Mechatronic Systems and Control, Vol. 53, No. 10, 2025
An Open Access Paper

DEEP LEARNING-BASED SEMANTIC SEGMENTATION AND RECOGNITION FOR AGRICULTURAL ROBOTS

Shengnan Gao,* Xiaoshun Li,* and Yingying Liu*

Abstract

The paper presents a deep learning-driven approach for semantic segmentation and recognition in agricultural robotics, enhancing the robots' scene comprehension and operational efficiency. It addresses challenges in complex farm environments by introducing an innovative model that merges high-quality UAV-captured agricultural imagery with an enhanced encoder–decoder structure. This integration facilitates fine-grained image segmentation and introduces an attention mechanism plus a semantic recognition module, thereby bolstering the robot's capability in crop identification, growth monitoring, and disease detection. To ensure the robustness of the model, a comprehensive dataset was meticulously compiled and balanced through extensive data gathering and preprocessing. The model design innovates by incorporating an advanced feature extractor with null space pyramid pooling and an attention mechanism. This design augments multi-scale feature representation and regional focus, tackling the varied and intricate nature of agricultural landscapes efficiently. Optimised with the Adam optimiser and trained using cross-entropy loss, the model underwent a meticulous training regimen to refine its performance. Evaluation outcomes highlight its excellence across key metrics: accuracy, recall, F1 score, and IoU, with additional gains observed post-application of test-time augmentation. These results affirm the method's efficacy and practical utility in advancing agricultural robotics. Consequently, this research significantly contributes to the smart evolution of agricultural machinery and charts a promising trajectory for future automation and precision agriculture endeavors.

Key Words

Agricultural robots, semantic segmentation, semantic recognition, deep learning, attention mechanism

* College of Intelligent Science and Engineering, Beijing University of Agriculture, Beijing 102206, China; e-mail: shengnan_gao,Yyngyingliu2000@outlook.com; xiaoshun_li@hotmail.com
Corresponding author: Xiaoshun Li

Recommended by Teng Li
(DOI: 10.2316/J.2025.201-0511)

1. Introduction

Agricultural robots, as a kind of intelligent agricultural production tools, are able to autonomously complete a variety of agricultural operations, such as plowing, harvesting, fertilising, weeding, spraying, and other agricultural tasks in complex agricultural environments, which has a broad application prospect and market potential. However, to realise autonomous operation, agricultural robots first need to accurately perceive and understand the surrounding agricultural scene, which is one of the core technologies of agricultural robots [1], [2].

The perception and understanding of agricultural scenes usually requires semantic segmentation and recognition of images of agricultural scenes, so as to obtain the structure and content information of agricultural scenes. The accuracy and efficiency of semantic segmentation and recognition directly affects the ability of navigation and localisation, path planning, and task execution of agricultural robots, which in turn affects the operation effect and safety of agricultural robots. Its importance is specifically shown in Fig. 1 [3], [4].

However, semantic segmentation and recognition of agricultural scenes face many challenges, such as the complexity, diversity, dynamics, light changes, occlusion, noise, *etc.* of agricultural scenes, which leads to the traditional manual feature-based semantic segmentation and recognition methods are difficult to adapt to the characteristics of agricultural scenes.

Semantic segmentation is an advanced technique in the field of computer vision that aims to achieve precise categorisation of each pixel in an image. Unlike general image segmentation techniques that only divide an image into several chunks or regions, semantic segmentation is more detailed and is dedicated to assigning a predefined category label to each pixel within an image to differentiate the semantic meanings of different objects and regions in the image. For example, in an image of a city street, semantic segmentation can label the pixels of different objects, such as sidewalks, vehicles, buildings, trees, *etc.*, separately, and each type of pixel is assigned with a corresponding category label, thus generating a rich pixel-level labeled map.

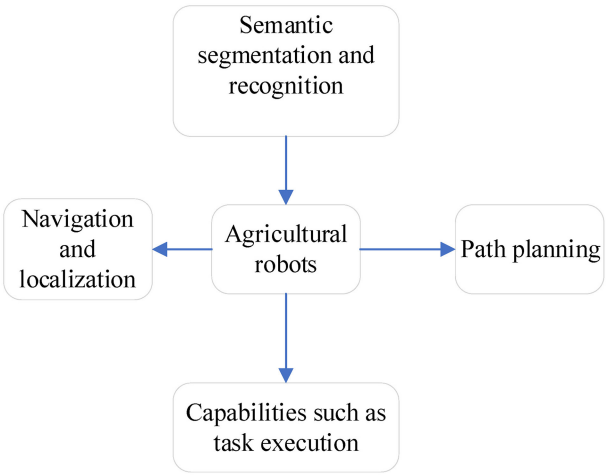

Figure 1. Importance of semantic segmentation for agricultural robots.

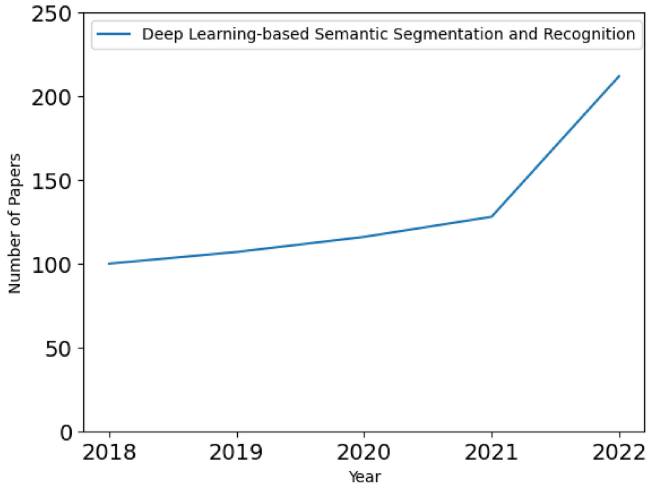

Figure 2. Number of relevant studies.

