U.P.B. Sci. Bull., Series C, Vol. 87, Iss. 3, 2025 ISSN 2286 – 3540

KEY TECHNOLOGIES FOR DENSE MUSHROOM GROUP PICKING BASED ON IMPROVED DBSCAN CLUSTERING ALGORITHM

Yafei LI[1*], Xuanzhang ZHU[2]

Traditional mushroom harvesting techniques are inadequate to meet the increasing demands of modern agriculture. This study proposes a dense mushroom cluster harvesting planning technique based on an improved Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The proposed method combines clustering and harvesting planning by optimizing the DBSCAN algorithm. Key metrics such as clustering accuracy, intra-cluster point omission probability, and running time were analyzed and compared to existing algorithms. The optimized DBSCAN showed superior performance with clustering accuracy of 94.6%, intra-cluster omission probability of 2.5%, and a reduced running time of 0.25s. The system achieved a recognition accuracy of 93.8% and 95.8% for mushroom clusters, with picking success rates of 93.2% and 94.7%, respectively. This study introduces a novel harvesting planning approach that improves the efficiency and success rate of mushroom harvesting, reducing damage and meeting the required harvesting efficiency.

Keywords: DBSCAN clustering algorithm; clusters of mushrooms; classification; concentrated; picking planning

1. Introduction

Mushroom plants contain rich nutritional and medicinal values, which can be processed and transformed into diverse foods, health products, and medicines, demonstrating extremely high economic value. However, the growth of mushroom colonies exhibits significant randomness, leading to diversity in size, shape, and density, and often overlapping between adjacent mushrooms. This undoubtedly increases the difficulty of harvesting and reduces harvesting efficiency. With the continuous growth of mushroom demand and production, the global mushroom industry is facing challenges such as harvesting costs difficultly to be controlled and urgent need to improve production efficiency [1-2]. In this context, mechanical harvesters are gradually being applied in the mushroom industry and have demonstrated the advantages of efficient harvesting. However, in high-value applications, there are extremely strict requirements for the size, quality, and integrity of mushrooms. Whether it is manual picking or one-time mechanical harvesting, it is difficult to find a perfect balance between quality, efficiency, and

[1] School of Information Engineering, Hunan University of Science and Engineering, Yongzhou, China, *Corresponding author, e-mail lyf07200601@126.com
[2] Information and Network Center, Hunan University of Science and Engineering, Yongzhou, China

cost. In response to the current situation, there is an urgent need to develop a Mushroom Harvesting Robot (MHR) with high-precision intelligent harvesting capabilities to meet production needs [3-4].

At present, the continuous improvement of artificial intelligence algorithms has further developed mushroom harvesting technology. To ensure timely harvesting of mushrooms and improve harvesting efficiency, Ji et al. proposed an online mushroom size detection algorithm built on deep image processing. In the process of machine vision recognition, the recognition accuracy of this algorithm was higher than 92.50%, the missed detection rate was lower than 4.95%, the false detection rate was lower than 2.15%, and the Diameter Measurement Error (DME) was less than 4.50%. This image processing algorithm had a high recognition rate and small DME, which could satisfy the demands of picking operations [5]. To achieve fully autonomous harvesting of shiitake mushrooms in greenhouses, Rong et al. proposed a shiitake mushroom harvesting robot based on deep model detection algorithm. Field experiments have verified the feasibility of the robot system, with a mushroom recognition Success Rate (SR) of 95% and a harvesting SR of 86.8%. The harvesting time for a single mushroom was 8.85 seconds [6]. To promote the development of the mushroom industry, Sujatanagarjuna et al. constructed an intelligent, automated, and scalable indoor mushroom harvesting system based on an improved convolutional neural network. It was found that the system had a training accuracy of 91.7% and a semi-automatic harvesting system, and its modularity and scalability allow for industrial grade use, which could be expanded according to the mushroom planting system required in the facility [7]. To promote harvesting robots to overcome complex growth environments, diverse shapes, dense shadows, and variable fields of view, Cong et al. proposed a lightweight mushroom detection model built on YOLO v3. The model achieved an average accuracy of 97.03%, with parameters of 29.8M and a detection speed of 19.78ms, demonstrating good real-time and detectability, and having 2.08 times fewer parameters than the original model. This study provided a vital theoretical basis for the automatic harvesting of fresh shiitake mushrooms [8].

Huang et al. designed a robotic mushroom harvesting machine to address the time-consuming and labor-intensive harvesting of mushrooms. This design includes a harvesting end effector based on bending motion, a mushroom stem trimming end effector, and an electric pneumatic control system. The machine ultimately achieved a SR of 97% [9]. To achieve intelligent and automated harvesting of white mushrooms, Recchia et al. developed a collaborative robot with dedicated arm end tools for harvesting mature mushrooms from selected areas. This method improved the working environment for workers and reduced work-related musculoskeletal disorders [10]. The labor cost of mushroom harvesting accounted for 50-80% of the total labor cost, and the high humidity and low temperature factory environment posed a risk of rheumatism for workers. Given this, Shi et al.

proposed a new underactuated gripper based on screw and linear bearings, which can perform flexible force control operations while measuring mushroom diameter. The static grasping force error of the gripper during the entire grasping process was 0.195N, and the mean separation force overshoot was 1.31N. The in-situ measurement of mushroom diameter achieved an accuracy of 97.3% and a SR of 98.3% [11]. To achieve the goal of automated mushroom harvesting, Hubay et al. proposed an image processing-based cultivation mushroom automatic harvesting technology. This technology processed images at a mean speed of 0.78s and generated coordinates with a SR of 92% [12].

In summary, current research on mushroom harvesting technology has achieved certain results in harvesting area segmentation, harvesting sequence planning, and recognition detection, which can realize automatic mushroom harvesting and grading. However, when mushrooms grow in clusters, some of them are obscured by other mushrooms, resulting in incomplete contour information and a large number of mushrooms growing in clusters. In this case, the accuracy of the detection algorithm will significantly decrease. Therefore, this study innovatively combines the ideas of clustering and harvesting planning and proposes a Dense Mushroom Cluster Harvesting (DMCH) planning technique based on an improved Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The purpose is to plan a suitable harvesting sequence based on the different growth characteristics of the mushroom population, ultimately achieving efficient and low loss harvesting of the mushroom population.

