

Research Article

Vol. 15, No. 3, 2025, p. 379-395

Sweet Pepper Detection Using Fast Point Features Histogram and Unsupervised Learning

O. Doosti-Irani^{1*}, M. H. Aghkhani¹, M. R. Golzarian¹

1- Department of Biosystems Engineering, Faculty of Agriculture, Ferdowsi University of Mashhad, Mashhad, Iran

(*- Corresponding Author Email: omiddoostiirani@gmail.com)

Received: 21 June 2023

Revised: 05 August 2023

Accepted: 16 August 2023

Available Online: 11 May 2025

How to cite this article:Doosti-Irani, O., Aghkhani, M. H., & Golzarian, M. R. (2025). Sweet Pepper Detection Using Fast Point Features Histogram and Unsupervised Learning. *Journal of Agricultural Machinery*, 15(3), 379-395. <https://doi.org/10.22067/jam.2023.83054.1174>

Abstract

Robotic harvesting in agriculture is an effective method for producing healthy fruit, reducing costs, and increasing productivity. Detecting and harvesting sweet peppers, however, remains a challenging task. This study aims to develop an unsupervised machine vision algorithm to recognize colored sweet peppers using a combination of geometric features (Fast Point Feature Histogram- FPFH) and color features (H, S, and V). Depth images were captured using a Kinect v2 sensor, and a 3D model was reconstructed. After extracting the geometric and color features, data preprocessing involved undersampling to ensure balance and applying the Z-score criterion to eliminate outliers. Principal component analysis (PCA) was used to reduce the feature dimensions, and the K-means clustering model was implemented to categorize the data using six geometric features and three color features. The silhouette coefficient was employed to evaluate clustering quality, and human evaluation demonstrated that the algorithm achieved a detection accuracy of 95.10% for sweet peppers.

Keywords: Fast point feature histogram, Fruit detection, Machine vision, Sweet pepper, Unsupervised learning

Introduction

Agriculture is increasingly moving towards automation, which requires smarter frameworks and technologies (Tang *et al.*, 2020). The use of agricultural robots has become a popular topic in farming, alongside emerging ideas like digital and intelligent agriculture (Ball *et al.*, 2016). Manual fruit harvesting poses risks such as injuries, falls from heights, bee stings, and other hazards, in addition to the high labor cost. However,

robotic harvesting still remains a challenge (Shamshiri, Hameed, Karkee, & Weltzien, 2018). Challenges in robotic harvesting include high-precision fruit recognition, building robots with high speed and accuracy, and the ability to harvest while preserving fruit quality. Smart agriculture uses AI technologies like machine vision and robots for decision-making in real-time (Chidambaranathan, Handa, Ramanamurthy, & Ramanamurthy, 2018; Dharmaraj, & Vijayanand, 2018). Accurate fruit recognition by machines is affected by various environmental factors including light, canopy structure, fruit color, occlusion, plant care, fruit maturity, and leaf density (Gongal, Amatya, Karkee, Zhang, & Lewis, 2015). RGB cameras are commonly



©2025 The author(s). This is an open access article distributed under Creative Commons Attribution 4.0 International License (CC BY 4.0).

<https://doi.org/10.22067/jam.2023.83054.1174>

used for fruit harvesting in robots, as they can capture images in three channels simultaneously and extract color, geometric, and texture features (Kurtulmus, Lee, & Vardar, 2014; Hemming, Bac, & Van Tuijl, 2011; Fu, Majeed, Zhang, Karkee, & Zhang, 2020). Kurtulmus *et al.* (2014) used RGB color imaging to detect green immature fruits under natural illumination conditions in their study. The green color of peaches was identified as a major challenge. The maximum accuracy of detecting immature fruits obtained in this study was 84.6%, and the processing time was in the range between 72.8 and 112 s for individual images in the validation set. Sa *et al.* (2016) used deep convolutional neural networks to develop an accurate and efficient fruit detection system. However, this method requires a large number of images for model training and significant time for labeling these images. Previous studies indicate that RGB cameras struggle to perform effectively and yield satisfactory results when the color of the fruit closely resembles that of the background. Stein, Bargoti, and Underwood (2016), proposed a framework based on multi-sensor for identifying mango fruit. The aim of this research was to estimate the farm yield using a 3D model of each tree. A total of 522 trees were captured in the images. The R-CNN method has been used for object detection. The error rate obtained for each tree was 1.36%. In this, they address the problem of detecting fruits enclosed by leaves. To detect immature green citrus, Gan, Lee, Alchanatis, Ehsani, and Schueller (2018) proposed a new algorithm based on the combination of color and thermal features to distinguish green fruit from the background. The detection accuracy has increased from 86.6% to 95.5% by combining color and thermal images. But sensors such as Lidar and thermal imaging are costly; therefore, utilizing these sensors is often impractical for fruit detection. In research Mohamad zamani, Javidan, Zand, and Rasouli (2023) developed a deep neural network for cucumber detection. Testing this method on 120 samples showed that the accuracy of the network in correctly identifying the position of

the cucumber fruit in the images was 95.3%.

Depth cameras have become increasingly popular among researchers in recent years. These cameras can capture both color and depth images of the target simultaneously, providing information about the target's position. This allows for the computation of three-dimensional geometric features of objects, which is useful for detecting fruits. In one study, depth cameras have been used in combination with RGB cameras to detect repetitious apples and achieve a precision of 87.0% (Gongal *et al.*, 2016). In another research, an algorithm was developed to detect apples on trees using an RGB-D camera and clustering with Euclidean distance (Nguyen, Vandevoorde, Kayacan, De Baerdemaeker, & Saeys, 2014). Point cloud data was used to detect apples with different 3D descriptors and classification methods, achieving acceptable results (Tao & Zhou, 2017). A depth camera was also used to detect sweet peppers and their pedicels, achieving a reported area under the curve of 0.71% with an acceptable accuracy (Sa *et al.*, 2017). In a different study, a dense point cloud of crops was created using an RGB-D camera to recognize red sweet peppers, and the algorithm accuracy was 90.69% (Zhao *et al.*, 2020). Researchers also evaluated approach strategies for sweet pepper harvesting robots. (Ringdahl, Kurtser, & Edan, 2019). Moghimi, Aghkhani, and Goltzarian (2015) used RGB images to detect green sweet peppers. By utilizing texture features, the accuracy of detection reached 86%. Moreover, by combining texture features with color features, the accuracy increased to 92%. Doosti-Irani, Goltzarian, and Aghkhani (2023) used 3D geometric features and K-NN classification algorithm to detect green sweet peppers. The F1-score of the model was 82.85%, and the accuracy of detection based on human evaluation was 83.07%. Ning *et al.* (2022) proposed a new algorithm named AYDY, which achieved a 9.14% improvement in F1-score compared to YOLO-V4. The algorithm demonstrated high accuracy in sweet pepper detection and localization, with an average localization accuracy of

89.55% and a collision-free harvesting success rate of 90.04%.