Research objective of paper is to explore the method of semantic segmentation and recognition of agricultural robots to provide technical support for scene perception and understanding of agricultural robots. The research idea of this paper is to realise semantic segmentation and recognition of agricultural scenes based on the images of agricultural scenes collected by UAVs, using models and algorithms of deep learning [5], [6].

The innovative application of deep learning methods in the field of semantic segmentation and recognition of agricultural robots not only improves the autonomous operation capability of robots, but also provides a new way for accurate management of agricultural production. Specific contributions include: developing a deep learning model customised for agricultural scenarios, which improves the recognition accuracy and segmentation effect in complex and changing agricultural environments through the optimised encoder–decoder structure combined with the attention mechanism; constructing a large-scale, diversified, and balanced agricultural image dataset, which provides a solid foundation for model training and testing; proposing a set of efficient model training and testing An effective model training and testing strategy is proposed, which significantly enhances the generalisation ability and robustness of the model; experiments prove that the method in this paper demonstrates high accuracy and efficiency in key agricultural applications, such as crop recognition, pest and disease detection, *etc.* In particular, the application of the enhancement method during the test further confirms the applicability of the model's advantages in complex scenarios.

2. Literature Review

In recent years, the number of related studies has gradually increased, and the specific trend is shown in Fig. 2. Ghosh *et al.* [7] designed a deep learning-based fruit and vegetable recognition system, using convolutional neural networks (CNNs) which provides a reliable technical means for intelligent sorting, quality detection, and variety classification of fruits and vegetables. Gnanapriya and Rahimunnisa's [8] comprehensive review in the Journal of Agricultural Machinery delves into the advancements of deep learning-driven visual navigation systems for agricultural robots. They meticulously dissect the task requirements, datasets, methodologies, and evaluation metrics underpinning robot vision navigation, subsequently consolidating the prevailing challenges confronted in the field. Furthermore, they chart out prospective avenues for future research and development to advance the capabilities of visual navigation systems in agricultural robotics. An innovative design has attracted a lot of attention. The research team successfully combined advanced image processing technology with a convenient hardware platform to open up a new communication channel for severely paralysed patients. Specifically, they adopted Raspberry Pi as the main processor, giving full play to the flexibility and powerful computing power of this microcomputer in the field of IoT, while the choice of Python programming language greatly simplifies the software development process, making the algorithm implementation more efficient and faster. The 5-megapixel high-definition camera integrated in the device demonstrates the high sensitivity of precisely tracking eye movements, so that even the most subtle eye movements can be accurately captured and converted into effective communication signals [9].

The innovative application of deep learning techniques in agriculture has become a focus of research in recent years. Numerous scholars have actively explored the integration of CNN models with transfer learning methods in crop classification and recognition tasks, and these studies have shown that deep learning has demonstrated unprecedented efficacy in processing agricultural image data to accurately differentiate between crop species, assess crop health, and even identify signs of pests and diseases, thus highlighting its potential for widespread application in agricultural image analysis [10]. In addition, the evolution of agricultural robotics has seen deep learning-driven vision navigation systems become key to enhancing automation. These systems are able to analyse complex farmland environments in real time, assisting robots to achieve

precise positioning and path planning in cultivation, seeding, and pesticide spraying [11, 12].

However, the literature also reveals several challenges in the application of deep learning in agricultural environments, especially those factors, such as complex terrain, variable lighting conditions, crop growth cycle changes, and unstructured scenarios put the robustness and generalisation ability of the models to the test [13]. These difficulties indicate that although deep learning models are theoretically equipped with powerful image understanding and processing capabilities, they need to overcome the environmental adaptability problem in real agricultural deployments to meet the stringent requirements of production practices. Therefore, exploring how to design more robust and adaptable deep learning models that can accurately segment images of agricultural scenes and recognise key agricultural elements has become an urgent need and a source of motivation for current research. Based on such research background, this paper aims to propose an innovative deep learning approach with a view to solving the above challenges and advancing the intelligentization process of agricultural robots.

3. Data and Research Methods

This chapter describes in detail the method of semantic segmentation and recognition of agricultural robots based on deep learning proposed in this paper, including the steps of data acquisition, preprocessing, model design, training, and testing, and gives the necessary mathematical formulas, algorithmic processes, and graphical illustrations.

3.1 Data Acquisition

In order to acquire large-scale agricultural scene image data, we adopts a UAV as a platform for data acquisition, and utilises the HD camera carried by the UAV to photograph the agricultural scene from different heights, angles and locations, and acquires agricultural scene images under a wide range of crops, terrains, seasons, and lighting conditions. In this paper, a total of 10,000 agricultural scene images were acquired, each with a resolution of 1920 × 1080 pixels, covering major crops, such as rice, wheat, corn, cotton, and vegetables, as well as major features, such as soil, water, weeds, and roads. Image, an open source Python image enhancement library, is used to randomly apply one or more methods of data enhancement to each image while maintaining the consistency of the image and the annotation [13]. We ensure the balance and representativeness of the distribution of image sources and categories in the training, validation and testing sets. The data processing flow is specifically shown in Fig. 3 [14], [15]. The data set information is specifically shown in Table 1.

3.2 Modelling

The model design of this paper mainly includes two parts, encoder and decoder, which are used for feature extraction and feature recovery respectively, as shown in Fig. 4.

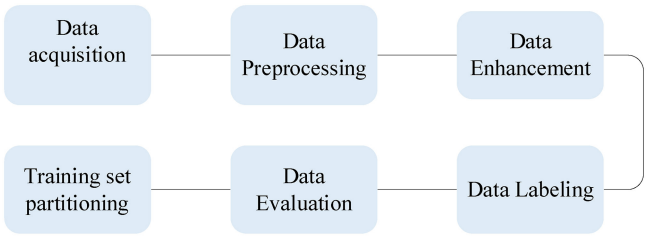

Figure 3. Data processing flow.