2. Methods and materials

2.1 Design of Dense Mushroom Group Classification Algorithm Based on Improved DBSCAN Algorithm

In practical application scenarios, traditional mushroom harvesting algorithms face two major challenges: low harvesting efficiency and high damage rate. Especially in the two key mushroom clusters where overlapping phenomena occur frequently and are densely packed, the problem is particularly prominent. To address these issues, this study conducts a detailed classification of mushroom populations based on their different growth characteristics. Specifically, this study categorizes mushroom populations with severe overlapping phenomena as the first type of region. The second type of area includes closely arranged mushroom clusters with small differences in height, characterized by high density and low overlap, making them relatively easy to pick [13-14]. The last category is easy to pick areas, where dispersed, low-density, and low overlapping mushroom clusters are the main ones. Based on the above observations, this study clearly divides mushroom populations into three categories: Overlapping Mushroom Populations (OverMP), Dense Mushroom Populations (DenseMP), and Discrete Mushroom Populations (DisMP) according to the two major indicators of overlap rate and density [15]. After establishing the classification criteria for mushroom clusters, this study designed a mushroom cluster classification algorithm based on density and overlap rate parameters. The overall structure is displayed in Fig.1.

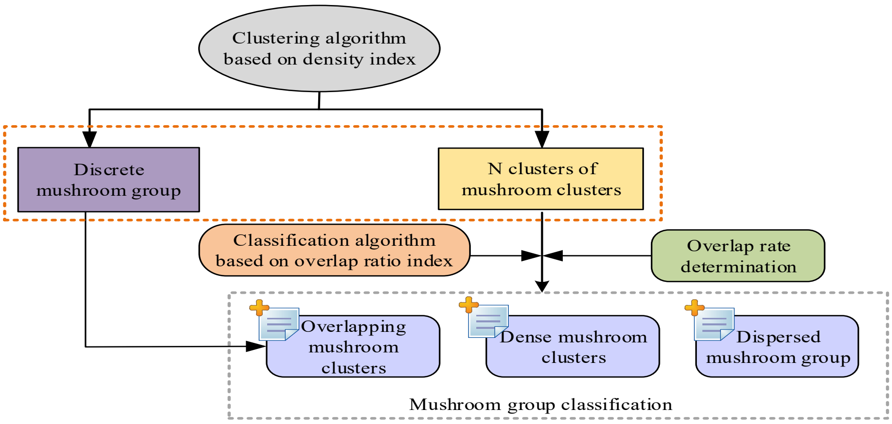


Fig. 1. Mushroom Group Classification Algorithm

In this algorithm, this study first divides the entire mushroom population within the view into clusters using a clustering algorithm constructed grounded on density indicators and filters out DisMP among them. Then, using a classification algorithm grounded on the Overlap Rate Index (ORI), the clustered mushroom clusters were classified into three categories: OverMP, DenseMP, and DisMP [16]. This study summarizes the clustering problem based on density indicators as a clustering problem and selects DBSCAN to achieve mushroom clustering based on specific information of the mushroom population. The core idea of the DBSCAN is to use the density of sample points to construct cluster classes and handle noise in the dataset. It does not require a preset number of clusters, can handle clusters of any shape, and is insensitive to outliers, as shown in Fig.2.

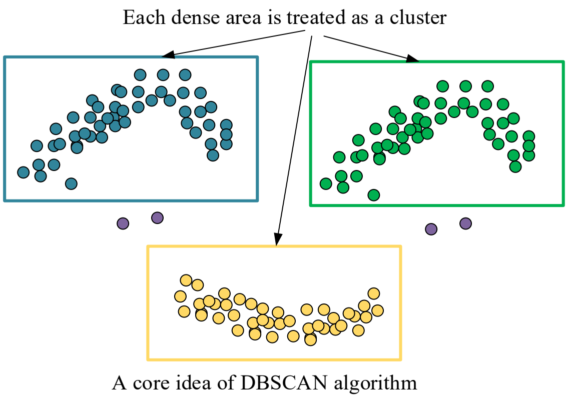


Fig. 2. DBSCAN Algorithm Structure

DBSCAN has a wide range of application scenarios, and the radius r and minimum sample size N of the algorithm will have a significant effect on the clustering performance. When dealing with the complex task of clustering mushroom clusters, the disorderly growth of mushroom clusters leads to significant density differences between them. The non-uniformity of this density poses a challenge to the algorithm, especially when selecting neighborhood radii [17]. If the radius value is not appropriate, it may cause a decrease in the SR of clustering. In other words, for datasets with uneven distribution or significant density changes, traditional clustering algorithms may struggle to achieve ideal clustering results. To solve this problem and improve the clustering effect, the DBSCAN clustering algorithm is improved. DBSCAN The improvement of the DBSCAN algorithm is to introduce adaptive local radius α, which dynamically adjusts the neighborhood range of the core point according to the distance of the furthest core point in the unclassified point, so as to filter instead of the preset parameter values. The improved algorithm flow is shown in Fig. 3.

The core of improving the algorithm is to change the neighborhood range of the core point but still pre-set the value of r. After randomly selecting the unclassified point O in the dataset, if O is determined as the core point, that is, there are at least N core points within a circle with O as the center and r as the radius. Afterwards, only N core points are retained in the O neighborhood, and the value of α is taken as the distance from O to the farthest core point in the neighborhood. Then, α is used to replace r for filtering at point O. In the modified DBSCAN algorithm, the algorithm first computes the distance from the unclassified point to the furthest core point within its neighborhood and dynamically determines the neighborhood radius of each core point based on that distance.