While various sensors (e. g. 2D and 3D) and descriptors (e. g. color, texture, and geometric) have been used to detect fruits and crops, several challenges remain, mainly due to changing environmental conditions and plant growth (Javidan, Banakar, Vakilian, & Ampatz, 2023). Sweet pepper detection and harvesting, in particular, pose a lot of difficulties such as complex and non-uniform backgrounds due to varying illumination intensity, fruits and pedicels covered by foliage, the fruits' color variations throughout their growth phases, sunburn, blotchy patches, and color fade. Therefore, relying solely on color characteristics to detect products is not always efficient or accurate. To tackle these challenges, it is essential to consider other information beyond color, such as three-dimensional geometric features, which can be

easily obtained and to a great extent address the challenges of color images (Mohammadi, Massah, & Asefpour Vakilian, 2023). The data for this study were collected in the summer season of 2021, using a Kinect v2 sensor in a greenhouse located in Mohsen Abad, a suburb of Mashhad, Iran. Three-dimensional models of point clouds were then created and used to preprocess information and extract geometric features. Finally, unsupervised clustering was employed to detect colored sweet peppers. Figure 1 provides a summary of the objectives of this research. The main objective of this study is to present a method for real-time detection of sweet peppers in a greenhouse environment based on 3D models. The innovation of this study lies in using a new method for noise reduction in depth images and employing a combination of geometric and color features to cluster 3D models and recognize sweet peppers.

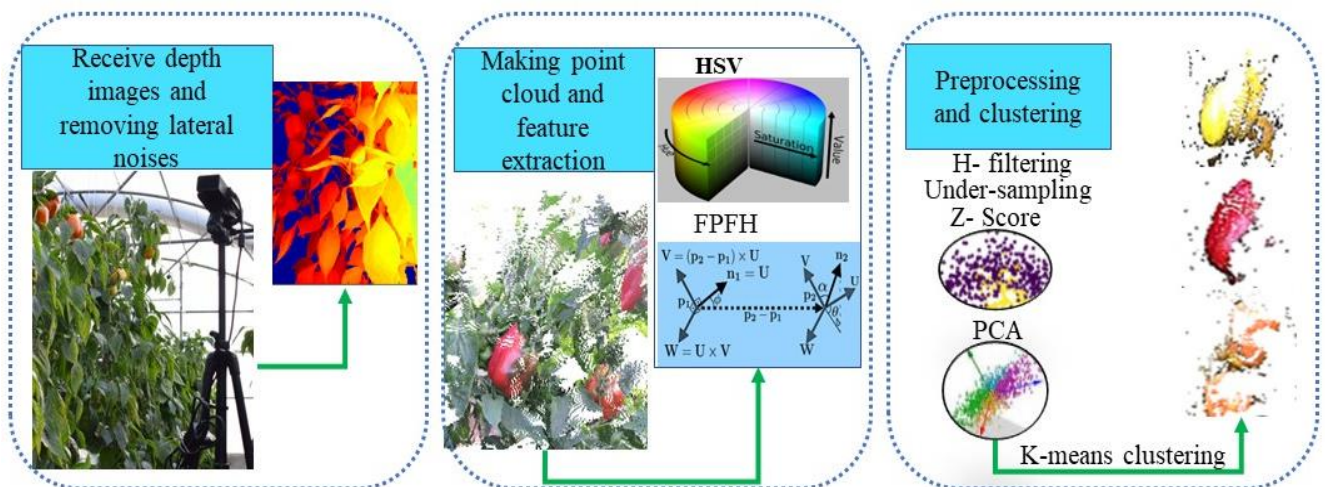


Fig.1. The summary of research steps

Materials and Methods

Point cloud obtainment

The Kinect v2 sensor was used to acquire depth and color images of sweet peppers in a greenhouse environment. After installing the related drivers for the Kinect v2 sensor in MATLAB 2018a software (MathWorks Inc, US), the samples were imaged from a distance of 80 cm. 3-D models were developed using

depth maps, and color images were utilized for adding color to individual pixels in the 3-D models. The value of each pixel in this two-dimensional image indicates the distance from that point to the camera's center (Shen, Wu, & Suk, 2017). The characteristics of this sensor are shown in (Table 1).

Table 1- The characteristics of Kinect v2 sensor

Depth map resolution (pixels)	RGB map Resolution (pixels)	Operative measuring range (meter)	Filed of view (degrees)
512×424	1920×1080	0.5-4.5	70×60

Removing lateral noise

One of the challenges associated with depth images is the presence of pixels with zero values located at the edges of the image. The existence of the zero pixels in the depth image can affect the accuracy of the subsequent processing (Rusu, Blodow, & Beetz, 2009; Wan, Li, Jiang, & Xu, 2020). The noise present in the depth images was computed using Algorithm 1, which is based on the change in sign of the first derivative in the neighborhood of each point. In this algorithm, ΔZ represents the difference between the depth values of next pixels (Np) and previous pixels (Pp). By dividing ΔZ by ΔX , the first derivative of the depth function is obtained. If the sign of the first derivative is positive, the zero values of edge pixels are substituted with Np ; otherwise, with Pp .

Algorithm 1: Correcting zero pixels based on the first derivative of depth function:

```

%Input= X: A  $n \times m$  matrix contains zero
and nonzero values
%Output= X: A  $n \times m$  matrix contains
nonzero values
for  $i = 1:n$ 
for  $j = 2:m - 1$ 
if  $X(i,j) == 0$ 
     $Pp = X(i, j - 1)$ 
     $h = 0$ 
    for  $k = 1:3$ 
         $h = h + 1$ 
        if  $X(i, j + h) \neq 0$ 
            break
        end
    end
     $Np = X(i, j + h)$ 
     $\Delta z = (Np - Pp) / (h + 1)$ 

```

```

        if  $\Delta z > 0$ 
             $X(i, j: j + h - 1) = Np$ 
        else
             $X(i, j: j + h - 1) = Pp$ 
        end
    end
end
end

```

To demonstrate how this algorithm corrects the zero pixels, the algorithm was applied to a one-row depth map (a vector of pixels) (Figure 2(a)). The corrected depth vector is shown in Figure 2(b).

In order to validate the proposed algorithm, the Monte Carlo method was employed. In this method, different inputs are applied to the algorithm using randomly generated numbers, and the results obtained from each run of the algorithm are compared with each other. (Adams *et al.*, 2015). In order to assess the effectiveness of the proposed algorithm in substituting zero-valued points in depth maps, N points were randomly chosen from each depth map, which contained non-zero values, and then converted to zero. Next, the values of these converted points were estimated using the proposed algorithm and subsequently compared with the original non-zero values. The Root Mean Square Error (RMSE) index was computed to measure the accuracy of the algorithm. In this method, 50 images were utilized, and 20 samples were randomly selected from each image. In order to assess the performance of the proposed algorithm in estimating missing data, its RMSE value was compared with those of the commonly used B-fill and F-fill methods.

