




Article

MLGNet: Multi-Task Learning Network with Attention-Guided Mechanism for Segmenting Agricultural Fields

Weiran Luo ^{1,2}, Chengcai Zhang ^{1,2}, Ying Li ^{3,4,*}  and Yaning Yan ^{1,2}¹ School of Water Conservancy and Civil Engineering, Zhengzhou University, Zhengzhou 450001, China² Yellow River Laboratory, Zhengzhou University, Zhengzhou 450001, China³ Henan Institute of Meteorological Sciences, Zhengzhou 450003, China⁴ CMA-Henan Agrometeorological Support and Applied Technique Key Laboratory, Zhengzhou 450003, China

* Correspondence: walnutclip@163.com

Abstract: The implementation of precise agricultural fields can drive the intelligent development of agricultural production, and high-resolution remote sensing images provide convenience for obtaining precise fields. With the advancement of spatial resolution, the complexity and heterogeneity of land features are accentuated, making it challenging for existing methods to obtain structurally complete fields, especially in regions with blurred edges. Therefore, a multi-task learning network with attention-guided mechanism is introduced for segmenting agricultural fields. To be more specific, the attention-guided fusion module is used to learn complementary information layer by layer, while the multi-task learning scheme considers both edge detection and semantic segmentation task. Based on this, we further segmented the merged fields using broken edges, following the theory of connectivity perception. Finally, we chose three cities in The Netherlands as study areas for experimentation, and evaluated the extracted field regions and edges separately, the results showed that (1) The proposed method achieved the highest accuracy in three cities, with IoU of 91.27%, 93.05% and 89.76%, respectively. (2) The Qua metrics of the processed edges demonstrated improvements of 6%, 6%, and 5%, respectively. This work successfully segmented potential fields with blurred edges, indicating its potential for precision agriculture development.

Keywords: agricultural fields; remote sensing images; multi-task learning; edge detection; semantic segmentation



Citation: Luo, W.; Zhang, C.; Li, Y.; Yan, Y. MLGNet: Multi-Task Learning Network with Attention-Guided Mechanism for Segmenting Agricultural Fields. *Remote Sens.* **2023**, *15*, 3934. <https://doi.org/10.3390/rs15163934>

Academic Editors: Enrico Corrado Borgogno Mondino, Filippo Sarvia, Samuele De Petris and Tommaso Orusa

Received: 17 June 2023
Revised: 31 July 2023
Accepted: 7 August 2023
Published: 8 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Agricultural fields serve as a vital pillar for the development of modern agriculture, promoting the advancement of agriculture towards greater efficiency, intelligence, and sustainability. Precise delineation of agricultural fields can provide basic data for agricultural production and management [1]. This valuable information enables agricultural producers to implement precision practices, such as targeted fertilization, precise irrigation, and pest monitoring, maximizing crop yields while reducing resource wastage [2,3]. During the early stages, the agricultural fields extraction required manual delineation. Although this way is capable of obtaining high-precision fields, it demands a considerable investment of human resources and time, greatly limiting the application of the data [4,5]. In recent years, satellite technology has made significant progress and development, especially the successful launch of high-resolution satellites, which has provided powerful data support [6–8].

Currently, there is growing interest in methods for extracting agricultural fields, which can be broadly classified into edge detection and region segmentation methods [9,10]. The edge-based methods use predefined kernels to perform convolution operations on images and detect object edges based on the change of spatial gradients. Turker et al. used the Canny operator to detect edge pixels and divided the fields into several sub-regions

using the extracted edges [11]. Yan et al. used a geometric contour model to segment multi-temporal Landsat images and divided field parcels into multiple independent sub-regions using the watershed segmentation algorithm [12]. Graesser et al. proposed a time-series-based method for extracting agricultural fields, which used multi-directional convolution kernels to obtain edge information of objects, the field parcels were segmented by using morphologically processed edges [13]. However, the above methods are limited by the type of convolution kernel and are sensitive to high-frequency noise. The region segmentation methods divide images into several sub-regions based on the similarity and mutual exclusivity of local features such as texture and color. Pedrero et al. used the simple linear iterative clustering method for super-pixel segmentation and employed supervised classification to merge adjacent regions [14]. Su et al. introduced a refined methodology based on mean-shift for farmland segmentation, which utilized a hybrid filter to ensure the homogeneity of internal pixels and the continuity of edges. This method improved the accuracy of farmland segmentation by using region merging techniques [15]. The region-based segmentation methods are highly dependent on parameters, which can lead to over-segmentation of internal regions with large differences and under-segmentation of smaller regions that may be overlooked [9].

Convolutional neural networks (CNNs) have played a pivotal role in the development of intelligent interpretation of remote sensing, which has been greatly facilitated by the rapid advancement of computer hardware and deep learning technology [16–19]. The CNNs have received much attention in the extraction of agricultural fields. Waldner et al. proposed a multi-task semantic segmentation model that could simultaneously perform the tasks of field segmentation, edge detection and learning of boundary distance features. Finally, the merged parcels were split by a watershed segmentation algorithm [20]. In addition, Long et al. proposed a multi-task learning network named BsiNet, which also learned segmentation, edge and distance tasks, this method enhanced the network's representation learning ability through a spatial grouping enhancement module [21]. To boost the accuracy of boundary extraction, Jong et al. used a generative adversarial network as a discriminator to assist in training ResUNet. Experimental results showed that this method improved the adaptability of the network [22]. The aforementioned methods have played important roles in agricultural fields extraction. However, they only use the high-level features generated by the last layer of the encoder, ignoring the guiding ability of feature aggregation [23,24]. Owing to the complex background, it is evident that the detected edges are often incomplete or isolated, and the above methods do not fully consider the respective advantages of segmentation and edge detection, making it difficult to obtain fine-grained agricultural fields.

The aim of this work is to explore the method of segmenting agricultural fields using the advantages of representation learning in deep learning. To achieve this, a multi-mask learning network with attention-guided mechanism (i.e., MLGNet) was proposed for agricultural fields extraction, which can learn complementary details in a progressive approach and improve the network's representation capacity. The proposed approach involves using a multi-task learning scheme to simultaneously train networks for semantic segmentation and edge detection tasks, which facilitates the exchange of information between different tasks, enabling the network to better generalize the acquired features. Finally, the broken edges are utilized to divide the merged fields based on the Gestalt laws, which helps rectify topological connectivity limitations of the network. To sum up, the contributions can be summarized as follows:

- (1) The MLGNet employs a guided attention fusion module to progressively learn edge details, thereby guiding the network to enhance region of targets. The learnable distance features are employed as a shared carrier for learning the segmentation and edge detection tasks.
- (2) A regional edge connectivity algorithm (i.e., ReCA) is designed based on principles of visual perception, which employs broken edges from detect task to divide merged fields into several sub-regions.

- (3) The effectiveness of various methods is compared, and the final results are evaluated based on edge and region indicators.

2. Materials and Methods

2.1. Study Area and Data

The study area is located in The Netherlands, which is known for its efficient, modern and sustainable agriculture. The government has established an agricultural information system (Basisregistratie Gewaspercelen) for the purpose of regulating and managing agricultural production. This system has made available fine-grained vector data on agricultural fields (<https://www.nationaalgeoregister.nl/geonetwork>) that is highly suitable for testing the effectiveness of the proposed method. This work utilized two cloud-free synthesized google images (18,000*18,000) with a resolution of 2 m to generate the samples, the data (<https://www.google.com/earth>) was obtained from the period of 1 January 2019 to 31 December 2020, and the location of the images is depicted in Figure 1. The images were sliced into 2400 patches, each with a size of 512*512 pixels. Out of these patches, 1800 were employed for training, and 600 were kept for validation.

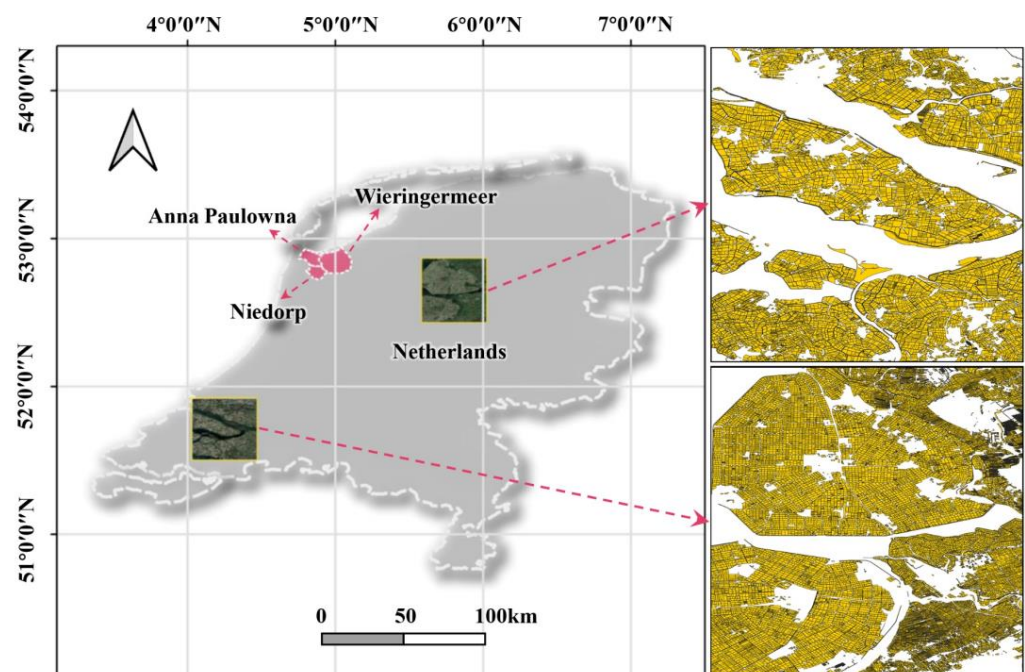


Figure 1. The overview of the study area. Anna Paulowna, Wieringermeer and Niedorp are three municipalities located in the province of North Holland in The Netherlands.

As experimental study sites, we selected three agricultural cities (i.e., Anna Paulowna, Wieringermeer and Niedorp) with notable differences, and the selection considered factors such as the size of fields, the density of fields, and the blurriness of edges. Anna Paulowna covers an area of approximately 64 square kilometers, and it is known for its beautiful tulip fields. In the northwest of Anna Paulowna, the agricultural fields show a characteristic of dense distribution, and the size of them is relatively small. Due to resolution limitations, the edges between these fields may appear blurry or less defined. Wieringermeer covers an area of approximately 165 square kilometers. It is known for its agricultural industry. In Wieringermeer, the agricultural fields present a fairly regular appearance, and the scale is relatively large. Niedorp covers an area of approximately 86 square kilometers. It is located in the northern part of North Holland and is known for its historic buildings and beautiful countryside. In the western of Niedorp, the fields do not have such a regular pattern, and the edges between the fields are blurry, making them difficult to be identified. Moreover, the google imagery of these cities are from different time periods. Although

color balancing has been applied to the synthesized imagery, color differences can still be noticed. Thus, these experimental areas can effectively evaluate the performance of the network. To enhance the variety of the samples, all training data were subjected to data augmentation, including random rotation, scaling and flipping, the probability of each augmentation strategy was set to 0.2 and the batch size of the training samples was set to 8.