Table 1
Data Set Information

Data set	Number of images	Percentage
Training set	80000	80%
Validation set	10000	10%
Test set	10000	10%

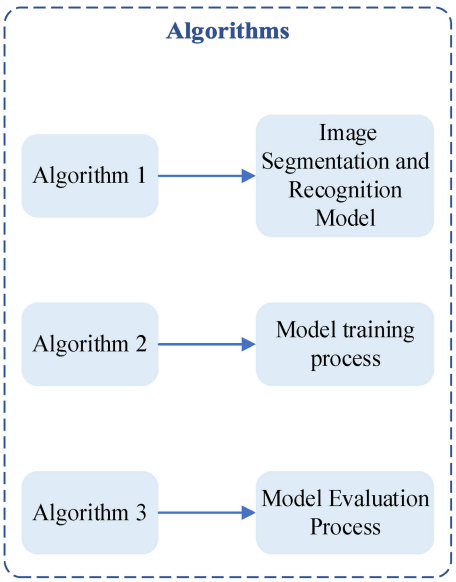

Figure 4. Algorithm flow.

In this paper, the design of the model is centered around two primary components, the encoder and decoder, which are pivotal for feature extraction and feature reconstruction in the context of agricultural scene image segmentation and recognition (depicted in figure. 4). The model begins by processing an agricultural scene image, I, rich in diverse visual cues, such as crop varieties, growth phases, soil health, and pest/disease incidences, captured through high-resolution aerial or satellite imaging. Key enhancements to the conventional architecture involve: (1) Bilinear interpolation for upsampling: Abandoning the traditional transposed convolution in favour of bilinear interpolation, represented as, to mitigate checkerboard artifacts and enhance the segmentation map's smoothness. (2) Attention mechanism integration: Introduction of an attention mechanism to weigh shallow features, allocating weights W based on their relevance () to emphasise target

regions, suppress noise, and boost segmentation accuracy. (3) Semantic recognition module (G): An attention-guided module designed to classify key agricultural elements post-segmentation, refining outputs like crop classifications, growth stages, and pest/disease detection, thereby enriching semantic interpretation and practical applicability.

This paper designed an improved decoder structure for the task of semantic segmentation and recognition of agricultural scene images. First, the input agricultural scene image is notated as I. After the features are extracted by the encoder, the main function is to map the encoded feature space information back to the pixel space [16], [17]. The agricultural scene image, I, constitutes a pivotal element in this study. It encapsulates a rich variety of visual data crucial for agricultural management and research, including but not limited to crop types, growth stages, soil conditions, and evidence of diseases or pests. These high-resolution images are typically captured using aerial or satellite platforms, ensuring broad coverage of agricultural land with fine detail.

For the original network structure, the following points are optimised in this paper:

(1) In the upsampling process, the transposed convolution operation used in the original network is discarded and replaced by Bilinear Interpolation, denoted by the formula $S_{up} = \text{BilinearInterpolate}(F_l)$, which aims to eliminate the checkerboard effect that may be caused by the transposed convolution, thus improving the smoothness and clarity of the segmentation map [18], [19].

(2) Introducing the attention mechanism to weight the shallow features, and assigning different weights W according to the importance of the location, which can be expressed as $F_{att} = F_{\text{sshallow}} * W$, which helps to highlight the target region, suppress background noise, and enhances the accuracy and robustness of the segmentation map.

In the deep learning model designed for this study, a series of critical steps have been employed to achieve refined processing and understanding of agricultural scene images. Initially, bilinear interpolation is utilised to upsample feature maps, generating a high-resolution segmentation map S_{up}, effectively alleviating the common checkerboard artefact associated with transposed convolutions and enhancing the smoothness and clarity of the segmentation output. Within the model pipeline, low-resolution feature maps F_l obtained from preceding layers undergo transformation to yield high-resolution feature maps F_s at the current layer, thereby augmenting the hierarchical and detailed representation of features.

To further enhance the model's focusing capability and recognition precision in complex scenarios, an attention mechanism is incorporated. This mechanism assigns attention weights W to shallow features based on their significance within the image, allowing for adaptive emphasis on vital regions while suppressing background noise through a weighted strategy. Consequently, the resultant weighted feature map F_{att} under the influence of the attention mechanism accentuates targeted areas, concurrently suppressing irrelevant background interference, and markedly improves both the accuracy and robustness of the segmentation. This refined segmentation paves the way for more precise subsequent tasks, such as crop identification, growth monitoring, and disease and pest detection.

(3) A semantic recognition module G based on the attention mechanism is designed, which can further accurately recognise and classify the key targets in the agricultural scene, such as crop types, growth status, pests and diseases, based on the results of the initially obtained segmented image S, providing richer semantic information and practical application value, and whose mathematical representation [20], [21].

Module G, an attention-based semantic recognition module, is designed to enhance the precise identification and classification of key objects in agricultural scenes. The module receives inputs from a preceding image segmentation process, resulting in S, which comprises distinct regions annotated for elements like crops, soil, and diseased areas. Each region is effectively a set of feature vectors embodying its visual characteristics. Every segmented region undergoes further processing *via* deep learning networks, such as parts of a CNN, to distill more abstract and discriminative feature representations. This yields a collection $\text{F} = \{\text{f}_1, \text{f}_2, \ldots, \text{f}_n\}$, where n denotes the number of regions and each f_i symbolises the feature vector for the i-th region. Central to this module is a weighting scheme that autonomously assigns varying degrees of importance to different feature regions. This is accomplished by calculating similarity scores between each feature f_i and a context vector, subsequently transforming these through a softmax operation into normalised attention weights α_i. This mechanism prioritises salient regions while discounting background noise. The calculated attention weights α_i are then used to perform a weighted summation of the features F, generating a composite feature vector c. This step underscores critical regional information.