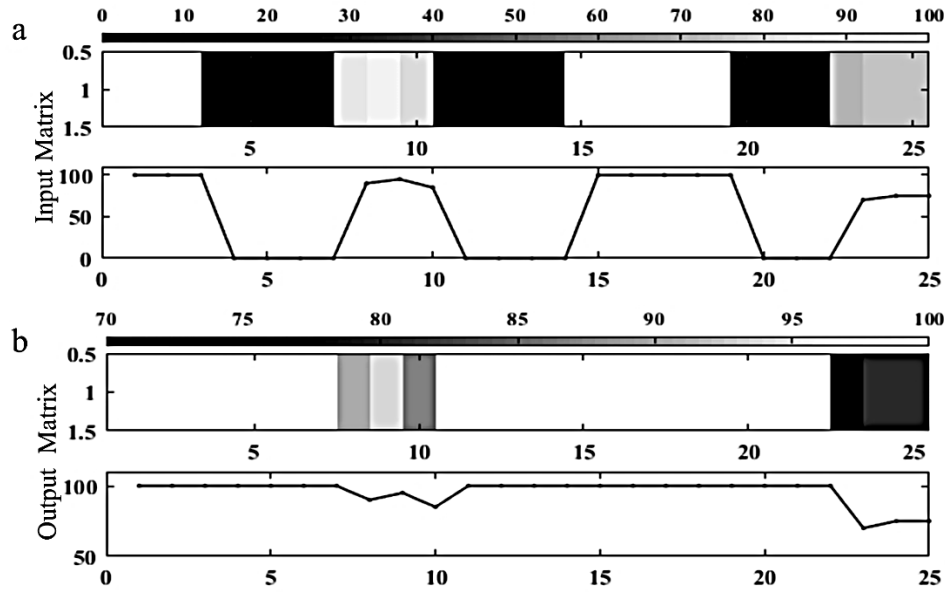


Fig.2. Simulation of first order derivative of depth function on a matrix: a) Input matrix, and b) Output matrix

Making the 3-D model

Calculate x and y

The positions x and y of the point $P(x, y, z)$ correspond to each pixel $p(j, k)$ in the depth map, as calculated using Eq. (1-2) (Lachat, Macher, Mittet, Landes, & Grussenmeyer, 2015):

$$x = \frac{j - Cx}{Fx} \times z \quad (1)$$

$$y = \frac{k - Cy}{Fy} \times z \quad (2)$$

where, (j, k) are parameters representing the position of pixel p in the depth map. In addition, (Cx, Cy) and (Fx, Fy) are internal parameters representing the coordinates of the focal point and focal length, respectively. Z represents the depth value for each pixel in the depth map. After calculating x and y for each pixel in the depth map, it is possible to have a point cloud.

3D Features extraction

In recent years, extensive research has been carried out on extracting descriptive 3D features. Local descriptors, such as PFH (Point Feature Histogram) (Rusu, Marton, Blodow, & Beetz, 2008) and FPFH (Fast Point Feature Histogram) (Rusu et al., 2009) which are extracted from the key points in the object's point cloud, are more suitable for instance

recognition and object classification. In this research, the FPFH descriptor was used to describe the surface of objects in the point cloud. The relative deviation between normal $n1$ and $n2$, which corresponded to two points $P1$ and $P2$, was calculated by defining a fixed local Darboux frame coordinate system (UVW) at one point, as shown in (Figure 3). A set of angle values was used to indicate the deviations between the two points using UVW coordinates, as shown in Eq. (3-6) (Han, Sun, Song, & Xiao, 2018; Sa et al., 2017).

$$\alpha = V \cdot n2 \quad (3)$$

$$\phi = U \cdot \frac{P2 - P1}{d} \quad (4)$$

$$\theta = \arctan(W \cdot n2, U \cdot n2) \quad (5)$$

$$d = \|p2 - p1\|_2 \quad (6)$$

where, d is the Euclidean distance between two points ($p1$ and $p2$). α , ϕ , and θ are descriptor angles in the UVW coordinate system. Furthermore, density was utilized as a geometric feature in the clustering process (Eq. 7). The coordinates of all point clouds are then considered as candidate cluster centers. Thus, each point p_i with coordinates (x_i, y_i, z_i) is potentially a cluster center whose density D_i is given by the following equation (Zhao et al., 2020):

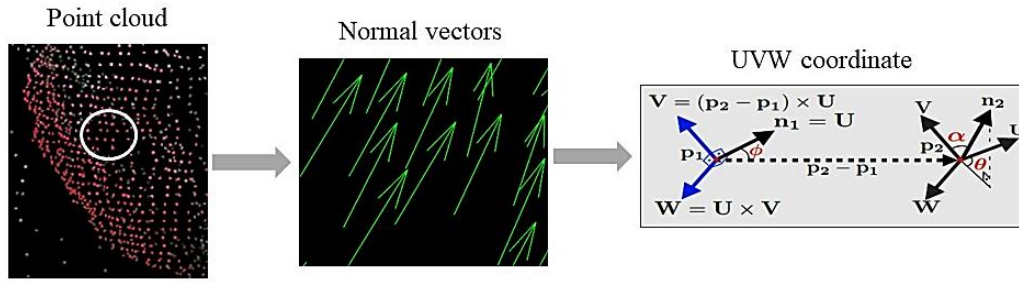


Fig.3. UVW Coordinates from a point cloud

$$D_i = \sum_{j=1}^N \exp \left(-\frac{(x_i - x_j)^2}{(\frac{R_{ax}}{2})^2} - \frac{(y_i - y_j)^2}{(\frac{R_{ay}}{2})^2} - \frac{(z_i - z_j)^2}{(\frac{R_{az}}{2})^2} \right) \quad (7)$$

where, d is the Euclidean distance between two points (p_1 and p_2). α , ϕ , and θ are descriptor angles in the UVW coordinate system. Furthermore, density was utilized as a geometric feature in the clustering process (Eq. 7). The coordinates of all point clouds are then considered as candidate cluster centers. Thus, each point p_i with coordinates (x_i, y_i, z_i) is potentially a cluster center whose density D_i is given by the following equation (Zhao *et al.*, 2020):

Color features extraction

In order to extract color features, the RGB color model was converted to the HSV model, and the three color components including H, S, and V were extracted. HSV color space was consistent with human eye color perception and showed fairly good light interference resistant capability. In general, the use of the HSV color space is useful in many areas of image processing because separate color features are easily accessible, which can help in object detection by providing more distinctive features.

Preprocessing

Data balancing with color filter

It is natural that simply using all the extracted values does not lead to a correct diagnosis. Therefore, it is necessary to conduct pre-processing on the data. Considering that the three-dimensional models were developed in greenhouse and under uncontrolled conditions, a wide range of data points across the X and Y axes has been obtained.

Therefore, the ratio of sweet pepper points is very low compared to the points related to branches, leaves, and the surrounding environment. Due to the imbalanced nature of the dataset, clustering may favor non-pepper data. To address this issue, the data were balanced using the sample reduction method and the under-sampling module in Python software. Therefore, the threshold values for color distinction between pepper and other parts of the plant were determined for orange pepper at $H > 0.13$ and for yellow pepper at $H > 0.17$ by checking the histogram of this color channel. Labeling and balancing were done based on this threshold value. After this step, the image background was removed by applying a depth filter of $Z < 120$ cm. Finally, the Z-Score criterion was used to remove outlier data that were far from the overall average in each feature. Using the Z-Score criterion, data points that fell outside the range $(X^- - 3\sigma, X^+ + 3\sigma)$ were identified as outlier data and removed.

Principal Component Analysis (PCA)

The number of extracted features is 36, which comprise three color features (H, S, and V) along with 33 geometric features. Due to the high dimensionality of the geometrical features, a PCA analysis was performed, and the number of dimensions of the geometrical features was reduced to five. Therefore, the final features for entering the clustering model include five PCA components, three components from the HSV color channel, and D_i .