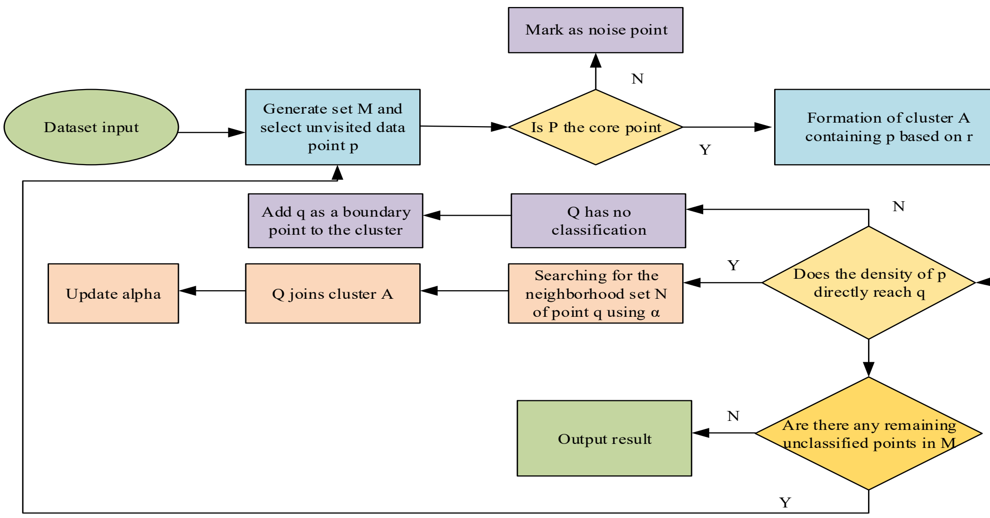


Fig. 3. Improved DBSCAN Clustering Algorithm Flowchart

For range image data, this improved method enables better adaptation to the density changes in different regions in the image. For example, in an image, the high density area (such as the target object) will have a smaller neighborhood radius, while the low density area (such as background or noise) will have a larger neighborhood radius. This dynamic adjustment allows the algorithm to more accurately identify the target area in the image while effectively filtering the noise. In the point cluster class, the adaptive radius mechanism optimizes the clustering process through local density information, which makes the algorithm perform better in processing uneven distributed data, and can more accurately divide cluster boundaries and reduce misclassification.

2.2 Design of Mushroom Group Classification Method Based on ORI

After using the density-based clustering algorithm to cluster the mushroom population, it will be subdivided into several mushroom population sets and several discrete mushroom population individuals. Next, this study needs to further classify these clustered mushroom populations into OverMP, DenseMP, or DisMP based on the ORI. Firstly, this study will examine whether there is overlap within each mushroom group set. If there is no overlap, then the mushroom group set is classified as DisMP. If there is an overlap, this study will further calculate the overlap rate of individual mushroom groups and calculate the average overlap rate of the mushroom group. If the average overlap rate is greater than or equal to the preset threshold, then the mushroom group is classified as OverMP. If it is less than the preset threshold, it is classified as DenseMP. To accurately calculate the overlapping area of the mushroom group, this study first uses the cosine theorem to calculate the angle between the centers of two circles, as well as their corresponding fan-shaped areas. Then, through mathematical operations, the overlapping area S between the two circles is calculated, as shown in equation (1) [18].

$$S = sqrt\left[s(s-a)(s-b)(s-c)\right] \tag{1}$$

In equation (1), a, b, and c represent the lengths of the three sides of a triangle. s represents half of the circumference of the triangle. If e and f are two mushroom groups with overlapping relationships. If f is obstructed by e, the overlap rate ω_e of individuals in mushroom group e can be calculated using equation (2).

$$\omega_e = \frac{S_{ef}}{S_e} \tag{2}$$

In equation (2), S_{ef} is the overlapping area between mushroom group e and f. S_e represents the area of mushroom group e. If a single mushroom group overlaps with several mushroom groups, the ω_e value is calculated twice. The formula for calculating the mean ORI of mushroom clusters is given by equation (3).

$$\omega = \frac{\sum_{i=1}^{e} \omega_i}{e} \tag{3}$$

In equation (3), ω is the overlap rate of the mushroom group. ω_i is the overlap rate of individuals in the mushroom group that have overlapping relationships. e is the number of overlapping individual mushrooms in the mushroom group. The mushroom community is classified and defined based on the actual environmental conditions. Specifically, this study defines mushroom populations with minimal overlap between individuals and an overlap rate of almost 0 as DenseMP. On the contrary, if there are a large number of overlapping individuals within the mushroom group, and the overlap rate is high, it is defined as OverMP. To clearly distinguish between these two mushroom groups, this study sets an overlap rate threshold of 4.5%. When the overlap rate of the mushroom group is below this threshold, it is considered DenseMP. When the overlap rate is higher than or equal to this threshold, it is considered OverMP.

2.3 Design of DMCH Planning Method

To output the sequence of harvesting all mushroom clusters while ensuring high efficiency and low damage rate, this study proposes a DMCH planning method that combines global planning and local planning. The overall process of this picking method is shown in Fig.4.

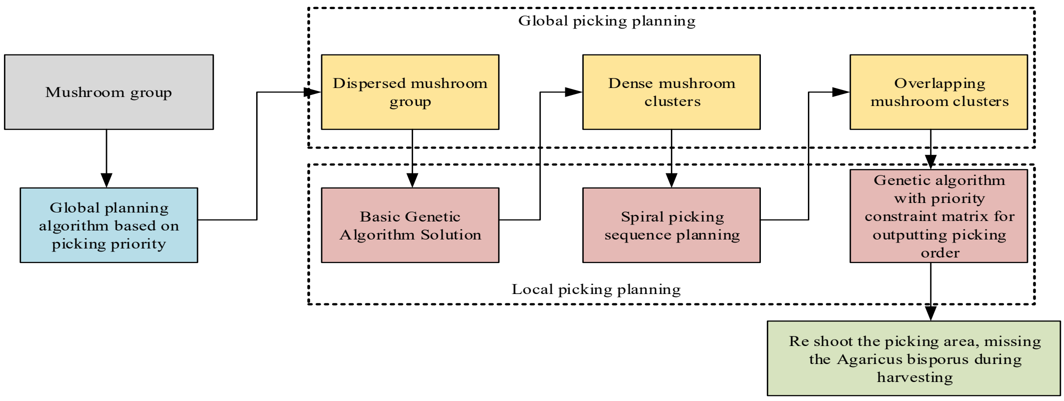


Fig. 4. Process of DMCH Planning Method

This method combines global planning with local harvesting strategies. At the global planning level, the main goal is to improve harvesting efficiency, and to plan the harvesting sequence for each mushroom group, treating the mushroom group as the smallest planning unit. The planning process comprehensively considers factors such as the difficulty of harvesting three types of mushroom clusters, the calculation time required by the algorithm, and the actual harvesting time, and finally outputs a reasonable mushroom cluster harvesting sequence. In terms of local harvesting strategies, this study focuses more on reducing the damage rate during the harvesting process, while also taking into account efficiency indicators. This study specifically designs a strategy to address the issue of high harvesting damage caused by overlapping mushroom populations. For densely grown mushroom clusters, this study aims to address high picking resistance caused by the close arrangement of mushroom clusters. For dispersed mushroom populations, this study aims to improve harvesting efficiency and conducts corresponding harvesting planning. The schematic diagram of the picking planning strategy that combines global and local aspects is shown in Fig. 5.

