conference on Computer Vision, Seoul, Korea (South), 27 October-2 November 2019; pp.9627–9636.
29. Chen, H.; Sun, K.; Tian, Z.; Shen, C.; Huang, Y.; Yan, Y. Blendmask: Top-down meets bottom-up for instance segmentation, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13-19 June, 2020; pp.8573–8581.
30. Ekern, P.C. Phyllotaxy of pineapple plant and fruit, *Botanical Gazette*, **1968**, *129*, 92–94.
31. Hikal, W.M.; Mahmoud, A.A.; Said-Al Ahl, H.A.; Bratovcic, A.; Tkachenko, K.G.; Ka˘c´aniov´a, M.; Rodriguez, R.M. Pineapple (ananas comosus l. merr.), waste streams, characterisation and valorisation: An overview, *Open Journal of Ecology*, **2021**, *11*, 610–634.
32. Bartholomew, D.P.; Hawkins. R.A.; Lopez. J.A. Hawaii pineapple: the rise and fall of an industry, *HortScience*, **2012**, *47*, 1390– 1398.
33. Liu. A.; Xiang. Y.; Li. Y.; Hu. Z.; Dai. X.; Lei. X.; Tang. Z. 3d positioning method for pineapple eyes based on multiangle image stereo- matching, *Agriculture*, **2022**, *12*, 2039.

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.