Unsupervised Learning

Unsupervised *K*-means clustering method was used for clustering. The *K*-means algorithm is a popular clustering algorithm in unsupervised learning. Its goal is to divide input data into *k* different clusters so that the data in each cluster are similar to each other and distinct from the data in other clusters. The algorithm works by randomly initializing *k* cluster centroids, assigning each data point to the nearest centroid, and then computing the new centroid for each cluster as the mean of all data points assigned to it. This process is repeated until the centroids converge to a stable solution. Based on the research problem of detecting sweet peppers, which is the focus of this study, the number of clusters is considered to be two. In order to check the correctness of the clustering, the degree of cluster divergence was calculated using the silhouette coefficient (Rousseeuw, 1987). The silhouette coefficient ranges from -1 to 1, where a value closer to 1 indicates better clustering. A negative value suggests unsuitable clustering. To identify the pepper cluster, it is necessary to perform clustering first. By identifying the cluster, the 3D model

corresponding to it can be retrieved using a threshold value, *H*, which is defined and utilized. The resulting 3D image is shown in Figures 6 and 7. After forming a 3D model of the sweet pepper class and comparing it with the initial model, it underwent human evaluation and its accuracy is reported in Table 3.

Dataset

The dataset comprised 20 3D models, including five orange and yellow pepper models for algorithm development, with the remaining fifteen models reserved for algorithm evaluation.

Results and Discussion

Results of removing lateral noise

Based on the results in Table 2, both the lower and upper bounds for the mean of RMSEs were found to be smaller than those of the other two methods (B-fill and F-fill). Therefore, this method was confirmed to be superior to the other two in terms of detecting the embedded noisy data.

Table 2- Descriptive statistics of obtained RMSEs

	F-Fill	B-Fill	Proposed algorithm
Minimum	1.85	2.01	1.89
Maximum	27.18	24.61	4.85
Average	5.43	5.25	3.97
Standard deviation	2.58	2.24	0.66
First quartile	4.09	4.07	3.56
Second quartile(median)	4.89	4.82	4.11
Third quartile	5.84	5.75	4.52
Lower bound of 95 % CI	4.92	4.81	3.8
Upper bound of 95 % CI	5.94	5.69	4.10

The smaller standard deviation observed in the proposed method suggests higher reliability compared to the other two methods, as demonstrated in the box plot presented in Figure 4. Additionally, Figure 5 displays the RMSE values obtained by the three methods in different iterations, highlighting the differences among them.

After eliminating noise from the depth maps, it becomes possible to carry out further

processing on 3D models with fewer errors. By removing noise from depth images and replacing erroneous values with accurate ones, creating 3D models, object detection, background removal, and feature extraction can be performed with greater precision. In the context of object detection in 3D models, the geometric properties of points in a local neighborhood are utilized, and the presence of noise can significantly complicate the

estimation of surface properties like curvature and surface normals (Rusu & Cousins, 2011).

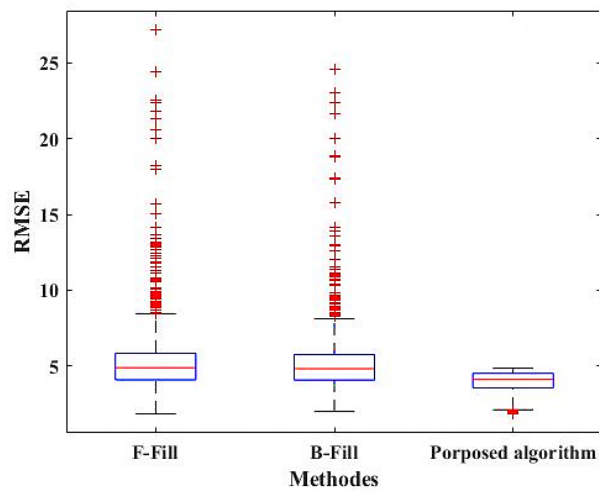


Fig.4. Results from the Monte Carlo method

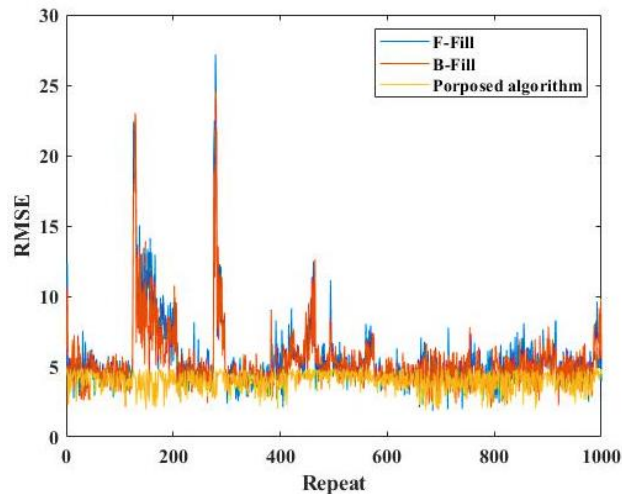


Fig.5. Root mean square error (RMSE) results from Monte Carlo simulations with 1,000 Iterations

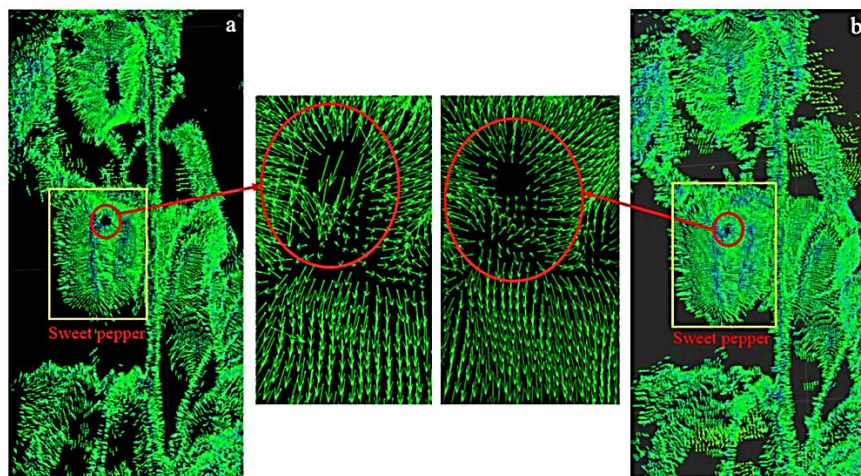


Fig.6. Demonstrate the normal vectors in 3D models: (a) before, and (b) after removing noises

The quality of extracted features such as PFH and FPFH (Sa *et al.*, 2017; Zhao *et al.*, 2020), SHOT (Muja, Rusu, Bradski, & Lowe, 2011), and VFH (Behley, Steinhage, & Cremers, 2012) is heavily dependent on the quality of the input data. In Zhao *et al.*'s (2020) study on detecting sweet peppers in a greenhouse environment, it was noted that the extraction of normal vectors is susceptible to noise from RGB-D sensors, leading to poor expression of surface curvature. Figure 6 displays two 3D models alongside their respective surface normal vectors. In Figure 6(b), the normal vectors in the 3D model are displayed after noise removal. As indicated, the normal vectors in Figure 6(b) appear more regular compared to those in Figure 6(a), which contains noisy points, and they more accurately illustrate changes in surface curvature for sweet peppers. Therefore, the methods presented in this study can be utilized for preprocessing depth data in the robotic harvesting of agricultural products. Hence, the techniques proposed in this study can be

employed for preprocessing depth data in robotic harvesting of agricultural products.