2.2. Methods

2.2.1. Architecture of MLGNet

In the MLGNet network, adaptive channel fusion module (i.e., ACFM) and attention-guided fusion module (i.e., AGFM) are designed to integrate multi-scale semantic and detail information. As depicted in Figure 2, this architecture mainly consists of four parts: encoder, decoder, multi-scale branches and multi-task learning scheme. The encoder mainly refers to the architecture of ResNet34 [25], the residual block is followed by a corresponding down-sampling layer, and the bottom layer of the encoder is input to stacked atrous convolution module (SACM) to expand the model’s receptive field [26]. The SACM consists of four layers of dilated convolution with a kernel size of 3×3 , dilation factors are set to 1, 2, 4, and 8 respectively. Each decoder block contains two convolutional layers and one deconvolutional layer. The encoder and decoder are connected through ACFM modules, and the output features of each ACFM module are passed through a series of up-sampling layers to the AGFM modules. In the multi-scale branches, the AGFM module is used to guide the network to supplement edge details layer by layer. The multi-task learning scheme (i.e., distance task, edge task and segmentation task) are added to the end of the network to improve its generalization.

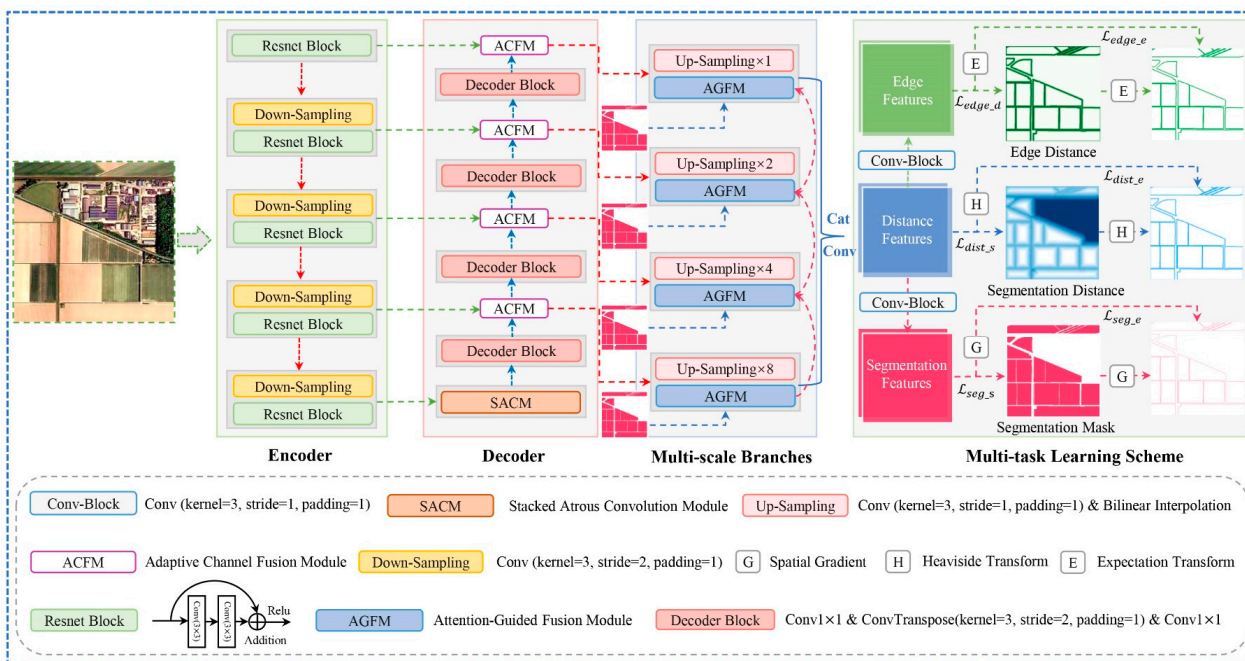


Figure 2. Illustration of the MLGNet architecture.

(1) Adaptive channel fusion module

The ACFM Module is mainly inspired by the Channel Attention Network [27]. The structure of the ACFM is shown in Figure 3. The Global Average Pooling (GAP) is employed to compress the encoder and decoder features in the spatial dimension. The compressed features have a global receptive field, meaning that the entire spatial information on a channel is compressed into a single global feature. Assuming that the output feature of the l -th layer from the encoder is $U_e^{(l)} \in \mathbb{R}^{H \times W \times B}$, and the output feature of the l' -th layer from

the corresponding decoder is $U_d^{(l)} \in \mathbb{R}^{H \times W \times B}$, the compressed encoder feature $z_e^{(l)} \in \mathbb{R}^{1 \times B}$ and decoder feature $z_d^{(l)} \in \mathbb{R}^{1 \times B}$ can be formulated as:

$$z_e^{(l)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W U_e^{(l)}(i, j) \tag{1}$$

$$z_d^{(l)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W U_d^{(l)}(i, j) \tag{2}$$

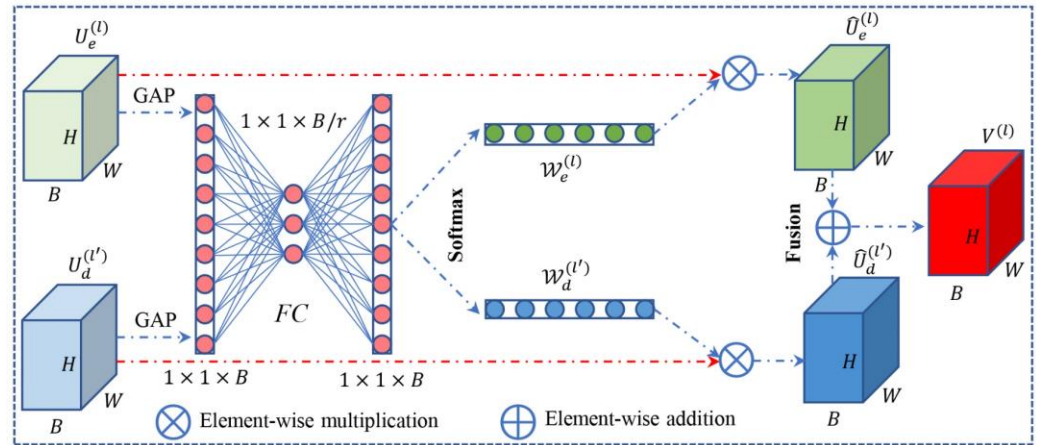


Figure 3. The schematic diagram of ACFM.

To learn more compact features, a fully connected network is used to capture the non-linear relationships between channels in $z_e^{(l)}$ and $z_d^{(l)}$. The compact features can be articulated by the following mathematical notation:

$$\mathcal{F}_e^{(l)} = ReLU\left(\theta_{a1}^{(l)} \times \left(z_e^{(l)}\right)^T\right) \tag{3}$$

$$\mathcal{F}_d^{(l)} = ReLU\left(\theta_{a1}^{(l)} \times \left(z_d^{(l)}\right)^T\right) \tag{4}$$

where $\theta_{a1}^{(l)} \in \mathbb{R}^{d \times B}$ represents the parameters of the first fully connected layer in the adaptive fusion module corresponding to the l -th layer encoder. $ReLU(\cdot)$ represents a non-linear activation function, and T denotes the transpose symbol used for matrix transposition. The fully connected network is a self-encoding structure, and the learning of features primarily benefits from the intermediate hidden layer. This layer compresses the dimensions to $d = B/r$ using a scaling factor r , and then restores them back to the original dimension B . Finally, the Softmax function is used to compute the weights connecting the encoding layer and the decoding layer:

$$W_e^{(l)} = \frac{e^{\theta_{a2}^{(l)} \times \mathcal{F}_e^{(l)}}}{e^{\theta_{a2}^{(l)} \times \mathcal{F}_e^{(l)}} + e^{\theta_{a2}^{(l)} \times \mathcal{F}_d^{(l)}}} \tag{5}$$

$$W_d^{(l)} = \frac{e^{\theta_{a2}^{(l)} \times \mathcal{F}_d^{(l)}}}{e^{\theta_{a2}^{(l)} \times \mathcal{F}_e^{(l)}} + e^{\theta_{a2}^{(l)} \times \mathcal{F}_d^{(l)}}} \tag{6}$$

where $\theta_{a2}^{(l)} \in \mathbb{R}^{B \times d}$ represents the parameters of the second fully connected layer in the adaptive fusion module corresponding to the l -th layer encoder. The fusion module

combines the features of the encoder and decoder adaptively using weight indicators. As a result, the fused feature $V^{(l)} \in \mathbb{R}^{H \times W \times B}$ can be expressed as:

$$V^{(l)} = \mathcal{W}_e^{(l)} \cdot U_e^{(l)} + \mathcal{W}_d^{(l)} \cdot U_d^{(l)} \tag{7}$$

(2) Attention-guided fusion module

Shallow networks can capture low-level spatial details but tend to lose semantic information, while deep networks are just the opposite. Existing studies have shown that there are specific differences in details among features of different scales, auxiliary supervised tasks encourage the network to learn hierarchical representations, leading to better capture these differences [28,29]. For this purpose, the AGFM module is designed to better fuse semantic and detail information of different scales, which adopts a forward learning mechanism from deep to shallow layers to capture the lost detail information. To be more specific, we use bottom-up predictions to gradually erase high-confidence non-edge regions layer by layer, and steer the network towards learning complementary features by using ground-truth masks, which are often distributed around the edges of the targets.