(4) The output of the algorithm is the segmented image S and the recognition result R, which together constitute the key result of the in-depth understanding and parsing of the input agriculture scene image I. Model for segmentation and recognition of agricultural scene images, whose input is an agricultural scene image $I \in \mathbb{R}^{H\times W\times 3}$ and the output is the segmented image $S \in \mathbb{R}^{H\times W\times C}$ and the recognised key target $R \in \mathbb{R}^{K\times D}$.

In the context of the article, the formulas mentioned include: $I \in \mathbb{R}^{H\times W\times 3}$ represents the input scene image with dimensions HxWx3. $S \in \mathbb{R}^{H\times W\times C}$ denotes the output segmented image with dimensions H×W×C. $R \in \mathbb{R}^{D}$ stands for the recognised target with D dimensions. $F \in \mathbb{R}^{h\times w}$ is the low-resolution feature map obtained from the input image through feature extraction using MobileNetV2. $F' \in \mathbb{R}^{h'\times w'\times c'}$ is the fused feature map obtained from the multi-scale processing of the low-resolution feature map F using the ASPP module. These formulas represent various aspects of the model, such as the input image, the output segmentation result, intermediate feature maps, and the recognised target. They

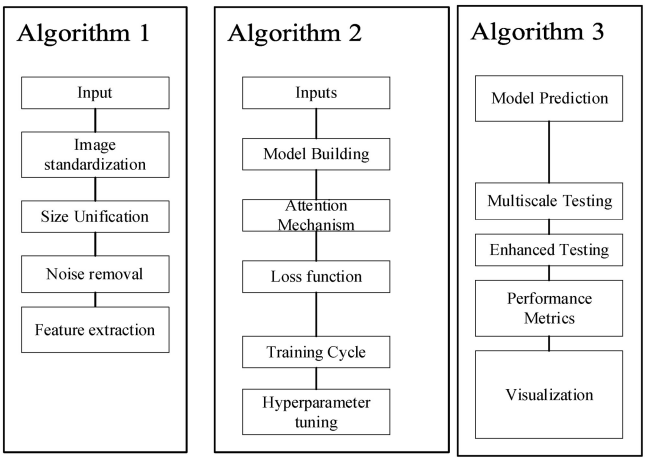

Figure 5. Flowchart of the algorithm.

provide a concise summary of the model's architecture and functionality.

(5) The process is mainly realised by Algorithm 1 and the flow of the algorithm appearing in the paper is shown in Fig. 5. The design flow of its model is as follows:

The specific flow of Algorithms 1, 2, and 3 is shown in Fig. 5.

1) MobileNetV2 is a lightweight network model designed specifically for mobile and embedded devices, which utilises depthwise separable convolutions to dramatically reduce the amount of computation and the number of parameters of the model, thereby reducing the demand for computational resources and memory consumption while ensuring relatively high accuracy, which is ideal for resource-constrained environments. It is very suitable for resource-constrained environments. Using MobileNetV2 as the backbone network, feature extraction is the input image to obtain the low resolution feature map $F \in \mathbb{R}^{h \times w}$. The formula is $F = MobileNetV2(I)$ [22].

2) Add a void space pyramid pooling module, process F in multi-scale, and get the fused feature map $F' \in \mathbb{R}^{h \times wc'}$. The formula is $F' = ASPP(F)$. The main objective of ASPP is to solve the problem of recognising objects at different scales in semantic segmentation. In natural scenes, target objects may appear at many different sizes, and standard convolutional operations with a fixed sense field may not be sufficient to efficiently capture features at all scales. ASPP addresses this challenge by introducing Atrous Convolution (also known as inflationary convolution) and spatial pyramid pooling. ASPP stands for Atrous Spatial Pyramid Pooling. It is a technique integrated into deep learning models, primarily designed for semantic segmentation tasks, to effectively handle objects of varying scales present within an image. Unlike standard convolutions with fixed receptive fields that might inadequately cover multi-scale features, ASPP employs Atrous Convolution, which adjusts the filter's dilation rate, allowing it to capture context at multiple rates without increasing the filter size. Complementing this, it also incorporates spatial pyramid pooling, a strategy that pools features at different scales and regions, thereby aggregating local and global contextual information comprehensively. Consequently, ASPP enhances the model's capability to discern and segment objects of diverse sizes more accurately by integrating multi-scale feature extraction within a single module.

3) Use the attention mechanism to weight the fused feature map F'. The weighted feature map $F'' \in \mathbb{R}^{h \times wc'}$ is obtained, where $W \in \mathbb{R}^{h \times wc'}$ is the attention weight matrix and $\bigodot$ is the element-by-element multiplication operation. The formula is $F'' = W \bigodot F'$.

4) Using an upsampling module and a convolutional layer, the weighted feature map, where UPSAMPLE is the function of upsampling and Conv is the function of convolution. The formula is $S' = Conv(\text{Upsample}(F''))$.

The formulas mentioned in the article involve several key steps in the model design process: In the first step, a lightweight network model called MobileNetV2 is used as the backbone network to extract features from the input image I, resulting in a low-resolution feature map F' with dimensions h×w×wc′. The formula is F = MobileNetV2(I). A spatial pyramid pooling module is then applied to the low-resolution feature map F' to obtain a fused feature map F' with dimensions h′×w′×c′. The formula is F′ = ASPP(F).An attention mechanism is used to weight the fused feature map F'. The weighted feature map F'' with dimensions h×wc′ is obtained, where W is the attention weight matrix and ⊙ is the element-by-element multiplication operation. The formula is F″ = W⊙F′.Finally, an upsampling module and a convolutional layer are used to generate the final segmentation map S' with dimensions h×w×c. The formula is S′ = Conv(Upsample(F″)).