Result of preprocessing

Point cloud models consist of a large number of points, and using all of them for clustering may not result in optimal accuracy. To enhance clustering accuracy, data preprocessing was performed in this study. The Z-Score method was employed to eliminate outlier points, and data balancing was achieved by reducing samples based on the H color channel. Figure 7 illustrates the reduction in the number of prototype points achieved by balancing each model, indicating a significant decrease in the number of points for the balanced models compared to the initial models. In addition, the silhouette coefficient values displayed in Figure 7 demonstrate that including a higher number of points in the clustering model leads to lower clustering quality.

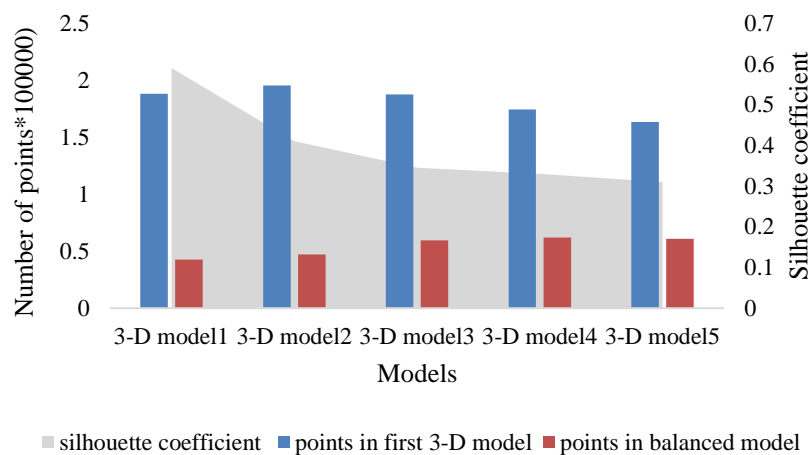


Fig.7. The reduction in points and silhouette coefficient for each model after dataset balancing

Results of PCA

The scree plot for PCA applied to the data is presented in Figure 8. The plot reveals that in this particular model, dimensions have been reduced to five components. This resulted in a considerably reduced slope and a minimal increase in the variance.

Results of K-means clustering

The K-means method was utilized as one of the unsupervised learning techniques in this study. The clustering quality was measured using the silhouette coefficient, and a human observer was employed to accurately identify

sweet peppers. In Figure 7, the silhouette coefficient plot is depicted for five models, with the maximum value equal to 0.59 and the minimum value equal to 0.31. These values indicate that the K-means clustering algorithm was able to effectively differentiate between the two clusters examined in this study (pepper and other plant components). Figure 9 illustrates the silhouette coefficient diagram

for a clustered model, where the horizontal axis represents the silhouette coefficient and the vertical axis shows the number of points in each cluster. The majority of the points on the graph fall within a range greater than zero, providing evidence for the successful clustering achieved using the K-means algorithm.

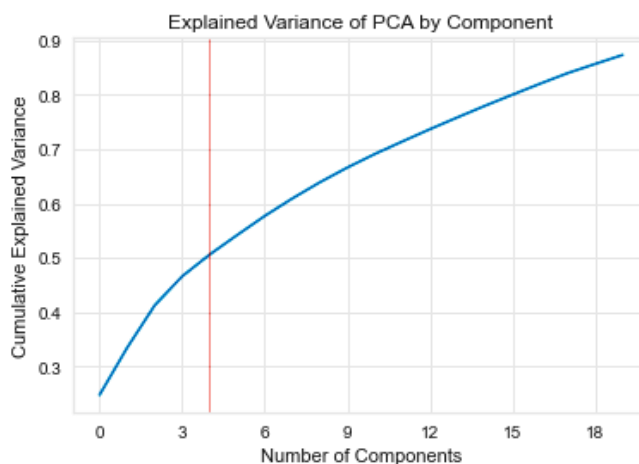


Fig.8. The graphic of the number of main components compared to the compressed variance is explained

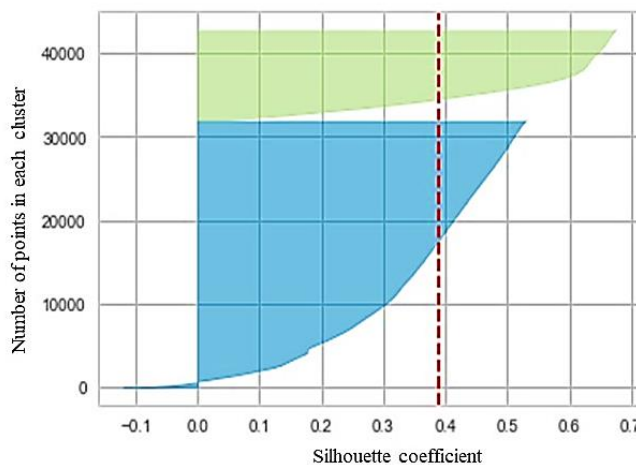


Fig.9. Silhouette analysis for K-means clustering on sample data with two cluster

To evaluate the performance of the proposed method in detecting sweet peppers, Figures 10 and 11 display the initial 3D model along with the number of detected sweet peppers. It is important to consider that the field of view for harvesting robots has limitations, and the harvesting operation can only be performed within a specific range.

Therefore, in this study, if the target is located at the margins of the image, it is not included in the final count. Figure 10 shows the results of pepper counting for two models of orange pepper, while Figure 11 shows the results for two models of yellow pepper. As demonstrated, the algorithm utilized in this study is capable of effectively differentiating

pepper points from those belonging to branches and leaves in the 3D point cloud model. In the examined 3D models, all available peppers were successfully identified. However, it should be noted that some points belonging to branches and leaves were also classified as peppers. This may lead to errors when a harvesting robot encounters these points, and it may mistakenly treat branches and leaves as peppers. Figure 12 demonstrates this issue. The presence of these misclassified points in the clustering process has resulted in

a decrease in the silhouette coefficient value. It is worth mentioning that the quality of clustering may vary across different models depending on various environmental factors, such as fruit-leaf overlap, lighting conditions, leaf density, maturity level, and so on. In this study, 3D models were created using the Kinect sensor without controlling these environmental conditions. Hence, the proposed algorithm is effective in tackling environmental challenges and can be employed in similar environments.