The network architecture of the AGFM is visualized in Figure 4, assuming that the fused feature from the l -th layer encoder and l' -th layer decoder is denoted as $V^{(l)} \in \mathbb{R}^{H \times W \times B}$, the segmentation prediction from adjacent bottom layers can be denoted as $Q^{(l+1)} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 1}$, which is a non-probabilistic logit. We generate the attention weight feature $\mathcal{W}_u^{(l+1)}$ based on $Q_u^{(l+1)} \in \mathbb{R}^{H \times W \times 1}$, which is upsampled by a factor of 2 using bilinear interpolation, $\mathcal{W}_u^{(l+1)}$ is obtained by performing edge transformation with the segmentation probability $Q_u^{(l+1)}$. Supposing the predicted segmentation probability is $P_u^{(l+1)}$. The edge transformation can be written as:

$$\mathcal{W}_u^{(l+1)} = 1 - 2 \times |P_u^{(l+1)} - 0.5| \tag{8}$$

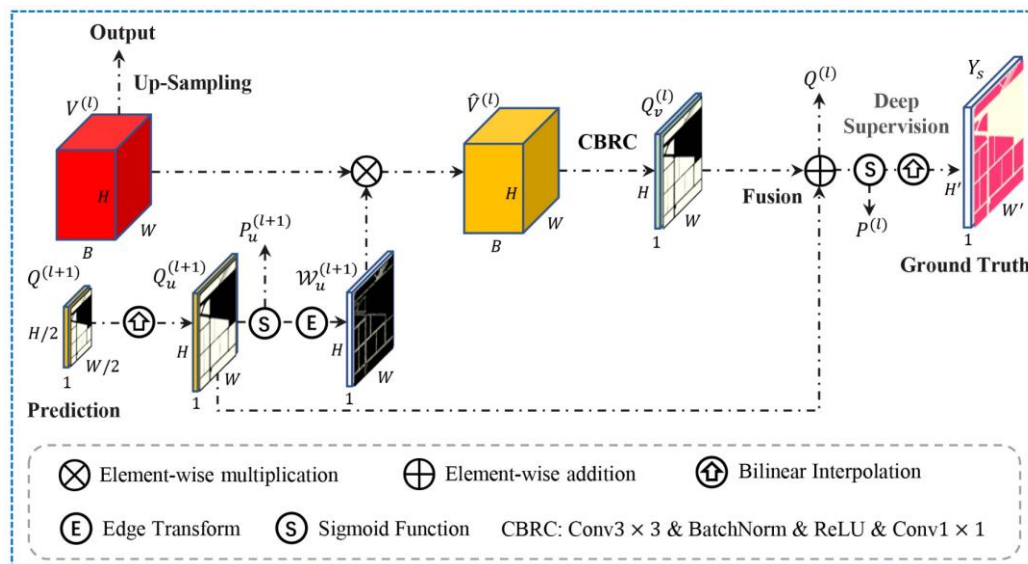


Figure 4. The schematic diagram of AGFM.

It is evident from the Equation (8) that when the segmentation probability of a pixel is close to 0.5, it indicates an unreliable segmentation prediction. In this case, the weight value tends towards 1. On the other hand, when the segmentation probability of a pixel is

close to 0 or 1, it indicates a high confidence in the prediction, and the weight value tends towards 0. The fused feature $\hat{V}^{(l)} \in \mathbb{R}^{H \times W \times B}$ can be described as:

$$\hat{V}^{(l)} = \mathcal{W}_u^{(l+1)} \times V^{(l)} \tag{9}$$

The unreliable predictions are usually found in the edge regions. We perform weight fusion on adjacent shallow features and use the fused feature $\hat{V}^{(l)}$ to perform convolution to obtain the segmentation prediction $Q_v^{(l)}$. To compensate for the lost detailed information in the high-level features, we execute addition operation on $Q_v^{(l)}$ and $Q_u^{(l+1)}$, and use the ground-truth mask $Y_s \in \mathbb{R}^{H' \times W'}$ to supervise the multi-level side outputs. Therefore, the total loss of AGFM can be formulated as:

$$\mathcal{L}_{guide} = \sum_{l=1}^L \ell_{wce}(\mathcal{B}(P^{(l)}), Y_s) + \ell_{jac}(\mathcal{B}(P^{(l)}), Y_s) \tag{10}$$

where $\ell_{wce}(\cdot)$ represents the weighted cross-entropy loss [30], which is advantageous for semantic segmentation that involve class imbalance. ℓ_{jac} represents the jaccard loss [31], and $\mathcal{B}(\cdot)$ represents the bilinear interpolation function. The AGFM is a deep supervision learning mechanism that not only accelerates the convergence of the model, but also enhances the model’s representation ability.

2.2.2. Multi-Task Learning Scheme

In general, the agricultural fields extraction often adopts a single segmentation task. Although this learning method can achieve acceptable results, which ignores other information related to segmentation, and it is a formidable challenge to substantially improve the accuracy [32]. The multi-task learning can explore the potential relationships between different tasks, this scheme helps the network learn more general shared representations, and improves the network’s generalization performance and inference speed. It has been widely used in agricultural field segmentation [33,34]. To improve the accuracy of segmentation, three learnable tasks (distance task, segmentation task and edge detection task) are added to the end of the network. The distance map represents the nearest distance from any pixel in the image to the target boundary. As a point moves away from the target edge, the distance value increases, and as it approaches the edge, the distance value approaches 0. Based on the learned distance map, we can not only obtain the boundary information of the target, but also obtain more connected segmentation regions. Therefore, the learnable distance features are used as a shared carrier for segmentation and edge detection tasks. The accuracy of region segmentation and edge detection significantly affects the learning of distance features, while distance features can improve the accuracy of semantic segmentation and edge detection. This scheme ensures the correlation between tasks to enhance their performance.

(1) Signed distance loss

To achieve flexible conversion between distance maps and segmentation maps (or edge maps), a Signed Distance Function (SDF) is employed for distance map calculation [35]. The SDF is defined as follows:

$$D(\vec{m}) = \begin{cases} \inf_{\vec{y} \in \Omega} \|\vec{m} - \vec{n}\| & \vec{m} \in \Omega_{in} \\ -\inf_{\vec{y} \in \Omega} \|\vec{m} - \vec{n}\| & \vec{m} \notin \Omega_{in} \end{cases} \tag{11}$$

where \vec{m} represents an anchor point within the region, and \vec{n} represents an arbitrary point on the boundary. When \vec{m} is inside the contour, the distance value is positive, and when m is outside the contour, the distance value is negative. Deep learning possesses excellent representation learning capabilities, enabling it to easily learn the mapping relationship

of distance. In this mapping relationship, the distance map is uniformly encoded into a one-hot format, and the encoded distance feature can be learned through classification. To better adapt the network’s learning, the truncated SDF with an integer threshold τ is adopted to calculate the distance of ground-truth, and these distance values are converted into non-negative integers in a special way. Thus, the converted distance $D_\tau \in \mathbb{R}^{H' \times W' \times 2\tau}$ can be calculated using the following formula:

$$D_\tau = (D + \tau) \times \mathbf{1}_{(-\tau \leq D \leq 0)} + (D + \tau - 1) \times \mathbf{1}_{(0 < D \leq \tau)} + (2\tau - 1) \times \mathbf{1}_{(D > \tau)} \quad (12)$$

where $D \in \mathbb{R}^{H' \times W'}$ representation signifies the signed distance map, $\mathbf{1}_{(\cdot)}$ represents a hard confidence threshold function. The value corresponds to 1 when the expression is satisfied, and 0 otherwise.

The estimation of encoded distance can be viewed as a pixel-level classification task. However, due to significant differences in the number of pixels between the interior and exterior of the target, the distribution of truncated distance values becomes highly imbalanced. To mitigate this imbalance, the weighted cross-entropy is also employed as the loss function:

$$\mathcal{L}_{dist_s} = \ell_{wce}(\hat{H}_\tau, H_\tau) \quad (13)$$

where H_τ is the one-hot encoding of D_τ , and \hat{H}_τ represents the predicted probabilities. To ensure the accuracy of the edge positions in the predicted distance map, a heaviside function [36] is used to convert the predicted distances into edge probabilities. Since \hat{H}_τ represents the predicted probabilities of distances, before performing the edge transformation, the predicted probabilities need to be converted into the expected distances using the following formula:

$$E(\hat{D}_\tau) = \sum_{k=0}^{2\tau-1} k \times \hat{H}_\tau^k \quad (14)$$

where \hat{H}_τ^k represents the probability of the predicted label belonging to the k -th class. It can be seen from the Equation (14) that the expected distance is calculated by taking a weighted average of the predicted probabilities for the classification labels. Therefore, the probability of the margin can be represented using the following Heaviside transform.

$$\mathcal{H}(E(\hat{D}_\tau)) = 2 - 2 \times \tanh(|E(\hat{D}_\tau) - \tau|/\epsilon) \quad (15)$$

where ϵ represents a hyperparameter used to adjust the distribution of the edge probability. The Equation (15) makes a simple adjustment to the original Heaviside function, where the margin probability approaches 1 when $E(\hat{D}_\tau) \rightarrow \tau$, otherwise the function rapidly increases to 0. This function has the desirable property of converting the truncated distance function into an edge probability. To ensure the accuracy of the edge positions in the predicted distance map, the following formula is used to constrain the margins of the distance map:

$$\mathcal{L}_{dist_e} = \ell_{smoothL1}(\mathcal{H}(E(\hat{D}_\tau)), \mathcal{H}(E(D_\tau))) \quad (16)$$

where $\ell_{smoothL1}(\cdot)$ represents the smooth L1 loss function [37]. Therefore, the final loss with respect to the distance map can be expressed as:

$$\mathcal{L}_{dist} = \lambda_{dist1} \cdot \mathcal{L}_{dist_s} + \lambda_{dist2} \cdot \mathcal{L}_{dist_e} \quad (17)$$

where λ_{dist1} and λ_{dist2} represent hyperparameters used to balance the various loss terms.

(2) Segmentation loss

The segmented image is obtained through convolutional operations based on the distance features. The segmentation loss function is calculated in the same way as Equation (10), it can be expressed as:

$$\mathcal{L}_{seg_s} = \ell_{wce}(\hat{P}_s, Y_s) + \ell_{jac}(\hat{P}_s, Y_s) \quad (18)$$

where $\hat{P}_s \in \mathbb{R}^{H' \times W'}$ represents the final predicted segmentation probability. The boundary information of the segmentation is obtained from the predicted segmentation map using spatial gradients. The gradient convolution kernel uses the Laplacian operator. A consistency constraint is added between the spatial gradient of the segmentation probability and the true gradient to ensure the accuracy of the edges. The Smooth L1 loss is used as the regularization term, and its expression is as follows:

$$\mathcal{L}_{seg_e} = \ell_{smoothL1}(\nabla \hat{P}_s, \nabla Y_s) \quad (19)$$

where ∇ represents the Laplace operator. The complete segmentation loss can be formulated as:

$$\mathcal{L}_{seg} = \lambda_{seg1} \cdot \mathcal{L}_{seg_s} + \lambda_{seg2} \cdot \mathcal{L}_{seg_e} \quad (20)$$

where λ_{seg1} and λ_{seg2} represent hyperparameters used to balance the various losses.

(3) Buffered edge Loss

Differing from conventional edge loss, a buffered distance is introduced to the edge as the target for network learning. When the absolute value of the distance is less than or equal to the buffer threshold η , it indicates the edge region, otherwise, it is considered a non-boundary region. Smaller distance values indicate closer proximity to the edge.

