RESEARCH ON WEED RECOGNITION AND CROP ROW EXTRACTION TECHNOLOGY BASED ON DEEP LEARNING
/
基于深度学习的杂草识别与作物行提取技术研究

Tengxiang YANG; Chengqian JIN*); Youliang Ni; Man CHEN; Zhen LIU [1]
Nanjing Institute of Agricultural Mechanization, Ministry of Agriculture and Rural Affairs, Nanjing, Jiangsu / China
Tel: +8602584346113; E-mail: jinchengqian@126.com;
DOI: https://doi.org/10.35633/inmateh-76-57

Keywords: *semantic segmentation; image recognition; corn crop row recognition; precision agriculture;*

ABSTRACT

To address the limitation of existing agricultural unmanned plant protection equipment in perceiving crop growth status in real time during the maize seedling stage, this study proposes a crop row extraction method based on image processing. A crop semantic segmentation network was developed using the UNet framework, with VGG19 as the encoder and transposed convolution as the decoder. Model testing demonstrated that the segmentation network achieved accuracy rates of 0.9865 on the training set and 0.9864 on the validation set, with corresponding loss values of 0.0254 and 0.0270. In continuous processing scenarios, the average time for semantic segmentation per image was 120 milliseconds, while crop row extraction required 23 milliseconds.

摘要

为解决现有农用无人植保机具在玉米苗期作业时无法现场感知作物生长状况的问题，提出一种基于图像处理的玉米苗期作物行提取方法。基于 Unet 框架构建了作物语义分割网络，编码器为 VGG19，解码器为转置卷积，通过添加注意力机制来增强主要特征信息的权重。采用传统图像处理技术获取作物的轮廓、最小外接圆的圆心与半径等关键信息，引入数据缓存机制，增加用于作物行提取的作物数据量。运用苗带自动聚类算法，在作物行数量与位置不确定的情况下，自动对作物行的数量及其所包含的作物对象进行聚类。模型测试结果表明：作物语义分割模型在验证集与测试集上的准确率分别达到 0.9865 和 0.9864，损失率分别为 0.0254 和 0.0270。在实际连续处理过程中，单张图片的语义分割时间平均为 120 毫秒，作物行提取时间为 23 毫秒。

INTRODUCTION

Currently, unmanned field management machinery primarily relies on satellite navigation and inertial navigation systems for positioning during field operations. In practice, the machinery follows a predetermined absolute spatial trajectory without accounting for the relative spatial relationship between the machinery and the crops *(Wang et al., 2022; Xie et al., 2023)*. However, in actual agricultural production, field operations fundamentally involve direct interaction between the machinery and the crops. Therefore, it is essential to investigate crop row extraction technology to enhance the crop perception capabilities of unmanned agricultural machinery, thereby improving operational accuracy and efficiency while minimizing crop damage during autonomous operations.

Crop row extraction is a fundamental requirement for enabling unmanned plant protection operations in agricultural machinery. Existing studies generally adopt image processing methods for crop row extraction, typically comprising two main steps: crop recognition and crop row fitting. Traditional crop recognition approaches include the super-green method for crop segmentation *(Jiang et al., 2021)*, wavelet transform processing *(Habib et al., 2021)*, support vector machine classification *(Wu et al., 2023)*, and classification based on texture and color features *(Zhang et al., 2023)*. In recent years, with the rapid advancement of deep learning technologies, several classical neural network architectures have been applied to crop and weed classification tasks, such as Artificial Neural Networks (ANN) *(Monteiro et al., 2021)*, Convolutional Neural Networks (CNN) *(Jabir et al., 2021; Yang et al., 2023)*, Sum-Product Networks (SPN) *(Kumar Kempegowda et al., 2022; Wang S. et al., 2019)*, Transformers *(Feng et al., 2024; K. Jiang et al., 2022)*, and Probabilistic Neural Networks (PNN) *(Venkataraju et al., 2023)*.

[1] *Tengxiang Yang, Stud. Eng; Chengqian Jin, Prof. Ph.D. Eng.; Youliang Ni, Associate Prof. Ph.D. Eng.; Man Chen, Associate Prof. Ph.D. Eng.; Zhen Liu, Associate Prof. Ph.D. Eng.;*

Methods for fitting crop rows include the least squares method *(Diao et al., 2024)*, Hough transform *(Chen et al., 2019; Li et al., 2022)*, skeleton extraction *(Wu et al., 2024; Zhao et al., 2019)*, feature engineering *(Zhang et al., 2023)*, partition clustering *(Ma et al., 2021)*, Fast-SCNN *(Chen et al., 2021)*, and Mean-Shift *(Weibing, 2019)*.

However, existing methods face several limitations. When the angle between the camera and the ground is large or under backlit conditions, direct sunlight entering the image degrades image quality, making processing more difficult. In practical operations, weeds often cross or overlap with crops. Traditional methods relying on shallow features such as color, texture, and shape struggle to achieve accurate, robust, and generalizable segmentation. The crop information extracted from a single image is limited. The crop position data derived from it is susceptible to random errors, leading to significant fluctuations in measurement results.