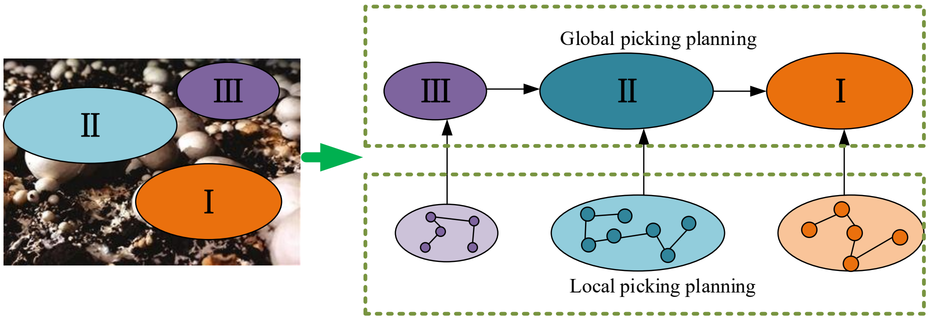


Fig. 5. DMCH Planning Method

In Fig.5, the proposed classification algorithm divides the mushroom population into three categories: I, II, and III. I is OverMP, II is DenseMP, and III is DisMP. The global harvesting plan plans the harvesting sequence of three mushroom groups based on corresponding indicators. The picking sequence is to pick DisMP first, then DenseMP, and finally OverMP. Local harvesting planning refers to the planning of harvesting sequences within three types of mushroom clusters: I, II, and III.

The path planning for mushroom harvesting this time uses Genetic Algorithm (GA). This algorithm simulates phenomena such as genetics, mutation, and natural selection in the process of biological evolution to solve complex optimization problems. It has strong global search capability and wide applicability and has been widely utilized in path planning issues in various fields [19-21]. GA has unique advantages in solving complex optimization problems, but it also has problems such as slow convergence speed and susceptibility to getting stuck in local optima. In practical applications, algorithms need to be improved based on the specific characteristics and requirements of the problem [22-23]. Therefore, this study focuses on OverMP and proposes an improved GA based on introducing a priority constraint matrix, considering its overlapping occlusion characteristics. The improved GA operation process is shown in Fig.6.

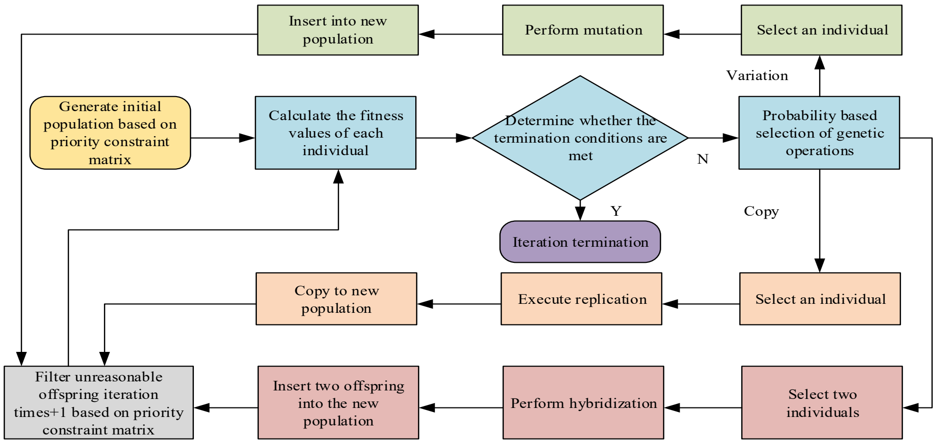


Fig. 6. Improved GA Process Diagram

In improving GA, the first step is to establish a priority constraint matrix M_p built on the occlusion between mushroom clusters. M_p is an $n \times n$-dimensional matrix, where n represents the quantity of harvestable mushroom populations and $i, j = 1, 2, ..., n$. The formula for generating the initial population based on M_p is shown in equation (4).

$$M_p = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1j} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2j} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots & p_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nj} & \cdots & p_{nn} \end{bmatrix} \tag{4}$$

Next, the fitness function f defined by calculating the actual path of mushroom harvesting is shown in equation (5).

$$f = \frac{1}{\sum_{i=1}^{n-1} \sqrt{|y_{i+1} - y_i|^2 + x_{i+1}^2}} \tag{5}$$

The optimization process first applies the roulette wheel selection operator and combines it with the elite retention strategy to ensure that the top n individuals with the best performance in the population can be retained. Subsequently, the roulette wheel selection method is used again to select individuals and involve them in the process of crossover and mutation. During the crossover process, this study employs the single point mapping crossover method. This method effectively avoids the occurrence of duplicated gene fragments by establishing a mapping table between two parental chromosomes. The mutation process uses the exchange mutation operator, which randomly selects two gene fragments and exchanges their positions. When the fitness value of the population gradually stabilizes or reaches the maximum iterations set by the algorithm, the optimization process ends. For densely growing mushroom clusters, due to their tight arrangement and high friction during harvesting, this study specially designs a spiral harvesting sequence that gradually advances from the periphery to the center. The dispersed mushroom clusters are optimized using basic GA to minimize the total harvesting path and output the optimal harvesting sequence.By the above methods, the overall scheme of dense mushroom group picking was obtained, as shown in Fig. 7.