Similarly, the distance map with one-hot format is more readily accepted by the classifier. The buffered distance can be computed using the following formula:

$$D_\eta = |D| \times \mathbf{1}_{(|D| \leq \eta)} + \eta \times \mathbf{1}_{(|D| > \eta)} \quad (21)$$

The edge detection task is considered as an optimization of the buffer distance loss.

To alleviate the issue of loss imbalance, the weighted cross entropy is used for boundary distance loss, and the expected transformation (i.e., edge distance) is added to the regularization constraint term. Its formula is defined as follows:

$$\mathcal{L}_{edge_d} = \ell_{wce}(\hat{H}_\eta, H_\eta) \quad (22)$$

$$\mathcal{L}_{edge_e} = \ell_{smoothL1}(E(\hat{D}_\eta), E(D_\eta)) \quad (23)$$

where H_η represents the true boundary distance, which is the one-hot encoded format of D_η , and \hat{H}_η represents the predicted probability. Therefore, the final loss with respect to the distance map can be expressed as:

$$\mathcal{L}_{edge} = \lambda_{edge1} \cdot \mathcal{L}_{edge_d} + \lambda_{edge2} \cdot \mathcal{L}_{edge_e} \quad (24)$$

where λ_{edge1} and λ_{edge2} denote hyperparameters used to balance the losses.

The above formulas involve many hyper-parameters, and it is difficult to find the optimal value unless we try many manual adjustments. To reduce manual parameter setting, a multi-task learning method is used to adaptively adjust the balanced parameters in all loss terms [38], which is a measure of the importance of each task by homoscedastic uncertainty, and the optimization process considers maximizing the Gaussian likelihood function and introduces the uncertainty noise parameters σ . The objective function can be written as:

$$\mathcal{L} = \sum_k \frac{1}{2\sigma_k^2} \cdot \mathcal{L}_k + \log\left(\prod_k \sigma_k\right) \quad (25)$$

where \mathcal{L}_k denotes the loss term corresponding to λ_k , $\sigma_k > 0$ means the noise parameter to be learned, σ_k can be understood as a learnable parameter corresponding to λ_k , the parameter σ_k can be optimized by minimizing the formula so that the tasks can be balanced. It can be seen from the Equation (25) that the homoscedastic uncertainty for this task increases as σ_k increases, and small weights are assigned to corresponding tasks that are hard to learn

due to high noise. This type of multi-task learning approach can significantly decrease the reliance on parameters and enhance the network's automated learning ability.

2.2.3. Dividing Fields with Broken Edges

Although semantic segmentation can obtain complete field information, the detection results for fuzzy edges are often interrupted, resulting in multiple sub-regions being merged together. To obtain refined fields, a regional edge lines connectivity algorithm (ReCA) is proposed, in which we divide the field units into several sub-regions using interrupted edge lines. Some of the edge lines are broken but can still be used, which can be joined or extended so that they can form closed sub-regions. To ensure the continuity of the edge lines, some perceptual rules are developed in the ReCA algorithm. According to Gestalt laws [11,39], humans are capable of organizing and arranging the position of objects in vision, and perceiving wholeness and continuity of the environment. Therefore, the designed method also follows some rules of the Gestalt laws when constructing sub-regions, including proximity, continuity, and closure.

Before connecting the edge lines, the whole external contours need to be extracted from the segmented probability map. Firstly, the initial segmentation mask is obtained using the morphological segmentation algorithm [40]. Then, the border following algorithm is applied to detect the external contours of all fields. [41]. Finally, the extracted contours are fused with an outer buffer to suppress the elongated edges, and the original boundaries are restored utilizing an inner buffer. Although these edges are isolated or interrupted, they can still be used to split the fields. In the ReCA algorithm, several rules are developed to connect and extend these broken edge lines to form closed sub-regions. Note that these edge lines are the skeleton lines of the edge, and the internal edge lines require individual processing for each field, rather than calculating all fields together.

Rule 1: For some shorter edge lines, forcing closure or connection may result in erroneous results, so it is necessary to remove these lines. Figure 5a shows an unbranched edge line AB . If its length satisfies the condition $len(AB) < \varepsilon$, then AB is removed. Figure 5b shows a branched edge line that is decomposed into several unbranched lines (OA , OB and OC) based on the crossing point O . If the decomposed line OA satisfies the condition $len(OA) < \varepsilon$, then OA is removed. Here, the length refers to the number of pixels in a connected edge line.

Rule 2: It is clear from the principle of continuity that people tend to perceive continuous objects rather than discrete forms. To maintain the continuity of the edge, we connect two approximately collinear edge lines and extend the edge lines in the direction of the endpoints to the outer boundary of the field, forming a closed sub-region. Collinearity is determined based on the angle between the broken lines and the distance from the endpoints to the line on which they lie. Figure 5c shows a set of isolated edge lines (AB and CD). Assuming that the minimum angle between AB and CD is α , the maximum distance from break points A and B to the line on which CD lies is $d_{AB \rightarrow CD}$, and the maximum distance from break points C and D to the line on which AB lies is $d_{CD \rightarrow AB}$. If $\alpha < \vartheta$ & $max(d_{CD \rightarrow AB}, d_{AB \rightarrow CD}) < \varepsilon$, then AB and CD are considered collinear, where collinear means that they share the same equation of a straight line. The break point is determined by counting the number of 8-neighborhood on the skeleton line with a value of 1. If the number of pixels is equal to 1, then the point is a break point.

Rule 3: The principle of closure shows that the visual system automatically tries to close up open graphics. Figure 5d represents an unbranched edge line. Suppose all points on the contour line AD have a maximum distance $d_{AD \rightarrow L}$ to the fitted straight line L . If $d_{AD \rightarrow L} < \varepsilon$, the edge line AD can be considered as a straight segment. The edge line is extended to the outer boundary along the direction of the straight segment. Otherwise, we calculate the directions of points A and D based on their neighboring points and extend them to the outer boundary. Figure 5e represents a branched edge line. The branch line is decomposed into multiple unbranched lines at the branch point O , and then extend the contour lines in the same manner as in Figure 5d. The branch point is determined

by counting the number of 8-neighborhood on the skeleton line with a value of 1. If the number of pixels is greater than or equal to 3, then the point is a branch point.

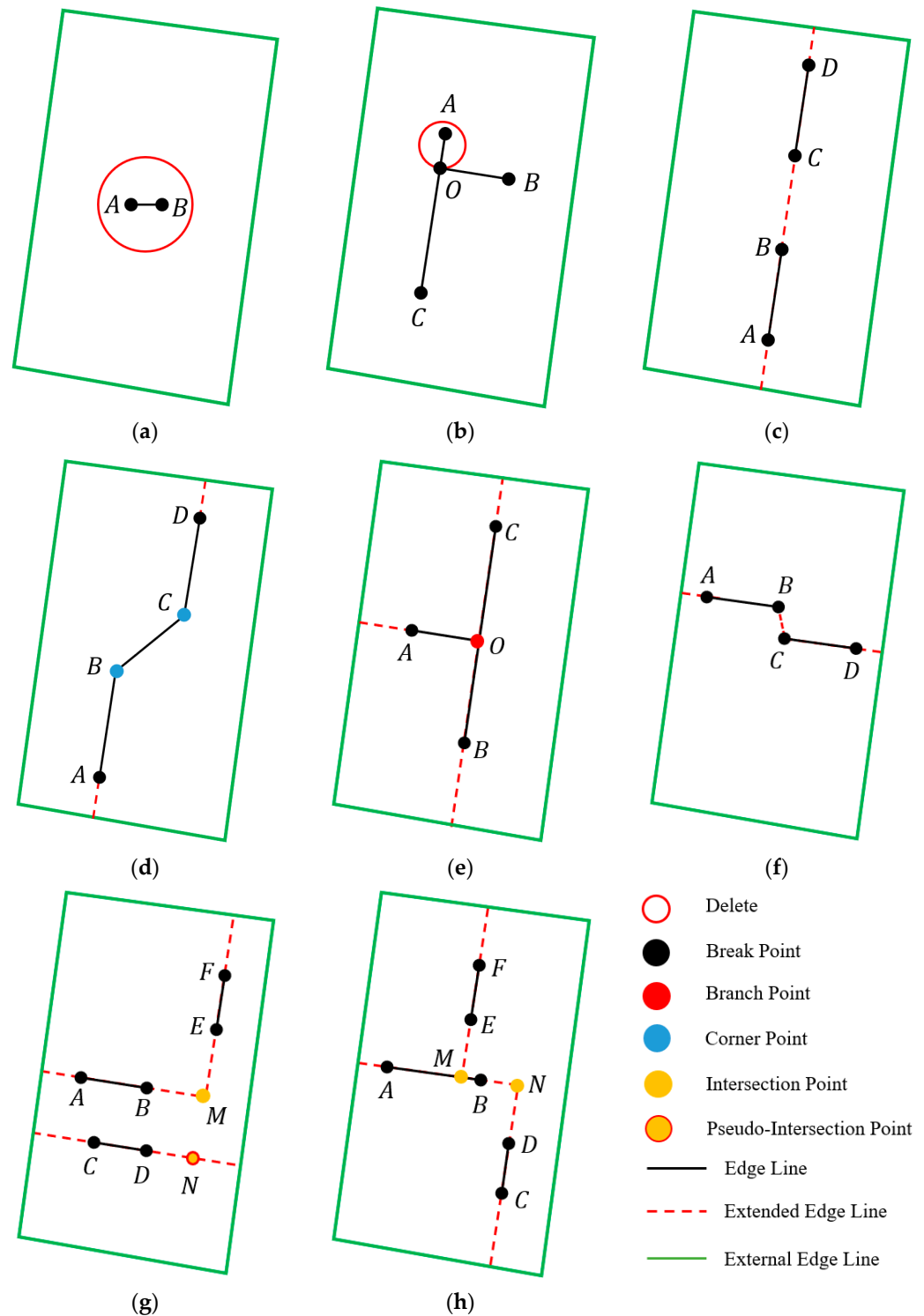


Figure 5. The rules of topological connectivity. (a–h) represent various potential edge connectivity scenarios, including deletion, connection, extension and intersection.

Rule 4: According to the proximity principle, the perceptions of objects in a perceptual field are grouped together according to the proximity. Figure 5f illustrates two adjacent breakpoints (B and C), if the distance d_{BC} between points B and C satisfies $d_{BC} < \epsilon$, the two points can be joined directly, here ensuring that the breakpoints are non-connected.

Rule 5: Figure 5g,h illustrate a complex situation concerning multiple edge lines. In Figure 5g, the extension lines of edge lines AB and EF intersect at point M. It should be noted that there is also an intersection point N between EF and CD. However, since there is no edge line between M and N, and N is not adjacent to any other breakpoints along the direction of FE, N needs to be removed. Given the proximity of M and E, it is essential to maintain their connection, while ensuring the interconnectivity between AB and EF, so they are no longer extended after the intersection. In Figure 5h, as both M and N have neighboring breakpoints A and B along the AB direction, it is necessary to preserve both intersection points M and N. Moreover, since point N establishes a connection between AB and CD, so they are no longer extended after the intersection.