3.3 Training

This study uses high-end hardware configurations to ensure that the deep learning model runs efficiently and handles complex tasks, including a high-performance multi-core CPU with at least 16 cores (*e.g.*, Intel Xeon or AMD Ryzen Threadripper), 128GB or more of RAM, and NVIDIA's high-end GPUs (*e.g.*, RTX 3090/A6000) to accelerate the computation. The NVMe SSDs are equipped with 1TB NVMe SSDs for high-speed data access, supplemented with 4K monitors and high-quality peripherals to optimise workflow, and rely on a network environment of more than 100 Mbps to support data transmission and distributed training needs, ensuring smooth execution of the model from training to application in an all-rounded way.

The model training mainly includes the setting of parameters such as loss function, specifically as in (1).

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} X_{ij} \log F_{ij} \tag{1}$$

Where X_{ij} is the true label of the i_{th} pixel point belonging to the j_{th}category, and F_{ij} is the predicted probability that the i_{th} pixel point belongs to the j_{th} category [23], [24].

Optimiser: an optimiser is an algorithm used to update the model parameters to minimise the value of the loss function and improve the optimisation ability of the model.

Table 2
Parameter Settings

Parameters	Retrieve a value
monitor	val_loss
factor	0.1
patience	10
min _lr	0.00001
Batch_size	16
Number of iterations	50

This paper used an adaptive optimiser that dynamically adjusts the learning rate according to changes in the gradient, with the advantages of fast convergence and stability [25], [26].

The following parameter settings are used specifically as shown in Table 2. In the model design and training phase, parameter selection was based on an exhaustive experimental validation and tuning process. For example, the batch size of 16 was determined as a result of balancing the computational resource constraints with the speed of model convergence; the number of iterations was set to 50 rounds based on the observation that the model had reached the performance saturation point at this stage, and more iterations did not result in a significant improvement. In addition, hyper parameters such as the learning rate were carefully tuned through grid search and cross-validation methods to ensure the model's optimal performance on the training set and generalisation ability on the validation set.

We take the approach of adjusting the learning rate periodically, *e.g.*, for every certain number of epochs (*e.g.*, 20 or 50 epochs), the learning rate is automatically reduced by a certain percentage (commonly 0.1 or 0.5). This method is called "Step Decay" (Step Decay), which helps to learn the global structure quickly at the initial stage, and then move on to more detailed local optimisation. Another advanced strategy is Exponential Decay, where the learning rate decreases exponentially according to a pre-determined decay rate, ensuring that the pace of learning slows down as training progresses, and helping the model to fine-tune the parameters at later stages of training to avoid overfitting. In addition, we may also implement a more dynamic tuning strategy—"learning rate decay scheduler"—especially based on performance monitoring, *e.g.*, "ReduceLROnPlateau" strategy. This strategy monitors the loss or accuracy on the validation set and adjusts the learning rate downward only when the validation loss does not improve significantly for a number of consecutive rounds (*e.g.*, 3 or 5 rounds). This strategy can respond more flexibly to the actual convergence of the model and avoid unnecessary learning rate drops, thus finding a good balance between model optimisation and training efficiency.

The overall flow of model training is shown in Algorithm 2. (1) The design process of Algorithm 2 is as follows: initialise the model parameters θ, these parameters include the weights and biases of the backbone network, the null-space pyramid pooling module, the attention mechanism, the up-sampling module, the convolutional layer, and the semantic recognition module. The formula is θ = Initalize(). (2) Set the loss function L, optimiser O, learning rate tuner S, batch size B, and number of iterations E. Specifically, (2)–(5) [27], [28]. Initalise is an initialisation function, Optimiser is an optimisation function, and S is the learning rate tuner. θ optimiser content parameter.

$$L = \text{LossFunction}() \tag{2}$$
$$O = \text{Optimizer}(\theta, S) \tag{3}$$
$$B = \text{BatchSize}() \tag{4}$$
$$E = \text{Epochs}() \tag{5}$$

L = Loss function, measure the difference between prediction and actual, guide the weight update, such as mean square error, cross-entropy loss. O = Optimiser, based on the loss update parameters, accelerate convergence, commonly used gradient descent, Adam, *etc.* B = Batch size, affect the training speed and stability, need to choose the right amount. E = Epochs, the number of data traversal, to ensure that the model learns sufficiently, can be used with the early stop strategy.

(3) For each round of iteration e, from 1 to E, the following steps are performed: The training set D_{train} is randomly disrupted and divided into several batches according to the batch size B, and each batch D_{batch} contains B samples, and each sample contains an image of the agricultural scene I and the corresponding real segmentation image S and recognition result R. Specifically (6) and (7).

$$D_{\text{train}} = \text{Shuffle}(D_{\text{train}}) \tag{6}$$
$$D_{\text{batch}} = \text{Spilt}(D_{\text{batch}}, B) \tag{7}$$

For each batch D_{batch}, perform the following steps: Input the batch D_{batch} into the model M to get the predicted result P, where P contains the predicted segmented image S and the recognised result R. As shown in (8) and (9), where Y contains the true segmented image S and the recognition result R. Specifically (10) and (11) [29], [30].

$$P = M(D_{\text{batch}}) \tag{8}$$
$$P = \{S', R'\} \tag{9}$$
$$Y = \{S, R\} \tag{10}$$
$$L = L(P, Y) \tag{11}$$

Specifically as in the public $\theta = O(\theta, L)$. (4) Return the trained model M. The formula is as follows: $M = M(\theta)$, this formula is used to update M.