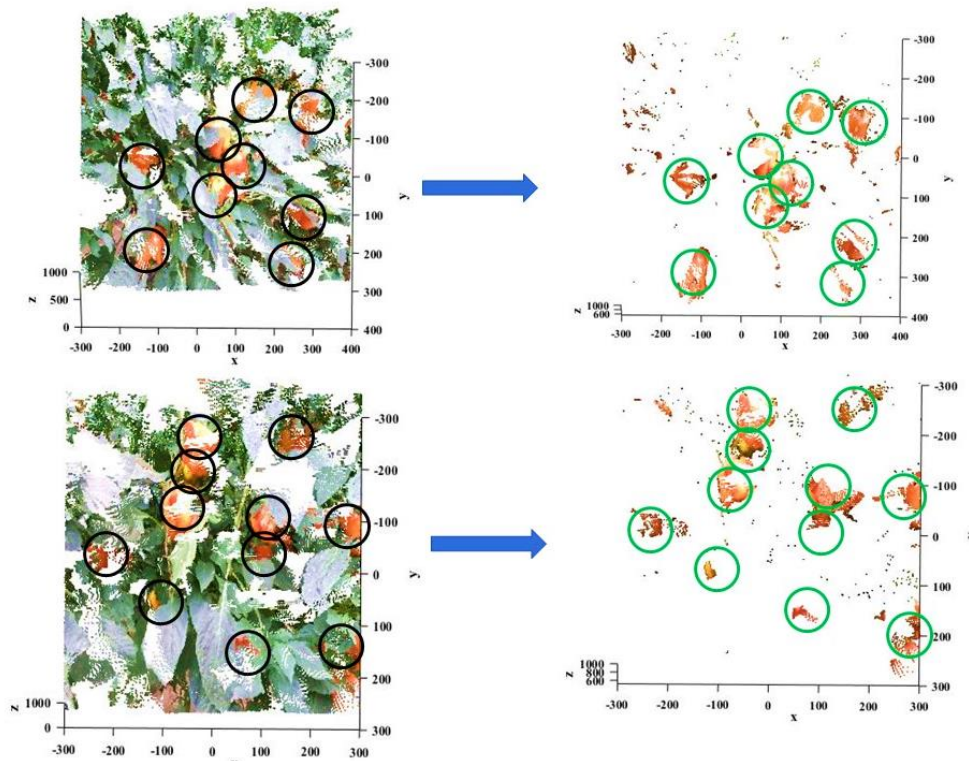


Fig.10. The initial 3-D model (left), and identified orange peppers (right) used for human monitoring

To detect sweet peppers in complex environments, various techniques were employed, including noise removal, data balancing via color filtering, 3D feature extraction, PCA, and unsupervised *K*-means clustering. To assess the performance of the *k*-means algorithm, the silhouette coefficient was utilized (as shown in Table 3). The maximum value of the silhouette coefficient was 0.59, while the minimum value was 0.35. Out of the 99 peppers present in the point cloud models, 93 peppers were detected. Finally, the average

accuracy of human-supervised detection was found to be 95.10% for the sweet pepper dataset. The average execution time of the algorithm was determined to be 33.32 seconds. Comparable levels of accuracy to the results for sweet pepper detection have been reported in similar studies. For example, (Zhao *et al.*, 2020) used R-G filter and the Euclidean distance to detect color sweet pepper, and the accuracy of their human monitoring was 90.69%. In a study, (Sa *et al.*, 2017) used PFH features to detect sweet peppers with 0.71%

accuracy, considered acceptable because they used only 3D features, without incorporating any color features. Ning *et al.* (2022) achieved a precision of 91.84% in detecting sweet peppers using the YOLO-V4 model. Nan *et al.* (2023) used the YOLOv5l model for detecting green sweet peppers and achieved a detection

accuracy of 81%. Compared to supervised detection of colored sweet peppers (Doosti *et al.*, 2023), the detection accuracy in this study increased from 83.07% to 95.1%, mainly due to appropriate preprocessing applied to the point cloud, including reducing the sample size.

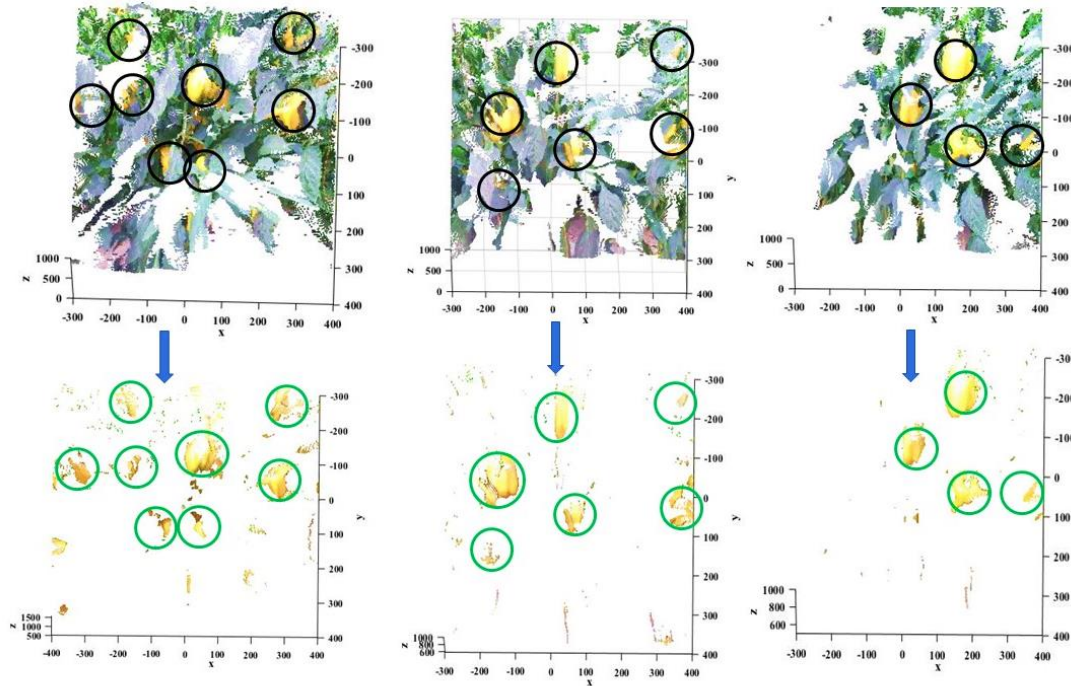


Fig.11. Detection of yellow pepper for human monitoring: initial 3D model (top), and detected peppers (bottom)

Table 3- The results of the human monitoring

3-D model number	Silhouette coefficient	Total number of sweet peppers	Number of identified sweet peppers	Detection accuracy (%)	Time (s)	Cumulative variance
1	0.59	9	9	100	37.36	50.76
2	0.41	11	11	100	31.29	51.55
3	0.44	4	4	100	34.85	54.92
4	0.40	6	6	100	29.45	54.14
5	0.40	8	8	100	30.67	54.83
6	0.51	7	6	85.71	33.11	63.36
7	0.46	8	7	87.5	32.62	66.16
8	0.52	3	3	100	13.45	64.28
9	0.44	5	5	100	32.12	65.97
10	0.58	2	2	100	16.84	65.74
11	0.44	10	9	90	46.64	60.2
12	0.38	6	6	100	47.54	60
13	0.48	4	4	100	20.12	60.78
14	0.39	6	5	83.33	54.20	59.8
15	0.35	10	8	80	50.66	60.30
Average	0.45	99	93	95.10	33.32	51.59