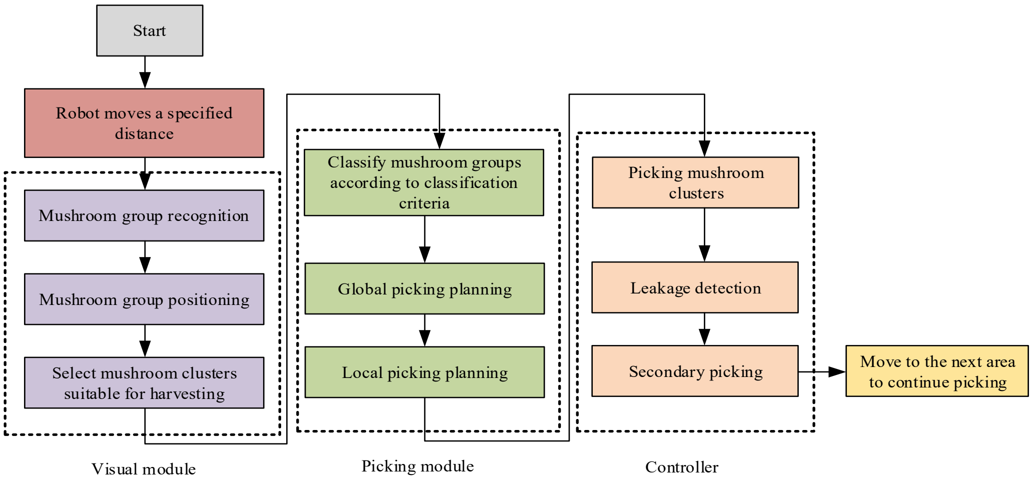


Fig.7 Design and architecture diagram of the overall picking scheme of dense mushrooms

In Fig. 7, the visual module is responsible for identifying the mushroom group in the field of vision, identifying and filtering the depth information through the algorithm, obtaining the actual position and size of the mushroom group after converting the coordinate system, and then transmitting it to the picking planning module. The picking planning module includes mushroom group classification and picking sequence planning. The former divides mushroom groups into three categories: overlapping, dense and dispersed, while the latter plans the picking order between and within clusters. The control module carries out the picking task according to the plan, and the area is re-detected to pick the leaky mushroom group. The whole process cycles until the current area is picked and then transferred to the next area. The system realizes the accurate identification, classification, planning of the picking sequence and the execution of the picking task, and improves the picking efficiency.

3. Results

3.1 Algorithm Performance Testing

To validate the DMCH method, this study captured 4,870 images (1280×720 pixels) of mushroom clusters across four growth stages in an industrial park using a RealsenseD435i depth camera mounted 25 – 30 cm above the mushrooms on an MHR. The dataset included 1,640 photos for the densest group, 1,450 for the second stage, 930 for the third, and 850 for the fourth. Using OpenCV, images were enhanced via flipping, cropping, rotation, color adjustment, noise addition, and scaling, expanding the dataset to 7,300 images for improved recognition accuracy. Table 1 shows the configuration parameters for this experiment.

Table 1

Experimental environment

parameter	Experimental environment
Tool	Intel Core i5-4200 CPU
Processor	11th GenIntel(R)Core(TM)i5-1135G7@2.40GHz-2.42GHz
Memory capacity	4GB RAM
Operating system	Windows7
Data mining software	SPSS Modeler18.0
Programming environment	Python3.8.3
Programming IDE	Anaconda3
model building	Python3.8.3

This study used Intel Core i5-4200 CPU as the platform and divided the corresponding training and validation sets in a 4:1 ratio, and trained the research model on them, as shown in Fig.8.

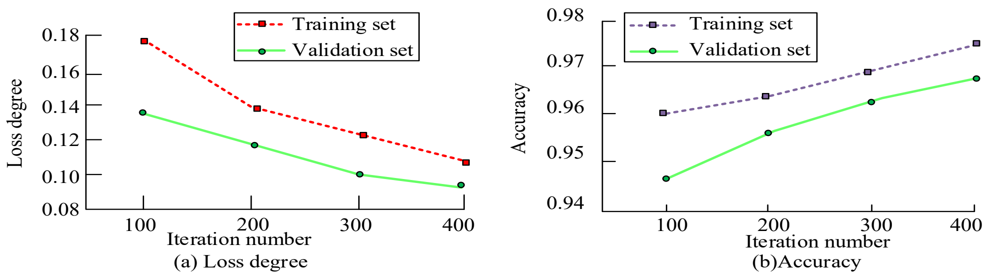
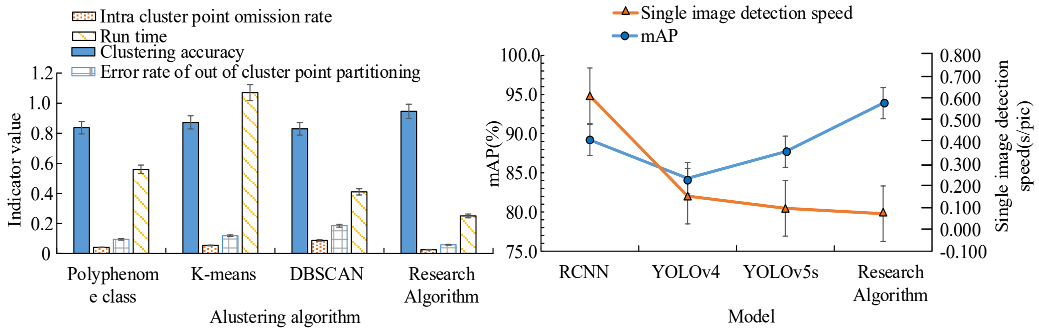


Fig. 8. Model Training Results

In Fig.8 (a), the loss of the research method in the two sets gradually decreases with the iteration of learning times. When the last training ended, the loss rates in the training and validation sets decrease from 0.1800 to 0.1084, and 0.1362 to 0.0915, indicating a continuous growth in generalization ability. In Fig.8 (b), the research model achieves a prediction accuracy of over 90% on both sets, and as the iterations increase, the final accuracies are 96.93% and 98.21%, respectively.