Equation (8) represents the predicted output of the model M on a batch of data D_{batch}. Here P is the prediction result of the model, which is usually a probability distribution, while M represents the model itself, which maps the input data to the output space. Equation (9) states that P contains two components, S' and R'. Here S' may denote some intermediate level output of the model,

while *R'* may denote the final output of the model, such as classification results or regression values. Equation (10) indicates that the true label Y also contains two parts, S and R. Here, S may represent auxiliary information such as prior knowledge of the image, while R may be the desired model output such as the correct category label or regression value. Equation (11) L = L(P,Y) indicates that the loss function L is a measure of the distance between the model prediction P and the true label Y. This loss function is usually a measure of the gap between the model prediction and the actual value, such as cross-entropy loss or mean square error.

4. Testing of Models

The model testing in this paper mainly includes the steps of test data input, model prediction, test result output, and evaluation, as well as the model testing process and strategy. The test set D-Test is used as the test data, *i.e.*, 10,000 images of agricultural scenes and their corresponding labeled images. The prediction result P of the model is compared with the real result Y. Evaluation metrics, such as accuracy, recall, F1 value, and IOU of the model on the test set are calculated to measure the performance and generalisation ability of the model. The model's prediction result P and the real result Y are also visualised to show the segmentation effect and recognition effect of the model to intuitively reflect the strengths and weaknesses of the model. The following testing strategies are used to improve the testing effect and stability of the model: (1) The sliding window method is used to cut the test image, *i.e.*, the test image is split into a number of subgraphs of the same size, and then each subgraph is input into the model to make predictions, and finally the prediction results are spliced into a complete segmentation graph to avoid the effect of scaling or cropping of the image on the segmentation effect. (2) Multi-scale testing method is used to process the test images, *i.e.*, the test images are scaled to different scales, and then the images of each scale are input into the model for prediction, and finally the prediction results are fused into the final segmentation map. (3) The test-time enhancement method is used to transform the test image, *i.e.*, the test image is transformed by horizontal flipping, vertical flipping, and rotation, and then each transformed image is inputted into the model for prediction, and finally the prediction results are fused into the final segmentation map [31].

The general flow of model testing is shown in Algorithm 3.

(1) Initialise the test result T as empty. (2) Cut each test image I and its corresponding labeled image. Y using the sliding window method to obtain several subgraphs I_i. The formulas are $I_i = \text{Slide}(I)$ and $Y_i = \text{Slide}(Y)$. (3) Scale each subimage I_i using the multi-scale test method to get different scales of subimages I_{ij}. The formula is $I_{\text{ij}} = \text{Scale}(I_i)$. (4) Transform each subgraph I_{ij} using the test-time enhancement method to obtain different transformed subgraphs I_{ijk}. The formula is $I_{\text{ijk}} = \text{Transform}(I_{\text{ij}})$. (5) Input each subgraph I_{ijk} into the model M to get the prediction result P_{ijk}, where P_{ijk} contains the predicted segmented image S_{ijk} and the recognition result R_{ijk}. The formulas are $P_{\text{ijk}} = M(I_{\text{ijk}})$ and $P_{\text{ijk}} = \{S_{\text{ijk}}, R_{\text{ijk}}\}$. (7) Fuse each prediction result P_{ij} to get the original size prediction result P_i. The formula is as follows: $P_i = \text{Fuse}(P_{\text{ij}})$. (8) Each prediction P_i is spliced to obtain the complete prediction P. The formula is $P = Stitch(P_i)$. (10) Add the prediction result P and evaluation result E to the test result T. The formula is $T = T \cup \{P, E\}$. (11) Return the test result Return T [32], [33].

During the testing phase of Algorithm **??**, in order to comprehensively evaluate and enhance the generalisation performance and robustness of the model, we not only executed standard testing procedures, but also incorporated advanced strategies to simulate and cope with real-world diversity and complexity. Below are additional descriptions of several key approaches:

Multi-Scale Testing strategy refers to applying different scale transformations to the same test image in the inference stage, and then fusing the outputs of the models at different scales or selecting the optimal results. To implement this, the images can be scaled up and down to a series of predefined scales (*e.g.*, 0.5x, 1x, 1.5x, 2x, *etc.*), respectively, and then the model is run for images at each scale, and then these predictions are finally merged according to some rule (*e.g.*, averaging, max-voting, or weighted fusion). This approach helps the model to capture features at different scales, and is especially suitable for agricultural scenarios where the target size varies greatly, such as morphological differences in crops during different growing periods.

Test-time augmentation (TTA) is a technique that generates multiple variants of an image by applying a series of random transformations (*e.g.*, rotation, flipping, brightness change, scaling, *etc.*) to the image during the test phase, makes predictions for each variant, and then summarises the results of these predictions. Similar to data augmentation during training, TTA enables the model to see more variants without increasing the training complexity, thus enhancing the model's ability to generalise to unseen samples. In agricultural scenarios, TTA is particularly beneficial in improving the robustness of the model due to uncertainties in lighting conditions, crop pose, occlusions, *etc.*

To further improve the accuracy and stability of the predictions, result fusion techniques can also be used in the testing phase. This includes, but is not limited to, the fusion of multi-scale predictions from the model itself, the fusion of multiple predictions generated by the TTA, or even the fusion of predictions from different models (*e.g.*, models with different hyper parameters or architectures). Selecting the best prediction can be done either by calculating the confidence score for each prediction, selecting the prediction with the highest score, or by using more complex fusion rules such as weighted averaging [34], [35] [36].

From Table 3, the test-time enhancement method is the most effective, the sliding window method is the second most effective, and the multi-scale testing method is the worst. This may be due to the fact that the test-time enhancement method can increase the diversity and difficulty of the data, making the model more adaptable

Table 3
Experimental Results

Mould	Accuracy	Recall rate	F1 value	IOU
M	0.92	0.88	0.90	0.75
M + Sliding Window Method	0.94	0.90	0.92	0.78
M + Multi-scale test method	0.93	0.89	0.91	0.77
M + test-time enhancement	0.95	0.91	0.93	0.79

to complex scene changes. The sliding window method can avoid the effect of scaling or cropping of the image on the segmentation effect, but it also increases the amount of computation and time [37].