To address these challenges, this study positioned the camera beneath the vehicle body with its central axis perpendicular to the ground to stabilize image acquisition and reduce interference from direct sunlight and reflections. For crop and weed segmentation, a deep learning-based semantic segmentation network was employed for initial crop recognition, followed by traditional image processing techniques to extract crop contour and positional information. During crop row extraction, contour information from the most recent five frames was collectively analyzed. An automatic seedling row clustering algorithm was then used to identify the seedling rows within the images, from which the crop row information was subsequently extracted.

MATERIALS AND METHODS

Data Acquisition and Processing

The dataset used in this study was collected by an onboard camera mounted at the bottom of a high-clearance boom sprayer, with a mounting height of 80 cm above ground level. The data acquisition was conducted on August 27, 2020, at the Baima Test Base of the Nanjing Research Institute for Agricultural Mechanization, Ministry of Agriculture and Rural Affairs. The vehicle-mounted camera employed was a RealWide USBFHD01M camera (maximum resolution: 1920×1080 pixels). A total of three video clips were recorded during the experiment. The first two video clips were used for model training, from which 1,436 images were obtained through frame extraction and cropping. The last video clip was reserved exclusively for testing purposes.

The crop objects in the images were annotated using the polygon tool in the image annotation software Labelme, where the crops were labeled as 1 and the rest as 0. The proposed method requires only crop image annotations, eliminating the need for weed classification labeling. The annotated data was divided into a training set and a validation set in an 8:2 ratio. The dataset was subjected to image data augmentation in the following ways with a 50% probability in sequence: random contrast adjustment, random brightness adjustment, random left-right flipping, and random noise addition. To facilitate training, the original image size obtained by the camera was adjusted to 512×512.

Semantic Segmentation Model Construction

The crop segmentation network was designed based on the classic semantic segmentation model UNet. The UNet model consisted of two main components: an encoder and a decoder. For the encoder, DenseNet121 served as the backbone network, extracting four top-level convolution maps with dimensions of 256×256, 128×128, 64×64, and 32×32 as feature maps. The decoder comprised two key elements: the Transition Layer and the UnSampling Layer. The Transition Layer followed a specific structure: CABM (Convolutional Block Attention Module) followed by 1×1 C2DT (Conv2Dtranspose), BN (Batch Normalization), and ReLU (Rectified Linear Unit). Meanwhile, the UnSampling Layer was structured as 2×2 C2DT + BN + ReLU. The overall network architecture is illustrated in Fig. 1.

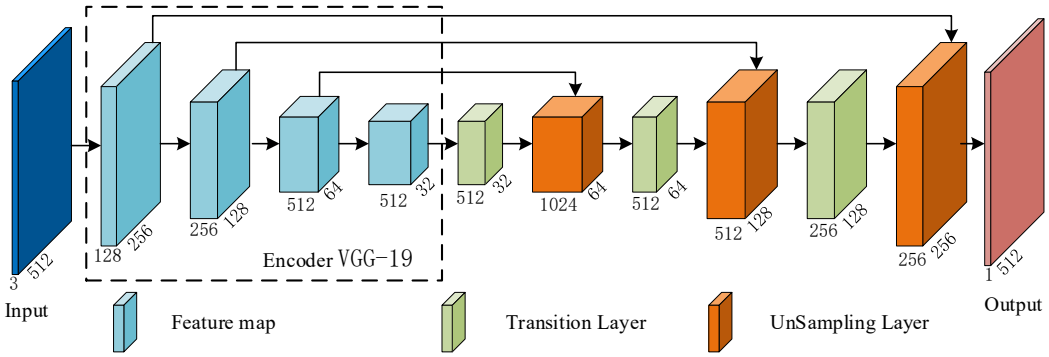


Fig. 1 - Network model of crop segmentation

The encoder employed VGG19 as its backbone network. VGG, an improved architecture derived from AlexNet, was developed by the Visual Geometry Group at the University of Oxford. This network came in two primary variants: VGG16, with 16 hidden layers (13 convolutional + 3 fully connected layers), and VGG19, featuring 19 hidden layers (16 convolutional + 3 fully connected layers). A key innovation of the VGG architecture was the use of smaller convolutional kernels, replacing a single 7×7 kernel with three consecutive 3×3 kernels and a 5×5 kernel with two 3×3 kernels. This design deepened the network while maintaining the same receptive field, enhancing feature extraction efficiency. In this study, four convolutional layers from VGG19—block5_conv1, block4_conv1, block3_conv1, and block2_conv1—were extracted as feature maps for crop segmentation. These layers had dimensions of 32×32×512, 64×64×512, 128×128×256, and 256×256×128, respectively.