The paper uses the Clustering Accuracy (CA), Intra-Cluster Point Omission Rate (ICPOR), and Out Of Cluster Point Partitioning Error Rate (OOCPPER) as evaluation indicators and compares the clustering effect of the proposed algorithm with spectral clustering, K-means, and traditional DBSCAN algorithms, as shown in Fig.9 (a). This study also uses mean Average Precision (mAP) and Single Image Detection Speed (SIDS) as indicators, and Faster-RCNN, YOLOv4, and YOLOv5s as comparison algorithms to test the performance. Fig.9 (b) shows the results.



(a) The effectiveness of clustering algorithms (b) Recognition effects of different algorithms

Fig. 9. Algorithm Comparison Results

Fig.9 (a) compares clustering algorithms, with the improved DBSCAN achieving 94.6% CA, 2.5% ICPOR, and 5.8% OOCPPER, outperforming others by automatically adjusting the neighborhood radius based on clustering data features. Its runtime (0.25s) is 0.16s faster than traditional DBSCAN, demonstrating superior efficiency in non-spherical clustering. Fig.9 (b) highlights recognition performance, where the research algorithm reaches 94.1% mAP (4.6% higher than competitors) and 0.073s/spic SIDS, indicating better detection capabilities. Table 2 shows the actual and fitted parameters of OverMP obtained.

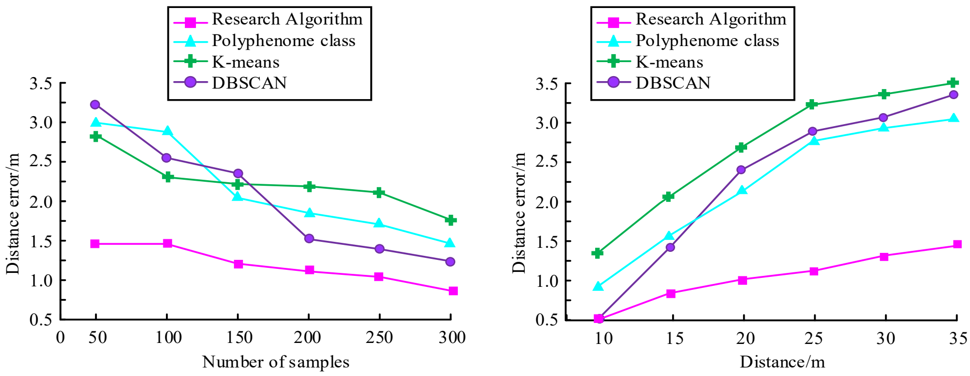
Table 2

Calculation results of OverMP actual parameters and fitting parameters

Figure serial number	Research algorithm			Traditional DBSCAN algorithm		
	Relative error of the center (%)	Relative error of the radius (%)	Relative deviation (%)	Relative error of the center (%)	Relative error of the radius (%)	Relative deviation (%)
1	0	3.92	0	2.76	3.92	2.89
2	1.47	0	0.87	3.28	4.41	1.93
3	1.81	29.49	1.43	1.81	28.21	1.43
4	4.08	4.08	3.39	8.65	0	7.19
5	4	2.53	2.35	4	1.27	2.35
6	6.56	14.71	4.86	27.74	26.47	20.56
7	15.69	13.73	12.45	19.98	17.65	15.86

8	5.42	3.95	3.27	8.82	5.26	5.32
9	5.98	3.33	3.69	20.76	2.22	12.79
10	10.53	10.53	6.84	47.05	36.84	30.56
Mean	5.554	8.627	3.915	14.485	12.625	10.088

The research algorithm exhibits superior performance in circle-fitting for mushroom colonies. Its calculated center position has a shorter distance and lower relative error (6.90% ARE) compared to the actual center than traditional algorithms (10.93% ARE). The fitted radius is also closer to the actual, with a 6.07% ARE versus traditional algorithms' 7.21%. Additionally, the radius deviation is 4.12%, 2.75% lower than traditional methods. Overall, the research algorithm significantly enhances accuracy and precision. The distance error under the above method is analyzed later, and the results are shown in Fig.10.



(a) Detection rate (b) Distance error

Fig. 10. Comparison Results of Distance Error

In Fig. 10(a), the ranging error of the optimization algorithm is significantly reduced, with its minimum distance error below 1.0m, while the minimum values of the other three algorithms are all greater than 1.25m. When the number of samples is less than 150, the error reduction rate of the optimization algorithm is slower compared to the other comparison algorithms; however, overall, the error values of the other three algorithms remain higher than those of the optimization algorithm. In Fig. 10(b), as the distance increases, the distance errors of the other three comparison algorithms show a more pronounced and rapid increase, with their maximum ranging errors all exceeding 3.0m. In contrast, the error of the research algorithm remains below 1.5m and is relatively less affected by distance.

To further verify the reliability of the proposed method, the publicly available datasets in literature [7] were used to compare the performance of each algorithm. Through the data preprocessing method described above, 5475 images were obtained, and the training set and test set were divided according to the ratio of 4:1. Of these, 1095 images were used for algorithm testing and 4380 images were used for algorithm training. The test results of each algorithm evaluation index are shown in Table 3.

Table 3

Test results of the different algorithms

Different algorithms	Algorithm cluster effect			Algorithm generalization ability	
	Cluster accuracy/%	Within-cluster point omission probability/%	Out-of-cluster point partition error rate/%	mAP/%	Detection speed/s
RCNN	81.4	9.7	8.3	78.9	0.306
YOLOv4s	79.8	11.2	5.7	82.1	0.142
YOLOv5s	82.4	4.9	6.8	85.3	0.223
Research algorithm	95.2	1.9	4.0	93.8	0.065

Table 3 reveals the proposed algorithm's notable advantages. It achieved 95.2% clustering accuracy, outperforming others, with only 1.9% omission and 4.0% error rates, ensuring comprehensive and precise mushroom group detection. It also boasts high generalization (93.8% mAP) and fast speed (0.065s).

Then, the study selected the widely recognized open dataset " FungalGrowth-2023 Dataset" in the field of agricultural robotics (containing 3D point cloud data for 180 real greenhouse mushrooms, covering different density distributions and lighting conditions, with each cluster of mushrooms labeled with the optimal picking path and priority). The performance of the improved DBSCAN algorithm compared to traditional DBSCAN, Mean-Shift Clustering (Mean-Shift), and Gaussian Mixture Model (Gaussian Mixture Model) was evaluated in terms of clustering accuracy, path optimization rate, noise handling capability, and real-time performance. The test results are shown in Table 4.