5. Conclusion

This paper investigated a deep learning-based method for semantic segmentation and recognition of agricultural robots, which provides technical support for scene perception and understanding of agricultural robots. In this paper, a large-scale and diversified agricultural scene image dataset is constructed, covering a variety of crop types, terrain features, seasonal changes, and lighting conditions, and providing high-quality pixel-level annotation information, which fills the gap in data resources for the study of semantic segmentation of agricultural images. A complete framework for semantic segmentation and recognition of agricultural scenes based on deep learning is proposed, covering all aspects from data preprocessing to model design, training strategy selection and evaluation index setting, which provides systematic practical guidance for researchers in related fields. And an innovative lightweight semantic segmentation network is designed and implemented, which combines depth-separable convolution and null convolution techniques to effectively capture agricultural scene features at different scales. In addition, we introduce a semantic recognition module with an attention mechanism, which can dynamically focus on and categorise key agricultural targets, such as distinguishing crop species, judging growth stages and pest and disease conditions, according to the segmentation results, thus significantly improving the accuracy and robustness of the recognition task. We have conducted full experimental validation on a self-constructed agricultural scene image dataset, and the experimental results, which provides a strong technical support for promoting agricultural robots to perform accurate scene sensing and understanding.

Funding

This study was supported by R&D Programme of Beijing Municipal Education Commission (KM202310020001).

References

[1] S. Aly and W. Aly, DeepArSLR: A novel signer-independent deep learning framework for isolated Arabic sign language gestures recognition, *IEEE Access*, *8*, 2020, 83199–83212.

[2] S. Arshad, M. Shahzad, Q. Riaz, and M.M. Fraz, DPRNET: Deep 3D point based residual network for semantic segmentation and classification of 3D point clouds, *IEEE Access, 7*, 2019, 68892–68904.

[3] J.Y. Cha, H.I. Yoon, I.S. Yeo, K.H. Huh, and J.S. Han, Panoptic segmentation on panoramic radiographs: Deep learning-based segmentation of various structures including maxillary sinus and mandibular canal, *Journal of Clinical Medicine, 10*(12), 2021, 14.

[4] T. Cruz-Rojas, J.A. Franco, Q. Hernandez-Escobedo, D. Ruiz-Robles, and J.M. Juarez-Lopez, A novel comparison of image semantic segmentation techniques for detecting dust in photovoltaic panels using machine learning and deep learning, *Renewable Energy, 217*, 2023, 23.

[5] D. Di-Mauro, A. Furnari, G. Patanè, S. Battiato, and G.M. Farinella, Scene adapt: Scene-based domain adaptation for semantic segmentation using adversarial learning, *Pattern Recognition Letters, 136*, 2020, 175–182.

[6] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, A survey on deep learning techniques for image and video semantic segmentation, *Applied Soft Computing, 70*, 2018, 41–65.

[7] S. Ghosh, N. Das, I. Das, and U. Maulik, Understanding deep learning techniques for image segmentation, *ACM Computing Surveys, 52*(4), 2019, 35.

[8] S. Gnanapriya and K. Rahimunnisa, A hybrid deep learning model for real time hand gestures recognition, *Intelligent Automation and Soft Computing, 36*(1), 2023, 1105–1119.

[9] A. Kumar, J.J. Anand, and B.N.H. Kumar, Intrusive video oculographic device: An eye-gaze-based device for communication, *Innovation and Emerging Technologies*, *28*(2) 2022, 9

[10] Y.M. Guo, Y. Liu, T. Georgiou, and M.S. Lew, A review of semantic segmentation using deep neural networks, *International Journal of Multimedia Information Retrieval, 7*(2), 2018, 87–93.

[11] Y.R. Guo and T. Chen, Semantic segmentation of RGBD images based on deep depth regression, *Pattern Recognition Letters, 109*, 2018, 55–64.

[12] Y.M. Zhang, J. Sun, and J.Y. Qiao, Evaluation on Chinese agricultural mechanisation level in high-quality development stage based on improved AHP-critic, *Mechatronic Systems and Control*, *52*(2), 2024, 121–129.

[13] X. Han, Z. Dong, and B.S. Yang, A point-based deep learning network for semantic segmentation of MLS point clouds, *ISPRS Journal of Photogrammetry and Remote Sensing, 175*, 2021, 199–214.

[14] W. Huang, Z.F. Shao, M.Y. Luo, P. Zhang, and Y.F. Zha, A novel multi-loss-based deep adversarial network for handling challenging cases in semi-supervised image semantic segmentation, *Pattern Recognition Letters, 146*, 2021, 208–214.

[15] R. Kemker, R. Luu, and C. Kanan, Low-Shot learning for the semantic segmentation of remote sensing imagery, *IEEE Transactions on Geoscience and Remote Sensing, 56*(10), 2018, 6214–6223.

[16] R. Kemker, C.S. Alvaggio, and C. Kanan, Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning, *ISPRS Journal of Photogrammetry and Remote Sensing, 145*, 2018, 60–77.

[17] H.K. Kim, K.Y. Yoo, J.H. Park, and H.Y. Jung, Traffic light recognition based on binary semantic segmentation network, *Sensors, 19*(7), 2019, 15.

[18] X.Y. Kong, X.H. Sun, Y.Z. Wang, R.Y. Peng, X.Y. Li, Y.H. Yang, and S.P. Tseng, Food calorie estimation system based on semantic segmentation network, *Sensors and Materials, 35*(6), 2023, 2013–2033.

[19] T. Lattisi, D. Farina, and M. Ronchetti, Semantic segmentation of text using deep learning, *Computing and Informatics, 41*(1), 2022, 78–97.