To provide a clearer description of the ReCA algorithm, a detailed step-by-step process is provided. The first step is to traverse each whole field and obtain its internal skeleton lines, then connect the closer breakpoints according to Rule 4. To avoid interference with the results, some short lines have been removed according to Rule 1. After that, we detect the break points of the skeleton lines and calculate the equation for the corresponding points. The breakpoints are grouped based on collinearity, and the intersection points are calculated between non-collinear equations. The collinearity here is based on Rule 2. The second step is to sort all points within the same group along a common direction, and then delete pseudo-intersection points according to Rule 5. If the intersection points are inside the field, they will be inserted into the corresponding group. If the points on either side of the sorted sequence are not intersection points, the points on both sides need to be extended to the boundary according to Rule 3. Otherwise, only connect the sorted points without extending the line. Finally, the whole field is divided into several sub-regions based on the connected lines.

3. Results

3.1. Experimental Settings

(1) Network architecture

Our network design was inspired by the DLinkNet architecture, with the key difference being that the encoding layer consisted of five residual modules with channel sizes of 32, 64, 128, 256, and 512, respectively. The first residual block was set to 1, while the remaining blocks followed the ResNet34 configuration with numbers of 3, 4, 6 and 3. The ACFM module used a uniform scaling factor of 8, and the output dimensions of the features from the four branches were all set to 32.

(2) Parameter settings

For all experiments, a momentum-based SGD optimization algorithm with a “poly” learning rate decay strategy was adopted to optimize the network [17], in which the initial learning rate, decay coefficient, total epochs and max epoch were set to 0.01, 0.9, 300 and 300 respectively. In the loss term, the distance threshold τ and the edge buffer threshold η were set to 32 and 8 respectively. In the ReCA algorithm, the distance error ε and angle error θ were set to 10 and 15 uniformly.

(3) Evaluation metric

To better evaluate segmentation performance, F1 score, intersection over union (IoU), recall and precision were used as evaluation metrics. Moreover, a buffer zone analysis was carried out on the edges of the fields, and the performance of the splitting edges was assessed using completeness (Com), correctness (Cor) and quality (Qua) [42], where Com and Cor represent the recall and precision of the edge buffer zones, respectively, and Qua is a comprehensive metric that encompasses both Com and Cor. Finally, the number of fields was used as an additional indicator, which evaluated the difference in the quantity of actual parcels and segmented parcels.

3.2. Agricultural Fields Extraction

Figure 6 shows the extraction results of agricultural fields. The composite map is a false-color representation created by overlaying distance feature, edge feature, and segmentation feature. The extracted fields are vectorized polygons using the ReCA algorithm. From a global perspective, it can be seen that the structures of extracted fields are complete through comparison with the imagery, and many small fields have been successfully segmented, the extracted fields exhibit high similarity with the ground truth map in terms of details. It is worth noting that the composite map is created by truncated distance map, buffered edge map and segmentation probability map, in which the semantic and detailed information of agricultural fields are more highlighted. To be more specific, it indicates that both large-scale and small-scale fields are capable of capturing edge details and holistic semantic information. The proposed network effectively integrates features encompassing a range of scales, leveraging the advantages offered by features at different scales. Specifically, the AGFM module is utilized to enhance the assimilation of complementary information, the network can be constrained and guided by different tasks, which facilitates the dissemination of information and enables the learning of more robust representations.

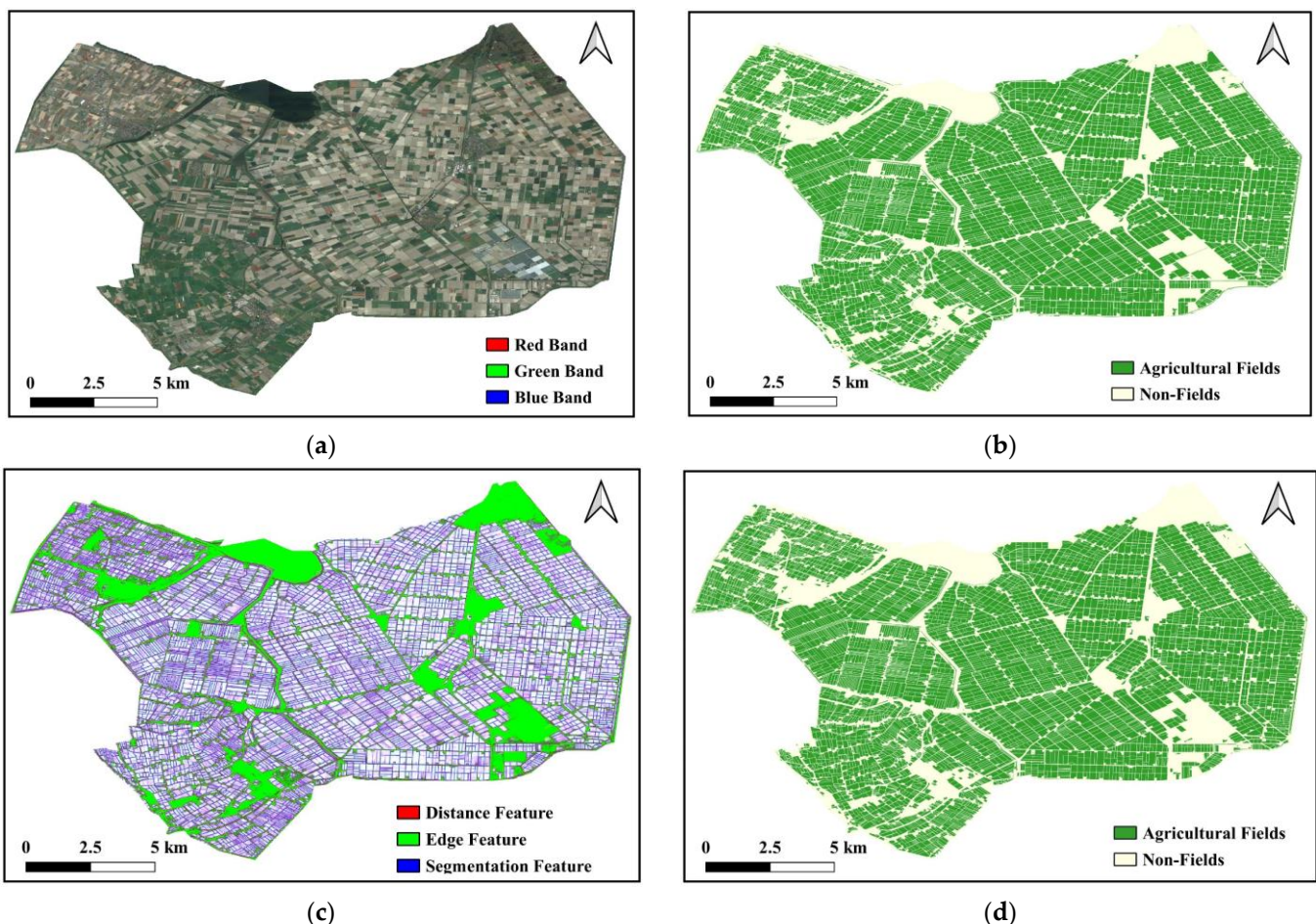


Figure 6. Visualization of agricultural fields. (a–d) represent google imagery, ground truth, composite map, and extracted fields.

From a local perspective (Figure 7), it is evident that there exists a certain correlation and complementarity among edge detection, distance estimation, and segmentation tasks, the segmentation map can obtain relatively complete fields, mainly because using the distance map as a carrier can ensure the integrity of the fields structure. Upon careful observation, edge detection is more capable of highlighting the edge details between

different fields, whereas the segmentation task primarily focuses on semantic information, potentially overlooking some faint intricacies. Hence, edge detection holds a relative superiority in highlighting edge details. Actually, it is evident that the significant edges between fields have been effectively extracted. However, some hard-to-discern edges are difficult to be fully captured, these edges are still interrupted. This is mainly due to the complexity of the scene, which weakens some texture features and makes it difficult to obtain continuous edges. In this scenario, it is challenging to split the merged fields no matter how the segmentation threshold is chosen. Actually, these broken edges are still very useful, as they can obtain connected fields based on the direction of the broken skeleton lines.

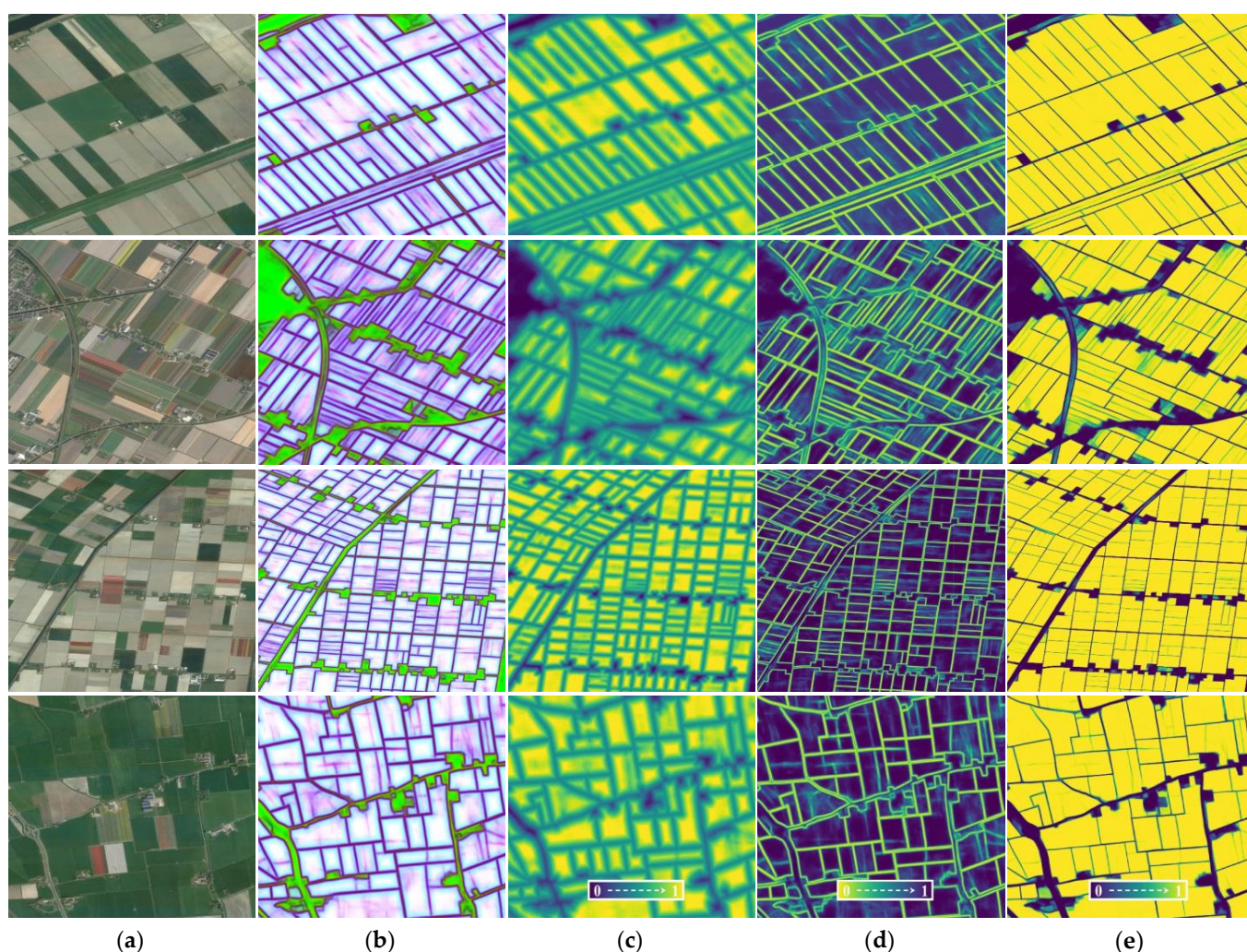


Figure 7. Examples of multi-task feature maps. (a–e) respectively represent google imagery, composite maps, truncated distance, buffered edge and segmentation probability. (c–e) represent the normalized results.