The Transition Layer served as the preprocessing part of the decoder and comprised two components: the CABM attention module and the transposed convolution module. CABM represented the convolutional block attention mechanism, which functioned as a hybrid attention mechanism combining spatial attention and channel attention. The spatial attention mechanism consisted of three parts. CBAM demonstrated its advantage through dynamically focusing on key image regions by adaptively assigning weights to different positions, which significantly enhanced model performance. It automatically identified and enhanced task-relevant areas including target objects, edges, or textures while suppressing irrelevant backgrounds or noise, thereby improving the model's robustness to occlusion and complex environments. The first part involved channel pooling, where a tensor of size (h×w×c) was processed through maximum pooling and average pooling to yield a tensor of size (h×w×2). The second part performed a convolution operation on this tensor to produce an output tensor of size (h×w×1). The third part normalized this tensor through Batch Normalization and activated it with a Sigmoid function to obtain a spatial weight tensor, which was then multiplied by the input tensor to adjust the data weights in the spatial dimension. The channel attention mechanism was divided into four parts. The first part executed global maximum pooling and global average pooling on each channel, compressing each channel's data into one pixel to ultimately generate two tensors of size (1×1×c). The second part processed these tensors through two fully connected layers respectively to produce connection weights between channels, thereby enhancing channel connectivity. The third part stacked the two tensors from the previous step, activated them with a Sigmoid function, and multiplied them by the input tensor to modify each channel's weight levels. The transposed convolution employed a 3×3 kernel size with a stride of 1, maintaining the feature map's original size after processing.

The UnSampling Layer functioned as the upsampling component of the decoder and consisted of two parts: the transposed convolution module and feature fusion. The transposed convolution module employed a stride of 2, which doubled the image size after processing. Since transposition had already been performed in the preprocessing stage, the positional information within the feature map remained preserved during this operation. The resulting upsampled feature map subsequently underwent batch normalization and ReLU activation. Feature fusion represented a fundamental design characteristic of the UNet network architecture. This process concatenated the upsampled feature map produced by transposed convolution with the corresponding feature map of identical dimensions extracted from VGG19, thereby combining high-level abstract features from low-resolution data with low-level surface features from high-resolution data within the final feature map.

Loss Function Construction

The loss function L employed for model training in this study was computed through a weighted combination of Dice Loss and Binary Cross Entropy Loss. The calculation followed this formula:

$$L = w \times L_{\text{bce}} + L_{\text{dice}} \quad (1)$$

where:

The symbol w represents the weight, L_{bce} represents the Binary Cross Entropy loss, and L_{dice} represents the Dice Loss.

$$L_{bce}(\alpha,\beta) = -\frac{1}{N}\sum_{i=1}^{N}(\alpha \ln\left(\frac{1}{1+e^{-\beta}}\right) + (1-\beta)\ln\left(1-\frac{1}{1+e^{-\beta}}\right)) \quad (2)$$

where:

The symbol α is the actual label (0 or 1), and β is the probability predicted by the model.

$$L_{dice} = 1 - \frac{2|X\cap Y|+0.01}{|X|+|Y|+0.01} \quad (3)$$

where:

$|X \cap Y|$ represents the number of pixels in the intersection of the prediction result and the ground truth label. $|X|$ and $|Y|$ respectively denote the number of pixels in the prediction result and the ground truth label.

The model evaluation metrics are the average recognition *accuracy* and *mIOU*, and their calculation formulas are as follows:

$$Accuracy = \frac{1}{c}\sum_{j=1}^{c}\frac{n_{jj}}{n_j} \times 100\% \quad (4)$$

where: c denotes the number of categories, n_j represents the total number of samples in the j-th category, and n_{jj} indicates the number of correctly predicted samples for the j-th category.

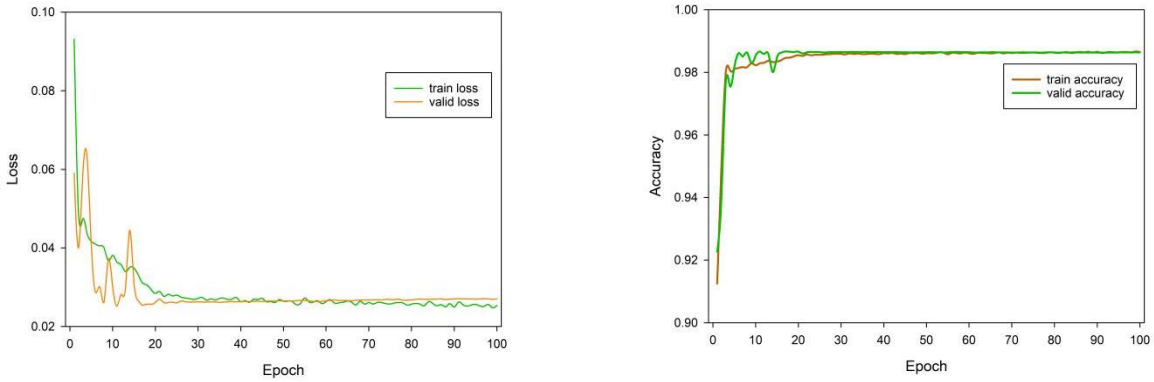
$$mIOU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \quad (5)$$

where: p_{ii} represents the number of cases where the true value is i and it is predicted as i, which is the number of true positives; p_{ij} represents the number of cases where the true value is i but it is predicted as j, which is the number of false positives; p_{ji} represents the number of cases where the true value is j but it is predicted as i, which is the number of false negatives; k+1 represents the number of categories including the empty class. Since this study only distinguishes between crops and background, k =1.