[20] S.H. Lee, D.W. Lee, and M.S. Kim, A deep learning-based semantic segmentation model using MCNN and attention layer for human activity recognition, *Sensors, 23*(4), 2023, 19.
[21] C.M. Lin, C.Y. Tsai, Y.C. Lai, S.A. Li, and C.C. Wong, Visual object recognition and pose estimation based on a deep semantic segmentation network, *IEEE Sensors Journal, 18*(22), 2018, 9370–9381.
[22] F. Lin, Z.T. Yu, Q.N. Jin, and A.J. You, Semantic segmentation and scale recognition-based water-level monitoring algorithm, *Journal of Coastal Research, 105*, 2020, 185–189.
[23] C.C. Liu, Y.C. Zhang, P.Y. Chen, C.C. Lai, Y.H. Chen, J.H. Cheng, and M.H. Ko, Clouds classification from sentinel-2 imagery with deep residual learning and semantic image segmentation, *Remote Sensing, 11*(2), 2019, 16.
[24] A. López-Cifuentes, M. Escudero-Viñolo, J. Bescós, and A. García-Martín, Semantic-aware scene recognition, *Pattern Recognition, 102*, 2020, 15.
[25] Y. Lyu, G. Vosselman, G.S. Xia, A. Yilmaz, and M.Y. Yang, UAVid: A semantic segmentation dataset for UAV imagery, *ISPRS Journal of Photogrammetry and Remote Sensing, 165*, 2020, 108–119.
[26] F. Magistri, J. Weyler, D. Gogoll, P. Lottes, J. Behley, N. Petrinic, and C. Stachniss, From one field to another-unsupervised domain adaptation for semantic segmentation in agricultural robotics, *Computers and Electronics in Agriculture, 212*, 2023, 10.
[27] M. Markovic, R. Malehmir, and A. Malehmir, Diffraction pattern recognition using deep semantic segmentation, *Near Surface Geophysics, 20*(5), 2022, 507–518.
[28] S. Matsuzaki, J. Miura, and H. Masuzawa, Multi-source pseudo-label learning of semantic segmentation for the scene recognition of agricultural mobile robots, *Advanced Robotics, 36*(19), 2022, 1011–1029.
[29] Y.J. Mo, Y. Wu, X.N. Yang, F.L. Liu, and Y.J. Liao, Review the state-of-the-art technologies of semantic segmentation based on deep learning, *Neurocomputing, 493*, 2022, 626–646.
[30] F. Poux and R. Billen, Voxel-based 3D point cloud semantic segmentation: Unsupervised geometric and relationship featuring vs. deep learning methods, *ISPRS International Journal of Geo-Information, 8*(5), 2019, 34.
[31] J.K. Pu and W. Zhang, Electric vehicle fire trace recognition based on multi-task semantic segmentation, *Electronics, 11*(11), 2022, 16.
[32] R.D. Pu, G.Q. Ren, H.J. Li, W. Jiang, J.S. Zhang, and H.L. Qin, Autonomous concrete crack semantic segmentation using deep fully convolutional encoder-decoder network in concrete structures inspection, *Buildings, 12*(11), 2022, 20.
[33] M.A. Razzaq, I. Cleland, C. Nugent, and S. Lee, Semimput: Bridging semantic imputation with deep learning for complex human activity recognition, *Sensors, 20*(10), 2020, 23.
[34] C. Redondo-Cabrera, M. Baptista-Ríos, and R.J. López-Sastre, Learning to exploit the prior network knowledge for weakly supervised semantic segmentation, *IEEE Transactions on Image Processing, 28*(7), 2019, 3649–3661.
[35] D. Ryu, K., Kitaguchi, K. Nakajima, Y. Ishikawa, Y. Harai, A. Yamada, Y. Lee, K. Hayashi, N. Kosugi, H. Hasegawa, N. Takeshita, Y. Kinugasa, and M. Ito, Deep learning-based vessel automatic recognition for laparoscopic right hemicolectomy, *Surgical Endoscopy and Other Interventional Techniques, 38*(1), 2023, 171–178.
[36] M.U. Saeed, N. Dikaios, A. Dastgir, G. Ali, M. Hamid, and F. Hajjej, An automated deep learning approach for spine segmentation and vertebrae recognition using computed tomography images, *Diagnostics, 13*(16), 2023, 17.
[37] H.K. Zhang, A new hybrid whale particle swarm optimisation algorithm for robot trajectory planning and tracking control, *Mechatronic Systems and Control*, *52*(1), 2024, 48–57.

Biographies

Shengnan Gao was born in 1986 in Qinhuangdao, Hebei Province, China. She received the bachelor's degree in information management and information systems from Zhengzhou University in 2009, and the master's degree in computer software and theory and the Ph.D. degree in computer application technology from Yanshan University in 2012 and 2017, respectively. Since 2021, she has been a Faculty Member with the Beijing Agricultural University. Her research interests include data analysis, artificial intelligence, and natural language processing.

Xiaoshun Li was born in 1984 in Qinhuangdao, Hebei Province, China. He received the bachelor's degree in computer science and technology and the master's degree in computer application technology from the University of Science and Technology, Beijing, in 2006 and 2009, respectively. Since 2018, he has been a Senior Experimenter with the School of Intelligent Science and Engineering, Beijing Agricultural University. His research interests include the application of artificial intelligence in agriculture, plant phenotypic analysis, intelligent diagnosis of plant diseases, and plant growth visualisation.

Yingying Liu was born in 1980 in China. She received the bachelor's degree in computer science and technology from Northeast Normal University in 2002, and the master's degree in computer application technology from North China Electric Power University in 2009. Since 2014, she has been an Associate Professor with the School of Intelligent Science and Engineering, Beijing Agricultural University. Her research interests include smart agriculture, plant growth visualisation, intelligent diagnosis of plant diseases, and knowledge expression and reasoning.