Table 4

Comparison of the average performance of each algorithm on the data set (n=180)

Algorithm	Clustering accuracy (%)	Path optimization rate	False acceptance rate (%)	time consuming (ms)
improve DBSCAN	94.2	1.12	1.5	320
tradition DBSCAN	86.7	1.31	4.8	285
Mean-Shift	79.4	1.45	8.3	410
GMM	82.1	1.38	6.9	375

Table 4 shows the improved DBSCAN excels in clustering accuracy (7.5% better than traditional DBSCAN) and path optimization (1.12x theoretical optimum). It adapts to mushroom density with a 1.5% false detection rate, meeting real-time needs post-optimization, and outperforms rivals by 14.8% in dense clusters.

3.2 System Integration Testing Analysis

To further verify the reliability of the proposed method, this study integrates various modules based on the Robot Operating System (ROS) platform to achieve data communication between modules. Then, this study conducts whole machine harvesting experiments on actual mushroom planting bases. 182 mushroom groups are tested for their three-dimensional localization. The number of successfully located mushroom clusters reaches 173, while only 9 fails, indicating a SR of 95.4%. Further analysis of cases of failed positioning reveals that the positioning error of 5 mushroom clusters is controlled within a range of 3-5mm. This indicates that although there are a few cases of positioning failure, the overall accuracy of positioning is still quite high. Table 5 shows the positioning of selected mushroom populations.

Table 5

Localization results of mushroom groups

The number of mushroom group	Identify the coordinates at/mm	Absolute error	Positioning results
1	(115.2,167.8,24.3)	(0.7,0.9,1.5)	√
2	(179.7,96.4,25.9)	(1.6,1.8,1.1)	√
3	(212.4,100.6,24.1)	(0.8,0.9,0.9)	√
4	(231.8,212.4,26.8)	(3.4,2.4,0.4)	×
5	(261.1,45.6,27.4)	(0.4,0.4,2.3)	√
6	(289.4,86.1,26.2)	(1.2,1.2,0.9)	√
7	(300.3,110.4,25.7)	(1.1,1.8,0.5)	√
8	(321.9,143.3,28.3)	(3.2,0.8,1.4)	×
9	(379.5,189.2,26.5)	(0.5,0.9,0.6)	√
10	(390.7,120.5,27.4)	(2.5,1.4,2.2)	×
11	(423.1,140.8,23.9)	(0.4,0.6,0.1)	√
12	(445.6,110.2,25.2)	(1.3,1.5,0.7)	√

After system integration, the robot's 3D positioning error is <2mm, with a 95.4% positioning SR. Even with 3-5mm errors, the adaptive suction cup enables successful picking, meeting MHR design goals and aiding harvest planning.To comprehensively assess the robot’s performance, this study also compares the recognition accuracy, picking SR, Average Picking Time (APT), and picking efficiency of different regions during the picking process, as shown in Fig.11.

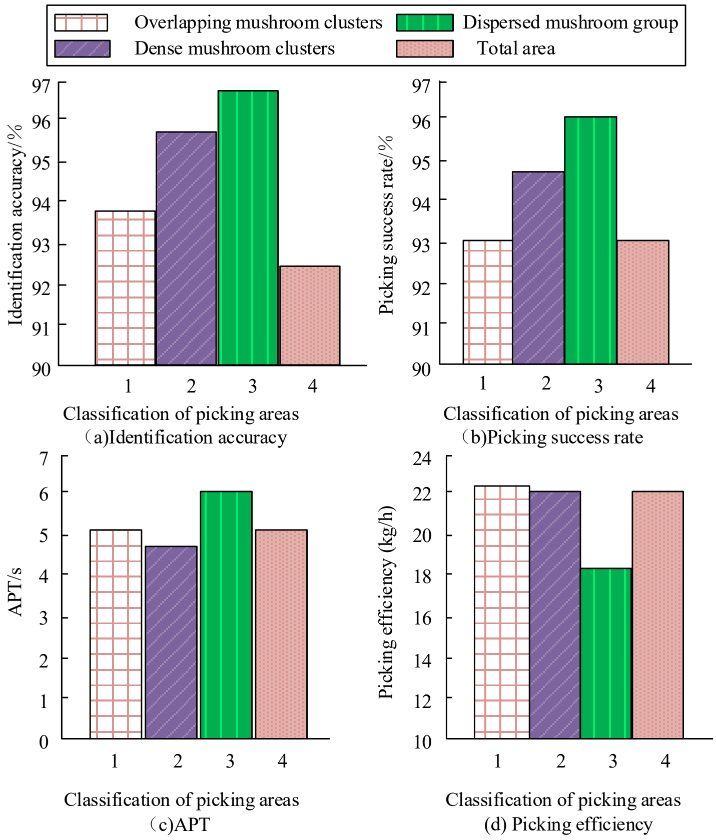


Fig. 11. Comparison Results of Data in Various Picking Areas

For the overlapping and DenseMP areas with high picking difficulty and damage rate, the robot has an accuracy rate of 93.8% and 95.8% for recognizing mushroom clusters, a SR of 93.2% and 94.7% for picking, and an APT of 4.52s and 4.15s for mushroom clusters. This proves that the harvesting algorithms designed for OverMP and DenseMP have good performance, improving the SR of mushroom harvesting, reducing the rate of harvesting damage, and ensuring that the harvesting efficiency meets the design requirements. In the whole-range picking process, the robot estimates a SR of 92.9%, with an APT of 4.23s for mushroom clusters and a picking efficiency of 21.25 kg/h. This satisfies the design goals of the robot and enables automated selection of mushroom clusters in the factory building.

4. Conclusion

This study proposed a global-local optimization picking planning technique for DenseMP using an improved DBSCAN algorithm to enhance cluster mushroom harvesting. The model achieved prediction accuracies of 96.93% and 98.21% on training and validation sets. The improved DBSCAN had a CA of 94.6%, ICPOR of 2.5%, and OOCPPER of 5.8%, outperforming other algorithms. Its runtime was 0.25s, 0.16s faster than traditional DBSCAN. The mAP reached 94.1%, with an optimal SIDS of 0.073s/pc. The 3D positioning error was <2mm, and SR was 95.4%. For OverMP and DenseMP, cluster recognition accuracy was 93.8% and 95.8%, with SRs of 93.2% and 94.7%, and APTs of 4.52s and 4.15s. Overall, the robot had a 92.9% SR and 21.25 kg/h efficiency, though occasional cap damage occurred, prompting future research.