Model Training

The model training platform was configured as follows: The CPU was an Intel Core i7 10700 with a main frequency of 4.8 GHz and 16 GB of memory, while the GPU was an Nvidia RTX 2060 with 11 GB of video memory. The operating environment consisted of Ubuntu 20.04 operating system, Python 3.9.0 programming language, and Tensorflow 2.5-GPU deep learning framework. The Adam optimization algorithm was employed for model training. The batch size was set to 8, with 150 training epochs and an initial learning rate of 0.01. A learning rate reduction mechanism was implemented where, if the loss remained unchanged for 5 consecutive epochs, the learning rate would decrease to 0.1 times its original value.

The training process completed 100 epochs, with the results presented in Fig. 2. Figure 2a demonstrated that the model started converging around the 25th epoch, ultimately achieving stabilized accuracies of 0.9865 for the training set and 0.9864 for the validation set. As shown in Fig. 2b, the loss rates similarly began converging around the 25th epoch, reaching values of 0.0254 (training) and 0.0270 (validation). After 70 epochs, the training loss became consistently lower than the validation loss, indicating the onset of model overfitting beyond this point.



a) - Training loss **b) - Training accuracy**

Fig. 2 - Process of model training

Weed Recognition

As illustrated in Fig. 3, after being processed by the crop-weed semantic segmentation network, a crop probability map was generated with a size of 512×512×1. The pixel values ranged from 0 to 1, representing the likelihood of each pixel belonging to a crop, where a value of 0 indicated background and a value of 1 indicated crop. To facilitate subsequent processing, the predicted map was binarized by considering pixels with probability values greater than 0.8 as crops, and those with values less than or equal to 0.8 as background.

The binarized predicted map exhibited spatial artifacts including isolated regions and fragmentation. To address these issues, a morphological opening operation was first applied using a 5×5 convolution kernel, which effectively removed isolated points, burrs, and bridging artifacts. Subsequently, a morphological closing operation with an identical 5×5 kernel size was performed to eliminate small holes, cracks, and surface pits in the processed image. This two-step morphological processing significantly improved the spatial coherence of the segmentation results while preserving the essential structural features of the crop distribution pattern.

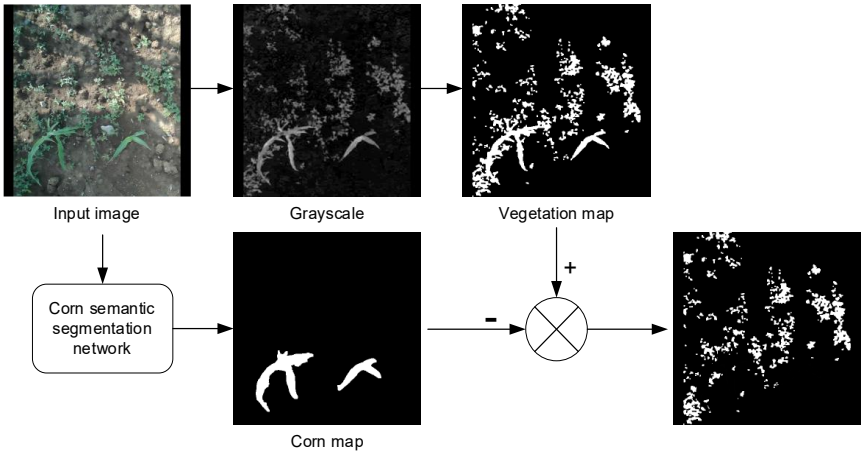


Fig. 3 - Process of weed extraction

Vegetation layer extraction was performed to obtain green vegetation information from images through specific processing methods. The image processing procedure consisted of the following steps. First, the red, green, and blue channels of the image were decomposed. The green channel was then enhanced using the super-green algorithm to produce a corresponding grayscale image, with pixel values constrained to the range of 0 to 255. Subsequently, the Otsu method (maximum inter-class variance method) was applied to perform binarization on the grayscale image, resulting in a binary image. Finally, a morphological opening operation was conducted using a 3×3 convolution kernel to generate the final vegetation layer. The super-green algorithm for green channel enhancement was calculated using the following formula:

$$G_{\mathrm{r}} = \min\left(255, \max\left(2G - B - R, 0\right)\right) \tag{5}$$

where: G_{r} represents the output grayscale image, while G, B, and R denote the green, blue, and red channels of the input image, respectively.

Crop row extraction

Contour detection was implemented using the findContours function from the OpenCV library. The detection mode was configured as RETR_EXTERNAL, ensuring only external contours were identified. For contour approximation, the CHAIN_APPROX_SIMPLE method was employed, which effectively compressed horizontal, vertical, and diagonal segments while preserving solely their endpoint coordinates.

The minimum enclosing circle corresponding to each contour was determined using the minEnclosingCircle function from the OpenCV library. This function automatically iterated through the 2D point set of each contour and calculated the precise center coordinates and radius of its minimum enclosing circle. The complete contour detection process, including this circle-fitting operation, was visually demonstrated in Fig. 4.