Figure 8 displays the final results using the ReCA algorithm. The whole fields needed to be obtained based on the segmentation mask with special topology processing, this way involved buffering only the external contours to merge some elongated edges. The potential benefit is to acquire whole fields without considering internal noise, and our primary focus was on how to effectively employ the available edge lines for segmenting the fields. The final processing results are shown in the white region in Figure 8b. To obtain reliable split lines, the skeleton lines were extracted from the buffered edges using a lower threshold (i.e., one-third of the buffered edge distance). This approach took advantage of the fact that values closer to the center of the buffered edge were lower, which helped

filter out a greater amount of noisy edges. The red lines in Figure 8b represented the final extracted skeleton lines. From Figure 8c, it can be clearly seen that many small fields had been divided, and they matched the actual parcels very well. Additionally, the algorithm had also detected some undivided fields that were not delineated in the ground truth. After verification, it was found that the detected results were correct. This indicated that the ground truth also had some omissions due to manual interference.



Figure 8. Examples of agricultural fields using ReCA algorithm. (a–d) represent google imagery, topological maps, final results and ground truth. The topological maps contain merged fields and red skeleton lines of disrupted edges.

Table 1 displays the evaluation results from three different cities. With the implementation of ReCA, the Com demonstrated improvements of approximately 7%, 6.5%, and 6.5%, respectively. A high value of Com indicates that the model is better at capturing true positives located at the edge regions, while the Qua simultaneously exhibited increases of around 6%, 6%, and 5%. This also indicates that the proposed method helps to improve the overall accuracy of the edges. A higher Cor signifies that the model can accurately detect

true positives located at the edges, and it is noticeable that there is a slight decrease in the Cor metric from Table 1. This suggests that the initial segmentation edges have higher accuracy. However, the refined results involved connecting or extending edges, which may introduce some errors when supplementing the edges. Nonetheless, the overall change in the Cor metric remains negligible. It is also observed that the number of extracted fields (Pre-N) was 1858, 3385, and 1602, while the corresponding reference counts (Ref-N) were 1750, 3367, and 1513, indicating a very small discrepancy in quantity. From a numerical perspective, the results of Pre-N exceeded the Ref-N by 50, 70, and 80, respectively. Upon comparing the Google image, it was discovered that some edges were missing in the ground truth map, resulting in some fields being merged together. The reason for this phenomenon is twofold: firstly, there were some omissions in the manual annotation process, and secondly, the data source used was cloud-free synthesized imagery from 2019–2020, which introduced a temporal difference between the ground truth and the imagery. Taken overall, these results demonstrate that the ReCA algorithm is indeed capable of dividing merged fields. The method combines the advantages of semantic segmentation and edge detection to achieve more precise delineation.

Table 1. Accuracy evaluation of field edges.

Cities	Com%	wo/ReCA			w/ReCA			Pre-N	Ref-N
		Cor%	Qua%	Pre-N	Com%	Cor%	Qua%		
Paulowna	68.46	82.93	60.09	1114	75.64	80.75	65.97	1858	1750
Wieringermeer	62.76	79.12	53.49	1429	69.29	78.34	59.83	3385	3367
Niedorp	65.00	79.32	55.64	942	71.45	78.39	61.00	1602	1513

wo/ReCA means the result without ReCA. w/ReCA means the result with ReCA.

3.3. Comparative Analysis

To assess the efficacy of the MLGNet, the study compared some advanced semantic segmentation methods, including ResUNet [43], DLinkNet [26], ResUNet-a [18] and BsiNet [21]. To ensure the fairness of the experiment, all comparison methods used the same optimization algorithm, batch size, and common samples. Table 2 summarized the evaluation metrics of different methods. The MLGNet achieved the highest IoU (i.e., 91.27%, 93.05%, and 89.76%) and F1 score (i.e., 95.44%, 96.40%, and 94.61%) in three cities. ResUNet-a and BsiNet had slightly lower IoU compared to MLGNet, which indicated that the network had a relatively large overlap between prediction and ground truth while achieving a good balance in predicting both positive and negative instances. Similarly, the proposed network achieved the highest recall of 96.98%, 97.24%, and 96.54%, respectively, indicating that it could better identify true positives. In terms of precision, DLinkNet outperformed MLGNet by approximately 0.9%, 0.1%, and 0.3%, respectively. Nevertheless, the proposed method continued to demonstrate superior accuracy compared to other approaches. Additionally, it exhibited significant potential in the extraction of agricultural fields.

To assess the variations among different methods, the partial segmentation images of different methods are shown in Figure 9. It can be seen that MLGNet performs better in learning agricultural features, the non-edge noises are effectively suppressed, and the edge details are more pronounced. The proposed method produces more complete segmentation, capturing both small elongated fields and large fields. From the image comparison, it is noticeable that the blue regions are predominantly located in areas with indistinct edges, making it challenging to separate these fields through semantic segmentation. In addition, from the third row in Figure 9, it can be seen that for some special types that are not easily distinguished, such as grasslands and fields, the blue area represents the extracted fields. In fact, these land types should be classified as grasslands, and this method can effectively improve this phenomenon, indicating that our method has high discriminability for non-fields. This is primarily due to the guiding effect of the AGFM module, which facilitates the network in learning complementary spatial details layer by layer. In addition, the multi-task learning method also helps the network to learn more refined features. These

results indicate that the proposed network framework and multi-task learning method are very effective.

Table 2. Accuracy evaluation of segmentation fields with different methods.

Cities	Methods	Evaluation Metrics (%)			
		IoU	F1	Recall	Precision
Paulowna	ResUNet	85.47	92.17	90.08	94.36
	DLinkNet	86.26	92.62	90.50	94.84
	ResUNet-a	89.39	94.40	95.76	93.08
	BsiNet	87.01	93.05	91.84	94.31
	MLGNet	91.27	95.44	96.98	93.94
Wieringermeer	ResUNet	90.71	95.13	94.78	95.49
	DLinkNet	91.29	95.45	95.20	95.70
	ResUNet-a	91.86	95.69	96.71	94.70
	BsiNet	90.19	94.83	94.00	95.68
	MLGNet	93.05	96.40	97.24	95.58
Niedorp	ResUNet	86.59	92.81	92.82	92.80
	DLinkNet	87.05	93.07	93.09	93.05
	ResUNet-a	88.11	93.68	95.99	91.49
	BsiNet	87.60	93.39	94.26	92.53
	MLGNet	89.76	94.61	96.54	92.75

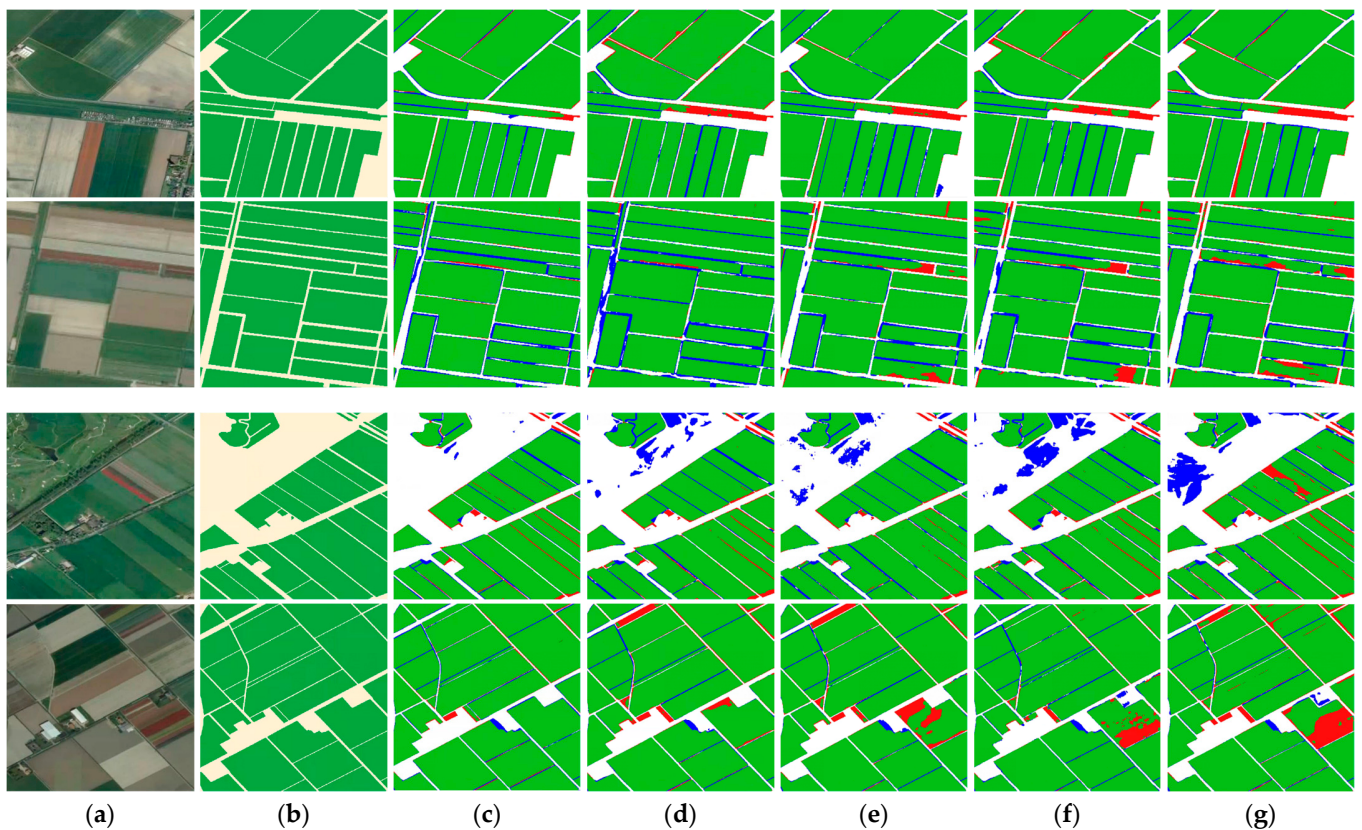


Figure 9. Examples of segmentation fields with different methods. (a) google imagery; (b) ground truth; (c) MLGNet; (d) ResUNet-a; (e) BsiNet; (f) DLinkNet. (g) ResUNet. The green color in the figure represents true positives, white represents true negatives, blue represents false positives, and red represents false negatives.