REFERENCES

[1] P. de Frutos, B. Rodriguez-Prado, J. Latorre, A Gravity Model to Explain Flows of Wild Edible Mushroom Picking: A Panel Data Analysis, Ecological Economics, Vol. 156, 2019, pp. 164-173.

[2] M. G. Mohanan, A. Salgaonkar, Robotic Mushroom Harvesting by Employing Probabilistic Road Map and Inverse Kinematics, BOHR International Journal of Future Robotics and Artificial Intelligence, Vol. 1, Iss. 1, 2021, pp. 1-10.

[3] K. K. Gameli, Towards Reduction of Postharvest Losses in Mushroom Farming in Ghana: Methods and Strategies, Social Sciences, Vol. 1, Iss. 2, 2020, pp. 34-44.

[4] S. Singhal, P. Rasane, S. Kaur, Mushroom Cultivation, Processing and Value-Added Products: A Patent Based Review, Recent Patents on Food, Nutrition & Agriculture, Vol. 10, Iss. 1, 2019, pp. 3-19.

[5] J. Ji, S. Du, M. Li, Design and Experiment of a Picking Robot for Agaricus Bisporus Based on Machine Vision, International Journal of Agricultural and Biological Engineering, Vol. 17, Iss. 4, 2024, pp. 67-76.

[6] J. Rong, P. Wang, Q. Yang, A Field-Tested Harvesting Robot for Oyster Mushroom in Greenhouse, Agronomy, Vol. 11, Iss. 6, 2021, p. 1210.

[7] A. Sujatanagarjuna, S. Kia, D. F. Briechle, MushR: A Smart, Automated, and Scalable Indoor Harvesting System for Gourmet Mushrooms, Agriculture, Vol. 13, Iss. 8, 2023, p. 1533.

[8] P. Cong, H. Feng, K. L. Lv, MYOLO: A Lightweight Fresh Shiitake Mushroom Detection Model Based on YOLOv3, Agriculture, Vol. 13, Iss. 2, 2023, p. 392.

[9] M. Huang, X. Jiang, L. He, Development of a Robotic Harvesting Mechanism for Button Mushrooms, Transactions of the ASABE, Vol. 64, Iss. 2, 2021, pp. 565-575.

[10] A. Recchia, D. Strelkova, J. Urbanic, A Prototype Pick and Place Solution for Harvesting White Button Mushrooms Using a Collaborative Robot, Robotics Reports, Vol. 1, Iss. 1, 2023, pp. 67-81.

[11] H. Shi, G. Xu, W. Lu, An Electric Gripper for Picking Brown Mushrooms with Flexible Force and In Situ Measurement, Agriculture, Vol. 14, Iss. 7, 2024, pp. 1181.

[12] C. Hubay, A. Geösel, N. H. Horváth, Technological Development of Automated Harvesting for Cultivated Button Mushroom Using Image Processing, Periodica Polytechnica Electrical Engineering and Computer Science, Vol. 68, Iss. 4, 2024, pp. 413-423.

[13] A. D. Arjun, S. K. Chakraborty, N. K. Mahanti, Non-Destructive Assessment of Quality Parameters of White Button Mushrooms (Agaricus bisporus) Using Image Processing Techniques, Journal of Food Science and Technology, Vol. 59, Iss. 5, 2022, pp. 2047-2059.

[14] B. Koirala, A. Zakeri, J. Kang, Robotic Button Mushroom Harvesting Systems: A Review of Design, Mechanism, and Future Directions, Applied Sciences, Vol. 14, Iss. 20, 2024, p. 9229.

[15] Z. Chen, X. Lei, Q. Yuan, Key Technologies for Autonomous Fruit- and Vegetable-Picking Robots: A Review, Agronomy, Vol. 14, Iss. 10, 2024, p. 2233.

[16] J. Wang, W. Song, W. Zheng, Spatial-Channel Transformer Network Based on Mask-RCNN for Efficient Mushroom Instance Segmentation, International Journal of Agricultural and Biological Engineering, Vol. 17, Iss. 4, 2024, pp. 227-235.
[17] A. Porichis, M. Inglezou, N. Kegkeroglou, Imitation Learning from a Single Demonstration Leveraging Vector Quantization for Robotic Harvesting, Robotics, Vol. 13, Iss. 7, 2024, p. 98.
[18] P. S. Coles, M. Mazin, G. Nogin, The Association Between Mushroom Sciarid Flies, Cultural Techniques, and Green Mold Disease Incidence on Commercial Mushroom Farms, Journal of Economic Entomology, Vol. 114, Iss. 2, 2021, pp. 555-559.
[19] B. Leichtmann, A. Hinterreiter, C. Humer, Explainable Artificial Intelligence Improves Human Decision-Making: Results from a Mushroom Picking Experiment at a Public Art Festival, International Journal of Human–Computer Interaction, Vol. 40, Iss. 17, 2024, pp. 4787-4804.
[20] S. Choudhuri, S. Adeniye, A. Sen, Distribution Alignment Using Complement Entropy Objective and Adaptive Consensus-Based Label Refinement for Partial Domain Adaptation, Artificial Intelligence and Applications, Vol. 1, Iss. 1, 2023, pp. 43-51.
[21] K. Yang, M. Zhao, D. Argyropoulos, Machine learning based framework for the detection of mushroom browning using a portable hyperspectral imaging system, Postharvest Biology and Technology, Vol. 219, Iss. 1, 2025, pp. 113247.
[22] J. Zhao, W. Zheng, Y. Wei, Machine vision-based detection of key traits in shiitake mushroom caps, Frontiers in Plant Science, Vol. 16, Iss. 1, 2025, pp. 1495305.
[23] L. Xie, J. Jing, H. Wu, MPG-YOLO: Enoki Mushroom Precision Grasping with Segmentation and Pulse Mapping, Agronomy, Vol. 15, Iss. 2, 2025, pp. 432.