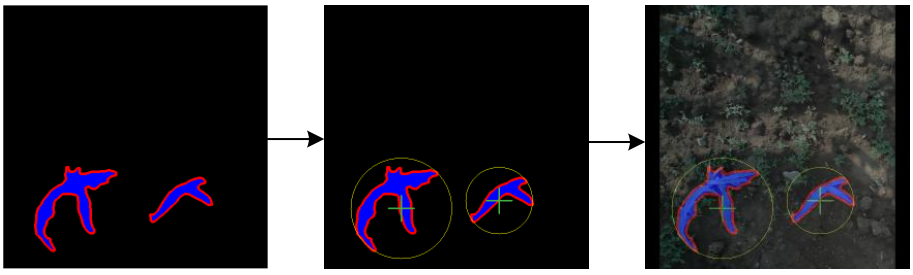


Fig. 4 - Process of crop positioning

The crop row positioning process was illustrated in Fig. 5. First, the semantic segmentation network extracted crop information from the input image, and the centers and radii of minimum enclosing circles for all crops were calculated and stored in memory. The system then verified whether the number of stored contour frames exceeded five; if so, the oldest frame was removed until the frame count satisfied the threshold requirement. Finally, all recorded crop contour data underwent clustering analysis. Upon identifying multiple seedling rows, the corresponding centerlines and boundary lines for each row were subsequently extracted.

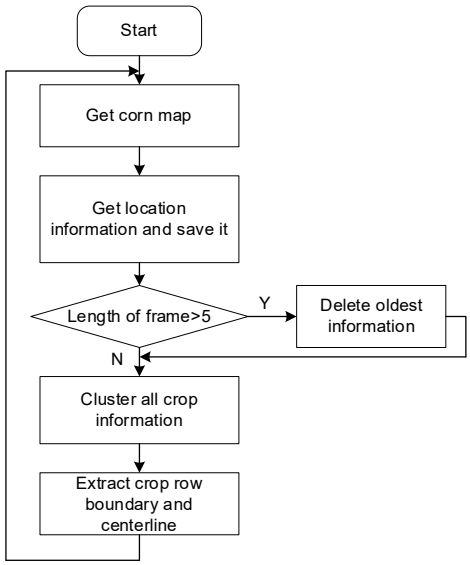


Fig. 5 - Process of crop row positioning

Due to the potential presence of multiple seedling rows within a single image, where both the quantity and spatial arrangement of these rows were unpredictable, the minimum-maximum distance algorithm was employed to perform clustering analysis on the contour coordinates. This clustering process enabled the determination of the exact number of seedling rows present in each image. As demonstrated in Fig. 6, the seedling row clustering algorithm operated according to the following procedure:

Algorithm 1: Crop row clustering

Data: set of point coordinates P, threshold of classification distance η, Minimum classification distance σ
Result: set of point classification Y

$center \leftarrow 0$;
$class \leftarrow 1$;
$Y \leftarrow \{1, 1, ..., 1\}$;
$maxDistance \leftarrow 0$;
calculate the points distance of P_1;
for $i \leftarrow 1$ **to** *Number of points* **do**
  $Di \leftarrow abs(P_{yi} - P_{y0})$;
  if $Di > max(D)$ **then**
    $maxDistance \leftarrow Di$;
    $center \leftarrow i$;
  end
end
while $max(D) > (maxDistance \times \eta)$ *and* $max(D) > \sigma$ **do**
  $class \leftarrow class + 1$;
  for $i \leftarrow 1$ **to** *Number of points* **do**
    $d \leftarrow abs(P_{yi} - P_{ycenter})$;
    if $Di > d$ **then**
      $Di \leftarrow d$;
      $Yi \leftarrow class$;
    end
  end
  $center \leftarrow$ *the index of max value in* D ;
end

Fig. 6 - Process of automatic seedling belt clustering

For a given input set of coordinate points $P=\{p_1, p_2, p_3, ..., p_n\}$, the algorithm selected p_1 as the initial clustering center. The distances from all points to p_1 were computed and stored in set D. The maximum value in D was identified as $maxDistance$, with its corresponding coordinate point designated as the next clustering center. Subsequently, the distance from each point to this new center was calculated and compared against the stored values in D. When a smaller distance was found, the point was assigned to the new cluster and its corresponding value in D was updated.

Following complete distance evaluation, the algorithm verified whether the maximum value in D satisfied the threshold condition: exceeding the minimum row spacing threshold *σ* while remaining below *η*×*maxDistance*, where *η* (a distance coefficient between 0 and 1) scaled the maximum classification threshold. If valid, the point associated with the current maximum distance became the subsequent clustering center, and the process iterated. Termination occurred when no values in *D* met the threshold criteria. The final output comprised classification set *Y* for point set *P*, representing the clustered results. The complete clustering procedure operated as follows:

Following point clustering, the classification set *Y* was obtained for the input set *P*, with values ranging from 1 to n (where n represented the maximum number of classifications, corresponding to the number of seedling rows). Through iterative traversal of set *Y*, the algorithm extracted the point set associated with each classification. Within each classified point set, the uppermost boundary (*LineA*) and lowermost boundary (*LineB*) of the contour were identified. The region bounded by these two lines constituted the seedling row area, while the midpoint between the upper and lower boundaries established the crop seedling row's centerline. This complete calculation process was illustrated in Fig. 7 and executed as follows:

Algorithm 1: Crop row extraction

Data: set of point coordinates P, set of point classification Y, set of circumcircle radius R
Result: Borderline $LineA, LineB$

calculate the number of crop rows;
for $i \leftarrow 1$ **to** *Number of categories* **do**
  for $j \leftarrow 1$ **to** *Number of points* **do**
    $LineA_i \leftarrow 0$;
    $LineB_i \leftarrow 512$;
    if $Y_j == i$ **then**
      $LineA_i \leftarrow max(LineA_i, P_{yj} + R_j)$;
      $LineB_i \leftarrow min(LineB_i, P_{yj} - R_j)$;
    end
  end
  $Line_i \leftarrow average(LineA_i, LineB_i)$;
end

Fig. 7 - Process of seedling belt boundary extraction

RESULTS

To validate the effectiveness and feasibility of the proposed algorithm, corresponding models and algorithms were constructed and verified using the third video clip collected during the experiment. The video recording specifications included a frame rate of 30 frames per second and an image resolution of 1920×1080 pixels. To accurately simulate the working environment, images were sequentially extracted from the video at 5-frame intervals. Each extracted image was then processed by cropping a central 1080×1080 area, which was subsequently resized to 512×512 pixels before being input into the algorithm for processing.

As shown in Fig. 8, the processing results of the first five extracted images during the experiment were presented. Horizontally, the images were arranged in chronological order, representing the first to the fifth image, with an interval of five frames between each. Vertically, the images were organized according to the processing steps: crop semantic segmentation, contour detection and localization, and navigation line extraction. Statistical analysis showed that the average processing time for crop semantic segmentation was 120 ms, while crop row extraction took an average of 23 ms.

From the crop semantic segmentation images, it was observed that corn was identified and segmented in all five images. The probability density was higher at the center of the leaves and lower at the edges, indicating better recognition performance in the core area and poorer performance at the leaf tips. Corn plants with two shoots per hole were recognized as a single plant, since the semantic segmentation network only classified objects by category without distinguishing individual instances. However, for optimal growth and increased yield per mu, single-plant-per-hole configurations were more desirable. Therefore, distinguishing individual seedlings was necessary to enable mechanized seedling removal operations.

In the contour segmentation images, the corn contours were accurately identified based on the segmented corn layer, and the center and radius of the minimum enclosing circles were calculated. Morphological operations and threshold filtering were applied to smooth the contours. When crops entered or exited the image frame, only partial contour information was captured, which limited the accuracy of the extracted contours. Thus, further refinement was required for contour optimization at the image boundaries to better reflect actual crop conditions.