4. Discussion

4.1. Module Convergence Analysis

To examine the impact of each module on the convergence of the network, we evaluated the variation of evaluation metrics with epoch on the validation set. Figure 10 presents the statistical results for different modules. The baseline indicates the statistical results of the network without the attention fusion module and multi-task learning. Note that, in this case, the objective function only consists of cross-entropy and jaccard losses. *wo/MLS* represents the scenario without multi-task learning scheme, where two attention fusion modules are added to the network. *w/MLS* represents the complete network structure with multi-task learning scheme. Furthermore, it can be noted that as the number of epochs increases, there is a gradual improvement in the IoU and F1 score on the validation set. When the number of epochs reaches approximately 100, the metrics stabilize, indicating the network's convergence. It can be inferred from Figure 10 that the baseline undergoes oscillations in the early phases of training. The frequency of oscillations in *wo/MLS* is noticeably lower, and the indicator values are higher than those of the baseline, suggesting that the proposed network architecture can speed up the convergence rate of the network and enhance its stability. This is primarily achieved through the guided attention module, which employs a form of deep supervision to assist in learning better representations for each branch, thereby boosting the model's performance. Taking a comprehensive view, MLS achieves the fastest convergence and highest accuracy on the validation set. The multi-task learning can enhance model performance by exploiting the similarities between different tasks. This implies that the same feature extractor can be utilized to process various tasks, thereby improving the model's generalization capability.

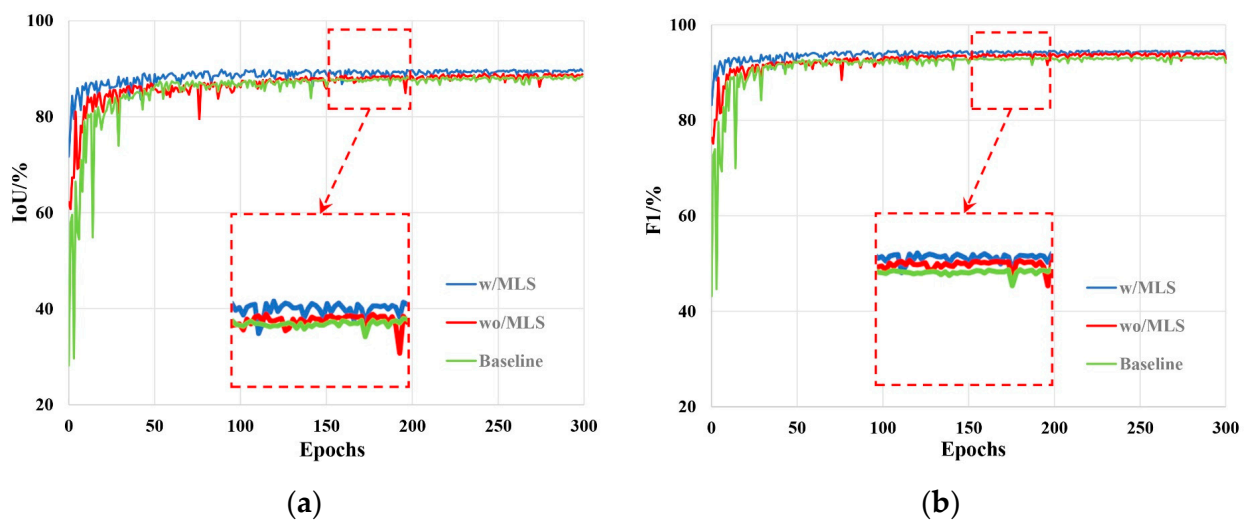


Figure 10. Convergence analysis of each module on the validation set. The red box represents a partially enlarged map. (a) IoU; (b) F1 score.

4.2. Component Effectiveness Analysis

To better evaluate the performance of each component, we conducted extensive analyses on each component in three areas. The DLinkNet architecture was used as the baseline, which is a classic encoder-decoder network, and the results are shown in Table 3. When introducing two fusion modules (i.e., ACFM and AGFM), IoU and Qua showed significant improvements, in which AGFM contributed more to the accuracy improvement, the IoU of the three areas increased by approximately 2.5%, 0.8% and 1.9%, while the corresponding Qua improved by approximately 4.1%, 2.1%, and 2.7%, respectively. This is mainly because AGFM can learn complementary details through a progressive learning approach, thereby guiding the network to enhance regions of the targets. The MLS module also proved helpful in enhancing the network's performance. More specifically,

this approach facilitated the network in learning more generalized representations. In addition, the implementation of the ReCA resulted in a significant improvement in the Qua metric. However, the overall improvement in the IoU metric was not very apparent. This is primarily because ReCA is an edge connectivity method that repairs broken lines, thus having a weak effect on the IoU of complete fields, which is mainly determined by segmentation accuracy. The experimental results confirm that the proposed modules have been helpful in improving accuracy.

Table 3. Accuracy evaluation with different components.

Cities	Components				Metrics	
	ACFM	AGFM	MLS	ReCA	IoU/%	Qua/%
Paulowna	×	×	×	×	86.26	52.03
	✓	×	×	×	87.14	54.21
	✓	✓	×	×	89.62	58.35
	✓	✓	✓	×	91.27	60.09
	✓	✓	✓	✓	92.13	65.97
Wieringermeer	×	×	×	×	91.29	50.19
	✓	×	×	×	91.52	50.24
	✓	✓	×	×	92.36	52.37
	✓	✓	✓	×	93.05	53.49
	✓	✓	✓	✓	93.71	59.83
Niedorp	×	×	×	×	87.05	50.59
	✓	×	×	×	87.14	50.81
	✓	✓	×	×	89.08	53.47
	✓	✓	✓	×	89.76	55.64
	✓	✓	✓	✓	90.35	61.00

4.3. Uncertainty in Dividing Fields

Although most of the divided fields are consistent with the ground truth, there are still some uncertain factors that can affect the final results. The first row in Figure 11 highlights the striking similarity in texture features between these elongated fields and the vegetation near the river, and the scarcity of samples makes it difficult for the algorithm to accurately distinguish between them, resulting in a lack of completeness in the segmentation results. The second row in Figure 11 presents another situation where a false edge is detected by the network, leading to the division of a whole field into two separate sub-regions. Although these erroneous edges are removed using a length threshold, their length exceeds the fixed threshold, resulting in them being considered reliable. However, this situation is not common. From the third row of Figure 11, it can be observed that there is an instance of omission where certain edges are not included in the segmentation results, where a blurry edge is visible within the red box in the segmentation map. However, due to a more confident threshold applied during skeleton extraction from the buffered edge, it fails to be detected. While a more confident threshold can reduce noise edges, it may also remove some normal results. From the fourth row in Figure 11, a deviation in the direction angle is shown. A slight tilt can be observed in the extended edge line when compared to the actual image. The main reason for this is that the initial skeleton line has a small length, resulting in high uncertainty in its direction. Overall, the proposed method satisfactorily handles the majority of cases and successfully detects many sub-regions that were not manually labeled. Therefore, our method exhibits significant potential and application value in agricultural fields extraction.

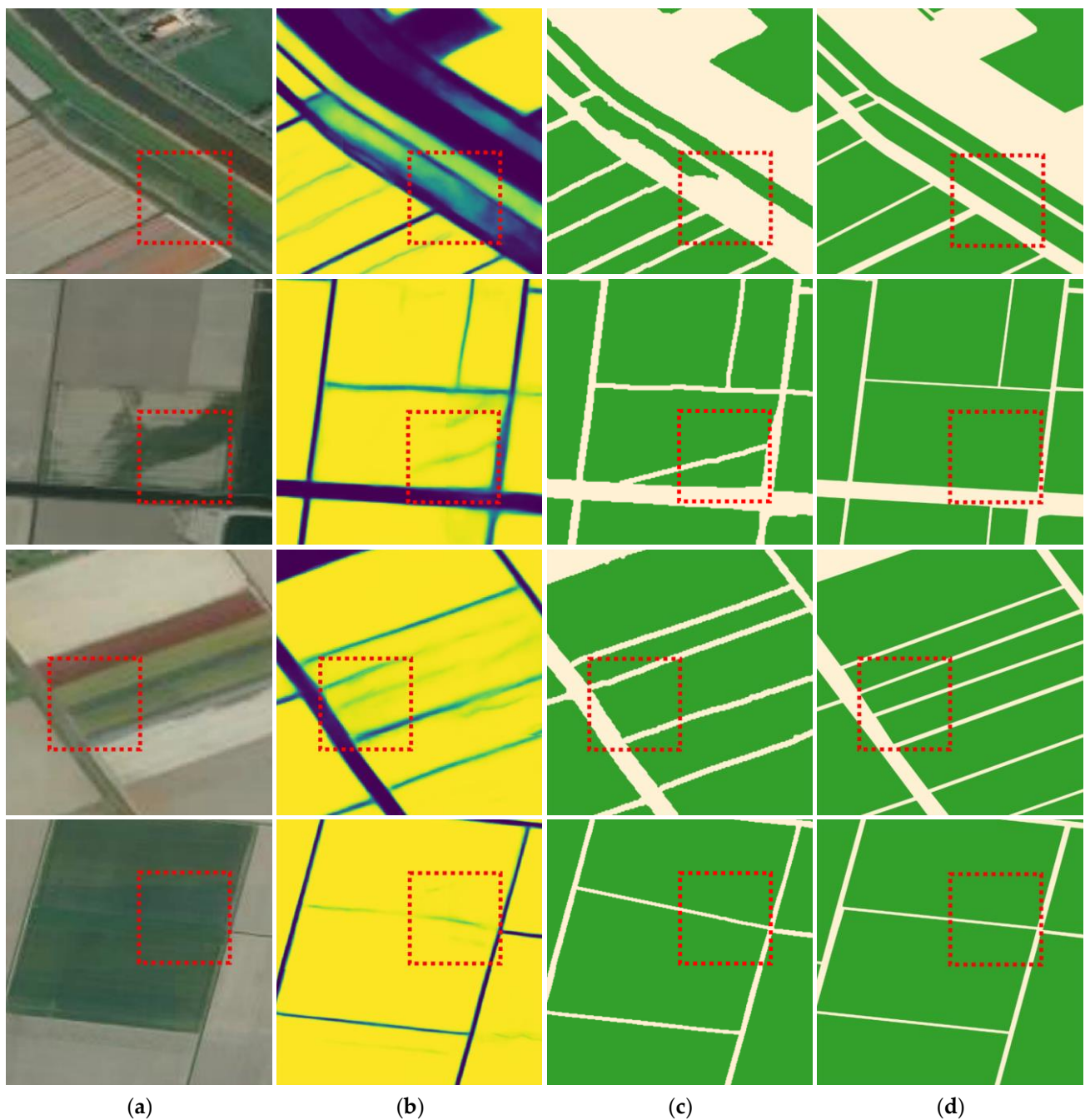


Figure 11. Visualization of uncertainty in agricultural fields. (a–d) represent google imagery, segmentation probability, extracted fields and ground truth. The regions of interest are highlighted by the red box.