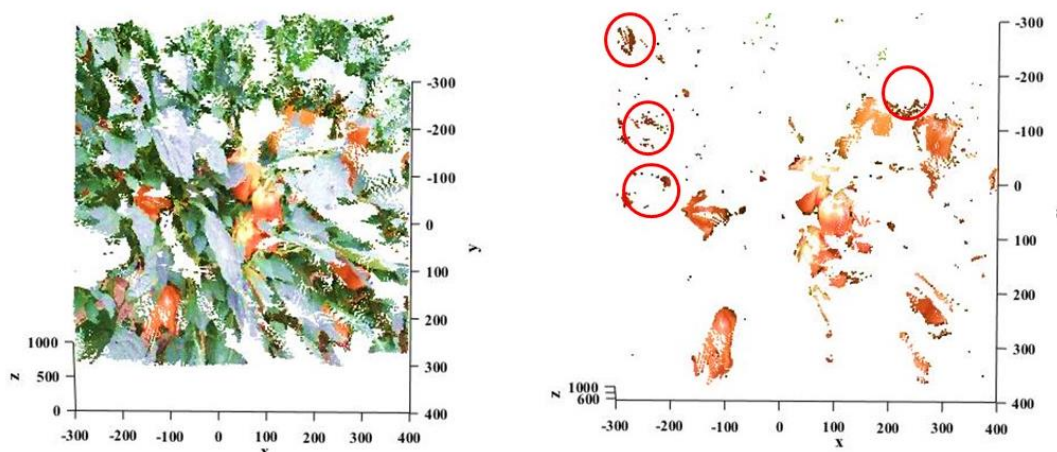


Fig.12. Recognition of the points of the branches and leaves as sweet pepper

The use of unsupervised learning algorithms in detecting agricultural products is prioritized over supervised learning algorithms. This is because, in supervised learning, changes in environmental conditions can affect the accuracy of detection and reduce it. In the detection of agricultural products, color supervision is mostly used, which can be influenced by changes in environmental lighting. Therefore, the superiority of unsupervised algorithms is due to the lack of need for supervision. Thus, these algorithms are more resistant to environmental changes, and their accuracy will be less affected.

Conclusion

Automatic detection and harvesting of sweet pepper is a challenging task in greenhouse cultivation due to the high density of branches and leaves, variations in lighting and environmental conditions, the presence of pests and diseases, and differing levels of maturity. In this study, a combination of geometric and color features using 3D models was used as a highly accurate method for detecting sweet peppers. The results showed that the algorithm used was capable of detecting sweet peppers with an accuracy of 95.1%. The algorithms used in this study can be used to detect other products with similar challenges such as green sweet peppers, cucumbers, green apples, and generally any product that is geometrically different from

other plant components. The authors aimed to investigate whether geometric features are capable of distinguishing fruit from other plant components in order to provide a solution for detecting green sweet peppers in future studies. As detecting green sweet peppers is also a challenging task, it is suggested that the detection of green sweet peppers based on local 3D descriptors be investigated in a separate study.

Acknowledgment

We would also like to thank Dr. Amirhossein Nayebi Astaneh for his guidance and the following organizations for their financial support of this research.

- 1- Ferdowsi University of Mashhad
- 2- Khorasan Science and Technology Park

Conflict of Interest: The authors declare no competing interests.

Authors Contribution

O. Doosti Irani: Study design, 3D imaging, Data preprocessing, Programming, Data analysis and modeling, Writing - original draft.

M. H. Aghkhani: Supervision, Scientific oversight, Review & editing.

M. R. Golzarian: Supervision, Scientific oversight, Technical consultation, Review & editing.

All authors reviewed and approved the final manuscript.

References

1. Adams, T., Nolen, S., Sweezy, J., Zukaitis, A., Campbell, J., Goorley, T., ... & Aulwes, R. (2015). Monte Carlo application toolkit (MCATK). *Annals of Nuclear Energy*, 82, 41-47. <https://doi.org/10.1016/j.anucene.2014.08.047>
2. Ball, D., Upcroft, B., Wyeth, G., Corke, P., English, A., Ross, P., ... & Bate, A. (2016). Vision-based obstacle detection and navigation for an agricultural robot. *Journal of Field Robotics*, 33(8), 1107-1130. <https://doi.org/10.1002/rob.21644>
3. Behley, J., Steinhage, V., & Cremers, A. B. (2012, May). Performance of histogram descriptors for the classification of 3D laser range data in urban environments. In *2012 IEEE international conference on robotics and automation* (pp. 4391-4398). IEEE. <https://doi.org/10.1109/icra.2012.6225003>
4. Chidambaranathan, C. M., Handa, S. S., Ramanamurthy, M. V., & Ramanamurthy, M. V. (2018). Development of smart farming-a detailed study. *International Journal of Engineering & Technology*, 7(2.4), 56. <https://doi.org/10.14419/ijet.v7i2.4.10042>
5. Dharmaraj, V., & Vijayanand, C. (2018). Artificial intelligence (AI) in agriculture. *International Journal of Current Microbiology and Applied Sciences*, 7(12), 2122-2128. <https://doi.org/10.20546/ijcmas.2018.712.241>
6. Doosti-Irani, O., Golzarian, M. R., & Aghkhani, M. H. (2023). Automatic recognition of sweet peppers based on the fast point features histogram (FPFH) 3-D descriptor and machine learning. *Journal of Researches in Mechanics of Agricultural Machinery*, 12(1), 27-40. <https://doi.org/10.22034/jrmam.2023.13863.587>
7. Fu, L., Majeed, Y., Zhang, X., Karkee, M., & Zhang, Q. (2020). Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosystems Engineering*, 197, 245-256. <https://doi.org/10.1016/j.biosystemseng.2020.07.007>
8. Gan, H., Lee, W. S., Alchanatis, V., Ehsani, R., & Schueller, J. K. (2018). Immature green citrus fruit detection using color and thermal images. *Computers and Electronics in Agriculture*, 152, 117-125. <https://doi.org/10.1016/j.compag.2018.07.011>
9. Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, 116, 8-19. <https://doi.org/10.1016/j.compag.2015.05.021>
10. Gongal, A., Silwal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2016). Apple crop-load estimation with over-the-row machine vision system. *Computers and Electronics in Agriculture*, 120, 26-35. <https://doi.org/10.1016/j.compag.2015.10.022>
11. Han, X. F., Sun, S. J., Song, X. Y., & Xiao, G. Q. (2018). 3D point cloud descriptors in hand-crafted and deep learning age: State-of-the-art. *arXiv preprint arXiv:1802.02297*. <https://doi.org/10.48550/arXiv.1802.02297>
12. Hemming, J., Bac, C. W., & Van Tuijl, B. A. J. (2011). CROPS project deliverable 5.1: Report with design objectives and requirements for sweet-pepper harvesting. Wageningen, The Netherlands: Wageningen UR Greenhouse Horticulture.
13. Javidan, S. M., Banakar, A., Vakilian, K. A., & Ampatzidis, Y. (2023). Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning. *Smart Agricultural Technology*, 3, 100081. <https://doi.org/10.1016/j.atech.2022.100081>
14. Kurtulmus, F., Lee, W. S., & Vardar, A. (2014). Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network. *Precision Agriculture*, 15, 57-79. <https://doi.org/10.1007/s11119-013-9323-8>
15. Lachat, E., Macher, H., Mittet, M. A., Landes, T., & Grussenmeyer, P. (2015). First experiences with kinect v2 sensor for close range 3d modelling. *International Archives of the*