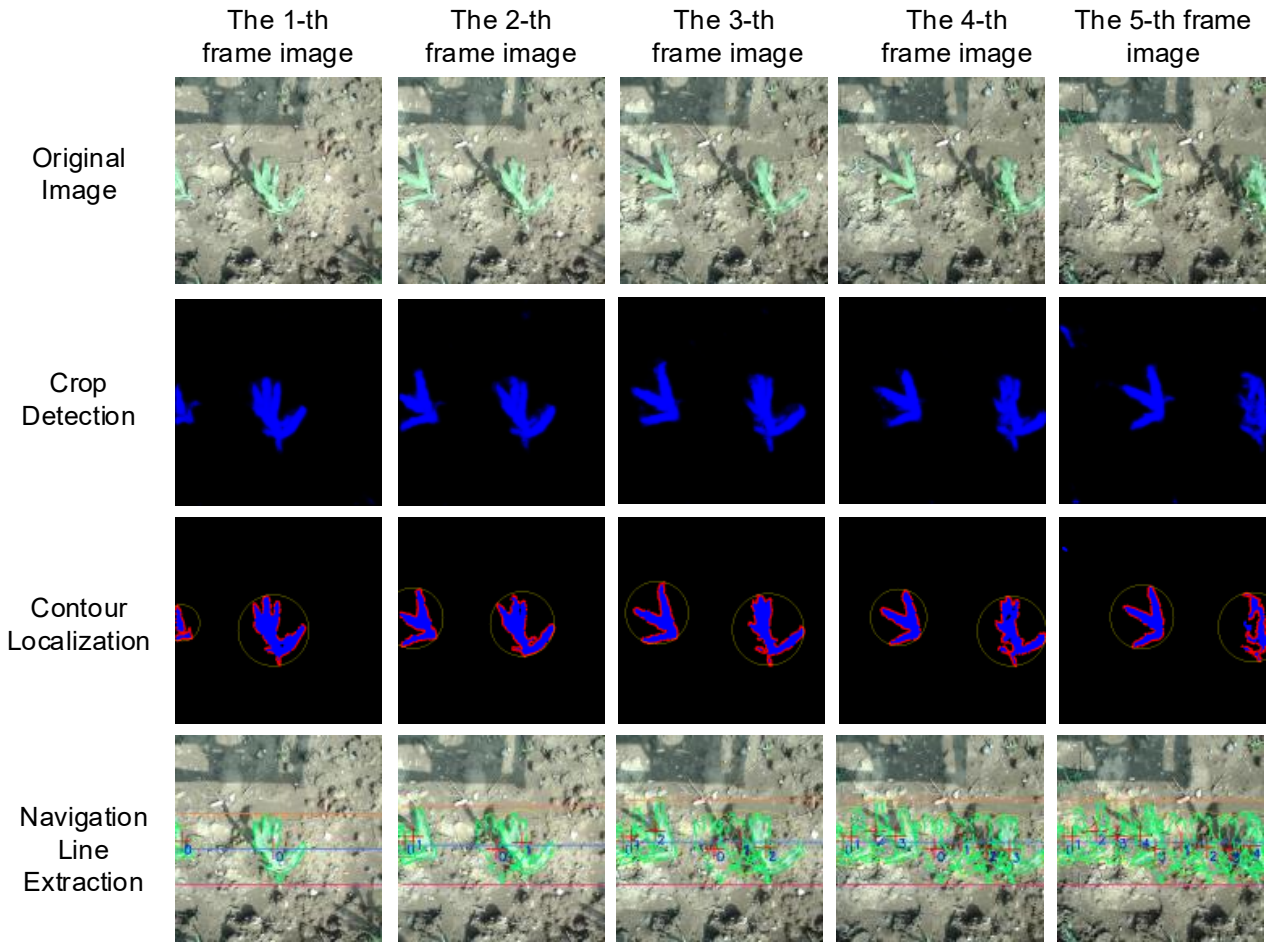


Fig. 8 - Algorithm testing results visualization

In the navigation line extraction images, the green dashed lines represented the crop contours, red crosses indicated the centers of the enclosing circles, yellow circles denoted the minimum enclosing circles, blue numbers marked the sequence of contour acquisition (i.e., the temporal order of contour generation), orange lines showed the upper boundaries of the seedling rows, dark red lines the lower boundaries, and blue lines the center lines. It was evident that seedling row information was extracted and the movement trajectories of the crops were tracked. Due to the asymmetrical shape of corn leaves, a deviation was observed between the enclosing circle center and the actual plant center. Additionally, differences in growth conditions led to vertical positional discrepancies even among adjacent plants. By comparing seedling row position data across the five images, the crop boundaries were successfully identified, reflecting the spatial distribution of the crops.

CONCLUSIONS

To address the challenges of crop recognition and segmentation in weedy field environments, this study proposed a crop recognition model based on a semantic segmentation network, enabling pixel-level differentiation between crops and weeds. From the segmented pure crop layer, the model extracted crop contour features and positional information to determine the location of the seedling rows in which the crops were situated.

A crop semantic segmentation network was constructed using semantic segmentation techniques. Model evaluation showed that the network achieved accuracy rates of 0.9865 on the training set and 0.9864 on the validation set, with corresponding loss values of 0.0254 and 0.0270. In real-time sequential processing, the average time required for semantic segmentation per image was 120 milliseconds, while crop row extraction took an average of 23 milliseconds.

By employing a multi-frame data caching strategy, the number of crop objects available for row extraction was increased in scenarios with sparse crop distribution, thereby enhancing the accuracy of crop row recognition.

ACKNOWLEDGEMENT

We greatly appreciate the careful and precise reviews by the anonymous reviewers and editors. This research was funded by the National Key Research and Development Plan project (2023YFD230050403) and the National Natural Science Foundation of China (32171911,32171911), and CAAS Center for Science in Smart Agriculture and Equipment (CAAS-SAE-202301).

REFERENCES

[1] Chen, Y., Geng, C., Wang, Y., Zhu, G., & Shen, R. (2021). Extraction method for centerlines of rice seedlings based on Fast-SCNN semantic segmentation. *INMATEH Agricultural Engineering*, Vol. 64, pp.335–344, Jiangsu/China.

[2] Chen Z., Li W., Zhang W., Li Y., Li M., & Li H. (2019). Research on Vegetable Crop Row Extraction Method Based on Automatic Hough Transform Accumulation Threshold (基于自动 Hough 变换累加阈值的蔬菜作物行提取方法研究). *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 35, pp.314–322, Beijing/China.

[3] Diao, Z., Ma, S., Zhang, D., Zhang, J., Guo, P., He, Z., Zhao, S., & Zhang, B. (2024). Algorithm for Corn Crop Row Recognition during Different Growth Stages Based on ST-YOLOv8s Network. *Agronomy*, Vol.14, pp. 1466, Henan/China.

[4] Feng, X., Chen, D., & Yongke, L. (2024). Research on Cotton Seedling and Weed Detection Model Based on Improved Detection Transformer (基于改进 Detection Transformer 的棉花幼苗与杂草检测模型研究). *Computer & Digital Engineering*, Vol. 52, pp. 2176–2182, Xinjiang/China.