5. Conclusions

This study proposes a method for segmenting agricultural fields, in which MLGNet shows significant advantages in segmentation and edge detection in terms of convergence and accuracy. From the segmentation results, it can be seen that the noises in non-edge areas can be effectively suppressed, and edge details are more prominent. These performance improvements are primarily attributed to the multi-scale attention fusion module and the multi-task learning scheme. When segmenting fields, we fully leverage the advantages of integrating edge detection and segmentation tasks, which allows us to successfully divide merged fields into multiple sub-fields. A comparison with the initial segmentation results shows that our results that are closer to ground truth in multiple indicators. Fur-

thermore, the proposed method is capable of delineating potential fields that were not manually labeled, demonstrating the value and significance of this research. Although the proposed method successfully extracted most of the sub-fields, there are still instances of erroneous results.

In the future, the related work will continue to be explored from the following two aspects: (1) The learning-based approach of edge direction will be prioritized to achieve edge connectivity. (2) Time series data will be introduced for crop identification, which can further enhance the practicality and application value of this work.

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

Funding: This work was supported by the National Natural Science Foundation of China (No. 41805090), the Natural Science Foundation of Henan Province (No. 222300420539).

Data Availability Statement: Not applicable.

Acknowledgments: We are very grateful for the valuable opinions and suggestions provided by the editors and reviewers.

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

References

1. Debats, S.R.; Luo, D.; Estes, L.D.; Fuchs, T.J.; Caylor, K.K. A generalized computer vision approach to mapping crop fields in heterogeneous agricultural landscapes. *Remote Sens. Environ.* **2016**, *179*, 210–221. [[CrossRef](#)]
2. Yli-Heikkilä, M.; Wittke, S.; Luotamo, M.; Puttonen, E.; Sulkava, M.; Pellikka, P.; Heiskanen, J.; Klami, A. Scalable Crop Yield Prediction with Sentinel-2 Time Series and Temporal Convolutional Network. *Remote Sens.* **2022**, *14*, 4193. [[CrossRef](#)]
3. Adeyemi, O.; Grove, I.; Peets, S.; Norton, T. Advanced monitoring and management systems for improving sustainability in precision irrigation. *Sustainability* **2017**, *9*, 353. [[CrossRef](#)]
4. Masoud, K.M.; Persello, C.; Tolpekin, V.A. Delineation of agricultural field boundaries from Sentinel-2 images using a novel super-resolution contour detector based on fully convolutional networks. *Remote Sens.* **2019**, *12*, 59. [[CrossRef](#)]
5. Basnyat, P.; McConkey, B.; Meinert, B.; Gatzke, C.; Noble, G. Agriculture field characterization using aerial photograph and satellite imagery. *IEEE Geosci. Remote Sens. Lett.* **2004**, *1*, 7–10. [[CrossRef](#)]
6. Wagner, M.P.; Oppelt, N. Extracting agricultural fields from remote sensing imagery using graph-based growing contours. *Remote Sens.* **2020**, *12*, 1205. [[CrossRef](#)]
7. Persello, C.; Tolpekin, V.; Bergado, J.R.; De By, R. Delineation of agricultural fields in smallholder farms from satellite images using fully convolutional networks and combinatorial grouping. *Remote Sens. Environ.* **2019**, *231*, 111253. [[CrossRef](#)] [[PubMed](#)]
8. Cheng, T.; Ji, X.; Yang, G.; Zheng, H.; Ma, J.; Yao, X.; Zhu, Y.; Cao, W. DESTIN: A new method for delineating the boundaries of crop fields by fusing spatial and temporal information from World View and Planet satellite imagery. *Comput. Electron. Agric.* **2020**, *178*, 105787. [[CrossRef](#)]
9. Hong, R.; Park, J.; Jang, S.; Shin, H.; Kim, H.; Song, I. Development of a parcel-level land boundary extraction algorithm for aerial imagery of regularly arranged agricultural areas. *Remote Sens.* **2021**, *13*, 1167. [[CrossRef](#)]
10. Wang, M.; Wang, J.; Cui, Y.; Liu, J.; Chen, L. Agricultural Field Boundary Delineation with Satellite Image Segmentation for High-Resolution Crop Mapping: A Case Study of Rice Paddy. *Agronomy* **2022**, *12*, 2342.
11. Turker, M.; Kok, E.H. Field-based sub-boundary extraction from remote sensing imagery using perceptual grouping. *ISPRS J. Photogramm. Remote Sens.* **2013**, *79*, 106–121. [[CrossRef](#)]
12. Yan, L.; Roy, D. Automated crop field extraction from multi-temporal Web Enabled Landsat Data. *Remote Sens. Environ.* **2014**, *144*, 42–64. [[CrossRef](#)]
13. Graesser, J.; Ramankutty, N. Detection of cropland field parcels from Landsat imagery. *Remote Sens. Environ.* **2017**, *201*, 165–180. [[CrossRef](#)]
14. Garcia-Pedrero, A.; Gonzalo-Martin, C.; Lillo-Saavedra, M. A machine learning approach for agricultural parcel delineation through agglomerative segmentation. *Int. J. Remote Sens.* **2017**, *38*, 1809–1819. [[CrossRef](#)]
15. Su, T.; Li, H.; Zhang, S.; Li, Y. Image segmentation using mean shift for extracting croplands from high-resolution remote sensing imagery. *Remote Sens. Lett.* **2015**, *6*, 952–961. [[CrossRef](#)]
16. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.

17. Luo, W.; Zhang, C.; Li, Y.; Yang, F.; Zhang, D.; Hong, Z. Deeply-supervised pseudo learning with small class-imbalanced samples for hyperspectral image classification. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102949. [[CrossRef](#)]
18. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *162*, 94–114. [[CrossRef](#)]
19. Wei, S.; Ji, S.; Lu, M. Toward automatic building footprint delineation from aerial images using CNN and regularization. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2178–2189. [[CrossRef](#)]
20. Waldner, F.; Diakogiannis, F.I. Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network. *Remote Sens. Environ.* **2020**, *245*, 111741. [[CrossRef](#)]
21. Long, J.; Li, M.; Wang, X.; Stein, A. Delineation of agricultural fields using multi-task BsiNet from high-resolution satellite images. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102871. [[CrossRef](#)]
22. Jong, M.; Guan, K.; Wang, S.; Huang, Y.; Peng, B. Improving field boundary delineation in ResUNets via adversarial deep learning. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102877. [[CrossRef](#)]
23. Pan, S.; Tao, Y.; Chen, X.; Chong, Y. Progressive Guidance Edge Perception Network for Semantic Segmentation of Remote-Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [[CrossRef](#)]
24. Li, M.; Long, J.; Stein, A.; Wang, X. Using a semantic edge-aware multi-task neural network to delineate agricultural parcels from remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2023**, *200*, 24–40. [[CrossRef](#)]
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
26. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 182–186.
27. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
28. Chen, S.; Tan, X.; Wang, B.; Hu, X. Reverse attention for salient object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 234–250.
29. Sun, D.; Yao, A.; Zhou, A.; Zhao, H. Deeply-supervised knowledge synergy. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6997–7006.
30. Zhen, M.; Wang, J.; Zhou, L.; Li, S.; Shen, T.; Shang, J.; Fang, T.; Quan, L. Joint semantic segmentation and boundary detection using iterative pyramid contexts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13666–13675.
31. Berman, M.; Triki, A.R.; Blaschko, M.B. The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4413–4421.
32. Zhang, Y.; Yang, Q. A survey on multi-task learning. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5586–5609. [[CrossRef](#)]
33. Xu, L.; Yang, P.; Yu, J.; Peng, F.; Xu, J.; Song, S.; Wu, Y. Extraction of cropland field parcels with high resolution remote sensing using multi-task learning. *Eur. J. Remote Sens.* **2023**, *56*, 2181874. [[CrossRef](#)]
34. Wang, Y.; Gu, L.; Jiang, T.; Gao, F. MDE-UNet: A Multitask Deformable UNet Combined Enhancement Network for Farmland Boundary Segmentation. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [[CrossRef](#)]
35. Wang, Z.; Acuna, D.; Ling, H.; Kar, A.; Fidler, S. Object instance annotation with deep extreme level set evolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7500–7508.
36. Kim, Y.; Kim, S.; Kim, T.; Kim, C. CNN-based semantic segmentation using level set loss. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019; pp. 1752–1760.
37. Wang, X.; Shrivastava, A.; Gupta, A. A-fast-rcnn: Hard positive generation via adversary for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2606–2615.
38. Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7482–7491.
39. Turker, M.; Koc-San, D. Building extraction from high-resolution optical spaceborne images using the integration of support vector machine (SVM) classification, Hough transformation and perceptual grouping. *Int. J. Appl. Earth Obs. Geoinf.* **2015**, *34*, 58–69. [[CrossRef](#)]
40. Meyer, F.; Beucher, S. Morphological segmentation. *J. Vis. Commun. Image Represent.* **1990**, *1*, 21–46. [[CrossRef](#)]
41. Suzuki, S. Topological structural analysis of digitized binary images by border following. *Comput. Vis. Graph. Image Process.* **1985**, *30*, 32–46. [[CrossRef](#)]
42. Wiedemann, C.; Heipke, C.; Mayer, H.; Jamet, O. Empirical evaluation of automatically extracted road axes. *Int. J. Comput. Vis.* **1998**, *12*, 172–187.
43. Zhang, Z.; Liu, Q.; Wang, Y. Road extraction by deep residual u-net. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.