- Photogrammetry, Remote Sensing & Spatial Information Sciences.*
<https://doi.org/10.5194/isprsarchives-XL-5-W4-93-2015>
16. Moghimi, A., Aghkhani, M. H., & Golzarian, M. R. (2015). Designing of Computer Vision Algorithm to Detect Sweet Pepper for Robotic Harvesting Under Natural Light. *Journal of Agricultural Machinery*, 5(1), 82-91. <https://doi.org/10.22067/jam.v5i1.23528>
 17. Mohamad zamani, D., Javidan, S. M., Zand, M., & Rasouli, M. (2023). Detection of Cucumber Fruit on Plant Image Using Artificial Neural Network. *Journal of Agricultural Machinery*, 13(1), 27-39. <https://doi.org/10.22067/jam.2022.73827.1077>
 18. Mohammadi, P., Massah, J., & Asefpour Vakilian, K. (2023). Robotic date fruit harvesting using machine vision and a 5-DOF manipulator. *Journal of Field Robotics*. <https://doi.org/10.1002/rob.22184>
 19. Muja, M., Rusu, R. B., Bradski, G., & Lowe, D. G. (2011, May). Rein-a fast, robust, scalable recognition infrastructure. In *2011 IEEE international conference on robotics and automation* (pp. 2939-2946). IEEE. <https://doi.org/10.1109/icra.2011.5980153>
 20. Nan, Y., Zhang, H., Zeng, Y., Zheng, J., & Ge, Y. (2023). Faster and accurate green pepper detection using NSGA-II-based pruned YOLOv5l in the field environment. *Computers and Electronics in Agriculture*, 205, 107563. <https://doi.org/10.1016/j.compag.2022.107563>
 21. Nguyen, T. T., Vandevoorde, K., Kayacan, E., De Baerdemaeker, J., & Saeys, W. (2014, July). Apple detection algorithm for robotic harvesting using a RGB-D camera. In *International Conference of Agricultural Engineering, Zurich, Switzerland*.
 22. Ning, Z., Luo, L., Ding, X., Dong, Z., Yang, B., Cai, J., ... & Lu, Q. (2022). Recognition of sweet peppers and planning the robotic picking sequence in high-density orchards. *Computers and Electronics in Agriculture*, 196, 106878. <https://doi.org/10.1016/j.compag.2022.106878>
 23. Ringdahl, O., Kurtser, P., & Edan, Y. (2019). Evaluation of approach strategies for harvesting robots: Case study of sweet pepper harvesting: Category:(5). *Journal of Intelligent & Robotic Systems*, 95(1), 149-164. <https://doi.org/10.1007/s10846-018-0892-7>
 24. Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)
 25. Rusu, R. B., & Cousins, S. (2011, May). 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation* (pp. 1-4). IEEE. <https://doi.org/10.1109/icra.2011.5980567>
 26. Rusu, R. B., Blodow, N., & Beetz, M. (2009, May). Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation* (pp. 3212-3217). IEEE. <https://doi.org/10.1109/robot.2009.5152473>
 27. Rusu, R. B., Marton, Z. C., Blodow, N., & Beetz, M. (2008, July). Persistent point feature histograms for 3D point clouds. In *Proc 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany* (pp. 119-128). <https://doi.org/10.3233/978-1-58603-887-8-119>
 28. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors*, 16(8), 1222. <https://doi.org/10.3390/s16081222>
 29. Sa, I., Lehnert, C., English, A., McCool, C., Dayoub, F., Upcroft, B., & Perez, T. (2017). Peduncle detection of sweet pepper for autonomous crop harvesting—combined color and 3-D information. *IEEE Robotics and Automation Letters*, 2(2), 765-772. <https://doi.org/10.1109/LRA.2017.2651952>
 30. Shamshiri, R. R., Hameed, I. A., Karkee, M., & Weltzien, C. (2018). Robotic harvesting of fruiting vegetables: A simulation approach in V-REP, ROS and MATLAB. *Proceedings in Automation in Agriculture-Securing Food Supplies for Future Generations*, 126, 81-105. <https://doi.org/10.5772/intechopen.73861>

31. Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19, 221-248. <https://doi.org/10.1146/annurev-bioeng-071516-044442>
32. Stein, M., Bargoti, S., & Underwood, J. (2016). Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors*, 16(11), 1915. <https://doi.org/10.3390/s16111915>
33. Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., & Zou, X. (2020). Recognition and localization methods for vision-based fruit picking robots: A review. *Frontiers in Plant Science*, 11, 510. <https://doi.org/10.3389/fpls.2020.00510>
34. Tao, Y., & Zhou, J. (2017). Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Computers and Electronics in Agriculture*, 142, 388-396. <https://doi.org/10.1016/j.compag.2017.09.019>
35. Wan, Y., Li, Y., Jiang, J., & Xu, B. (2020, March). Edge Voxel Erosion for Noise Removal in 3D Point Clouds Collected by Kinect. In Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing (pp. 59-63). <https://doi.org/10.1145/3388818.3388821>
36. Zhao, X., Li, H., Zhu, Q., Huang, M., Guo, Y., & Qin, J. (2020). Automatic sweet pepper detection based on point cloud images using subtractive clustering. *International Journal of Agricultural and Biological Engineering*, 13(3), 154-160. <https://doi.org/10.25165/j.ijabe.20201303.5460>

تشخیص فلفل دلمه‌ای با استفاده از هیستوگرام ویژگی سریع نقطه و یادگیری بدون نظارت

امید دوستی ایرانی^{۱*}، محمدحسین آق خانی^۱، محمودرضا گلزاریان^۱

تاریخ دریافت: ۱۴۰۲/۰۳/۳۱

تاریخ پذیرش: ۱۴۰۲/۰۵/۲۵

چکیده

برداشت رباتیک محصولات کشاورزی فرآیندی مهم و موثر برای تولید میوه سالم، کاهش هزینه‌های برداشت و افزایش بهره‌وری است. با پیشرفت بینایی ماشین، استفاده از اطلاعات سه‌بعدی به جای اطلاعات دوبعدی در حال گسترش است. با این حال، برداشت فلفل دلمه‌ای به عنوان یکی از محصولات گلخانه‌ای، به دلیل دقت پایین سنسورهای دوبعدی با چالش‌هایی مواجه است. هدف این مطالعه توسعه یک الگوریتم بینایی ماشین بدون نظارت برای تشخیص فلفل دلمه رنگی با استفاده از ترکیبی از ویژگی‌های هندسی (هیستوگرام ویژگی نقطه سریع - FPFH) و ویژگی‌های رنگی (HSV) است. تصاویر عمق با استفاده از حسگر Kinect-v2 دریافت و مدل سه‌بعدی بازسازی شده است. پس از استخراج ویژگی‌های هندسی و رنگ، داده‌ها با استفاده از روش زیر نمونه‌گیری و با اعمال معیار Z-score برای فیلتر کردن نویزها، پیش‌پردازش شدند. تحلیل مؤلفه اصلی (PCA) برای کاهش ابعاد ویژگی‌ها استفاده شد و مدل خوشه‌بندی k-means با استفاده از شش ویژگی هندسی و سه ویژگی رنگ، به داده‌ها اعمال شد. ضریب سیلوئت برای ارزیابی کیفیت خوشه‌بندی استفاده شد و ارزیابی انسانی نشان داد که الگوریتم با دقت ۹۵/۱۰ درصد قادر به تشخیص فلفل دلمه‌ای است.

واژه‌های کلیدی: برداشت رباتیک، بینایی ماشین، فلفل دلمه‌ای، هیستوگرام ویژگی سریع نقطه، K-means

۱- گروه مهندسی بیوسیستم، دانشکده کشاورزی، دانشگاه فردوسی مشهد، مشهد، ایران

(*)- نویسنده مسئول: (Email: omiddoostiirani@gmail.com)