[5] Habib, S., Khan, I., Islam, M., Albattah, W., Mohammed Alyahya, S., Khan, S., & Kamrul Hassan, M. (2021). Wavelet Frequency Transformation for Specific Weeds Recognition. *2021 1st International Conference on Artificial Intelligence and Data Analytics* (CAIDA), Vol. 9, pp. 97–100, Saudi Arabia.

[6] Jabir, B., Rabhi, L., & Falih, N. (2021). RNN- and CNN-based weed detection for crop improvement: An overview. *Foods and Raw Materials*, Vol. 9, pp. 387–396, Morocco.

[7] Jiang, J., & Gao, M. (2021). Retraction Note: Agricultural super green image segmentation method based on Gaussian mixture model combined with Camshift. *Arabian Journal of Geosciences*, Vol. 14, pp. 2697, Zhejiang/China.

[8] Jiang, K., Afzaal, U., & Lee, J. (2022). Transformer-Based Weed Segmentation for Grass Management. *Sensors*, Vol. 23, pp. 65, Republic of Korea.

[9] Kumar Kempegowda, S., Rajeswari, R., Satyanarayana, L., & Matada Basavarajaiah, S. (2022). Hybrid features and ensembles of convolution neural networks for weed detection. *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 12, pp. 6756, India.

[10] Li, X., Lloyd, R., Ward, S., Cox, J., Coutts, S., & Fox, C. (2022). Robotic crop row tracking around weeds using cereal-specific features. *Computers and Electronics in Agriculture*, Vol. 197, pp. 106941, United Kingdom.

[11] Ma, Z., Tao, Z., Du, X., Yu, Y., & Wu, C. (2021). Automatic detection of crop root rows in paddy fields based on straight-line clustering algorithm and supervised learning method. *Biosystems Engineering*, Vol. 211, pp. 63–76, Zhejiang/China.

[12] Monteiro, A. L., de Freitas Souza, M., Lins, H. A., da Silva Teófilo, T. M., Júnior, A. P. B., Silva, D. V., & Mendonça, V. (2021). A new alternative to determine weed control in agricultural systems based on artificial neural networks (ANNs). *Field Crops Research*, Vol. 263, pp. 108075, Brazil.

[13] Venkataraju, A., Arumugam, D., Stepan, C., Kiran, R., & Peters, T. (2023). A review of machine learning techniques for identifying weeds in corn. *Smart Agricultural Technology*, Vol. 3, pp. 100102, United States.

[14] Wang S., Wang S., Zhang H., & Wen C. (2019). Weed Identification in Soybean Fields Based on Lightweight Sum-Product Network and UAV Remote Sensing Images (基于轻量和积网络及无人机遥感图像的大豆田杂草识别). *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 35, pp. 81–89, Jiling/China.

[15] Wang, T., Chen, B., Zhang, Z., Li, H., & Zhang, M. (2022). Applications of machine vision in agricultural robot navigation: A review. *Computers and Electronics in Agriculture*, Vol. 198, pp. 107085, Beijing/China.

[16] Weibing, W. (2019). Target Detection and Analysis of Intelligent Agricultural Vehicle Movement Obstacle Based on Panoramic Vision. *INMATEH Agricultural Engineering*, Vol.59, pp. 277–284, Anhui/China.

[17] Wu, B. (2024). Virtual Reality-Integrated Solutions for Extracting Central Crop Row Skeleton Lines Using the Maximum Diamond Principle Algorithm. *Computer-Aided Design and Applications*, Vol. 17, pp. 37–48, Henan/China.

[18] Wu, Y., He, Y., & Wang, Y. (2023). Multi-Class Weed Recognition Using Hybrid CNN-SVM Classifier. *Sensors*, Vol. 23, pp. 7153, Tianjin/China.

[19] Xie, B., Jin, Y., Faheem, M., Gao, W., Liu, J., Jiang, H., Cai, L., & Li, Y. (2023). Research progress of autonomous navigation technology for multi-agricultural scenes. Computers and Electronics in *Agriculture*, Vol. 211, pp. 107963, Zhenjiang/China.

[20] Yang S., Zhang H., & Xing L. (2023). Improved MobileViT Network for Lightweight Field Weed Identification (改进 MobileViT 网络识别轻量化田间杂草). *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 39, pp. 152–160, Heilongjiang/China.

[21] Zhang, L., Li, K., & Qi, Y. (2023). Recognition of crop leaf diseases based on multi-feature fusion and evolutionary algorithm optimisation. *International Journal of Bio-Inspired Computation*, Vol. 21, pp. 163–173. , Guangzhou/China.

[22] Zhang, S., Liu, Y., Xiong, K., Zhai, Z., & Du, Y. (2023). Field Crop Row Centerline Recognition Method Based on Feature Engineering (基于特征工程的大田作物行中心线识别方法). *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 54, pp. 18–26, Beijing/China.

[23] Zhao, X., Ma, W., Gao, Y., & Zang, Y. (2019). Early Crop Row Detection in Drill-Seeded Crops Based on Inverse Perspective Transformation (基于逆透视变换的条播作物早期作物行识别). *Journal of Jiangsu University (Natural Science Edition)*, Vol. 40, pp. 668–375, Jiangsu/China.