Journal of Real-Time Image Processing (2024) 21:101
https://doi.org/10.1007/s11554-024-01474-0

RESEARCH

ResLMFFNet: a real-time semantic segmentation network for precision agriculture

Irem Ulku[1]

Received: 7 January 2024 / Accepted: 7 May 2024
© The Author(s) 2024

Abstract
Lightweight multiscale-feature-fusion network (LMFFNet), a proficient real-time CNN architecture, adeptly achieves a balance between inference time and accuracy. Capturing the intricate details of precision agriculture target objects in remote sensing images requires deep SEM-B blocks in the LMFFNet model design. However, employing numerous SEM-B units leads to instability during backward gradient flow. This work proposes the novel residual-LMFFNet (ResLMFFNet) model for ensuring smooth gradient flow within SEM-B blocks. By incorporating residual connections, ResLMFFNet achieves improved accuracy without affecting the inference speed and the number of trainable parameters. The results of the experiments demonstrate that this architecture has achieved superior performance compared to other real-time architectures across diverse precision agriculture applications involving UAV and satellite images. Compared to LMFFNet, the ResLMFFNet architecture enhances the Jaccard Index values by 2.1% for tree detection, 1.4% for crop detection, and 11.2% for wheat-yellow rust detection. Achieving these remarkable accuracy levels involves maintaining almost identical inference time and computational complexity as the LMFFNet model. The source code is available on GitHub: https://github.com/iremulku/Semantic-Segmentation-in-Precision-Agriculture.

Keywords Real-time semantic segmentation · Remote sensing · Precision agriculture

1 Introduction

Precision agriculture is a technique that aims to increase crop productivity while reducing costs and environmental impact [1]. Sensing technology is a tool for achieving this goal by monitoring vast lands. With the advancement of convolutional neural networks (CNNs), this technology has become even more powerful [2]. CNN models are used in early disease detection, leading to reduced yield losses by applying fungicides at the right time [1]. Additionally, CNN architectures can identify trees and crops to maximize agricultural efficiency [3]. However, CNN architectures [4] have high inference time measured in frames per second (fps), which makes them impractical for real-time applications.

Precision agriculture faces the challenge of balancing high accuracy with fast inference speed. Existing research on using CNN models for real-time precision agriculture focuses on a single specialized application [5–7] and does not provide sufficient accuracy [8–10]. Therefore, it is essential to adapt recent real-time models to provide high accuracy in various precision agriculture applications [1].

Real-time CNN architectures generally adhere to an encoder–decoder framework. Architectures like SegNet [11] employ encoders based on established backbone networks. In contrast, ENet [12], LEDNet [13], and FSFNet [14] use lightweight modules to build efficient encoders, resulting in fewer parameters. These models, however, lack accuracy compared to others.

The decoder parts of real-time semantic segmentation models may also have different designs. SegNet and ESNet [15] have symmetrically designed decoders. In contrast, DFANet [16], and FASSD-Net [17] architectures have adopted asymmetric decoder structures to enhance inference speed. Recent transformer-based models such as UNetFormer [18] achieve good performance without sacrificing real-time speed.

Remarkably, by introducing a split-extract-merge bottleneck (SEM-B) in its backbone network, the real-time LMFFNet [19] architecture achieves high accuracy with

✉ Irem Ulku
irem.ulku@ankara.edu.tr

[1] Department of Computer Engineering, Ankara University, 06830 Ankara, Turkey

Published online: 28 May 2024

fewer model parameters. A lightweight asymmetric decoder is used in the LMFFNet model to process multi-scale features, which improves inference time. However, with the challenging low latency and high accuracy requirements for various precision agriculture tasks, LMFFNet still needs improvement.

Realizing precision agriculture practices with high accuracy in real-time is challenging. In real-world remote-sensing images with high spatial resolution, capturing the intricate details of precision agriculture target objects poses considerable difficulties. This paper proposes the ResLMFFNet architecture to increase prediction accuracy and achieve a decent trade-off between high accuracy and fast inference speed. ResLMFFNet introduces the following novelties:

- LMFFNet is the base model since it already achieves an adequate trade-off between accuracy and efficiency. However, residual connections are added to the SEM-B blocks in this study to further increase accuracy without affecting inference speed (Fig. 1). By preserving low-level features lost through deep SEM-B blocks, these connections further enhance the performance of LMFFNet. Residual connections are preferred to dense or attention connections since the element-wise summation operation does not introduce trainable weights.
- Before upsampling, the dropout layer is used in the decoder, which allows the model to show higher generalization ability, making it better suited to a wide range of precision agriculture practices.

In the remainder of this paper, the details of the proposed architecture ResLMFFNet are described in Sect. 2. Section 3 presents the experimental results. Conclusions are given in Sect. 4.

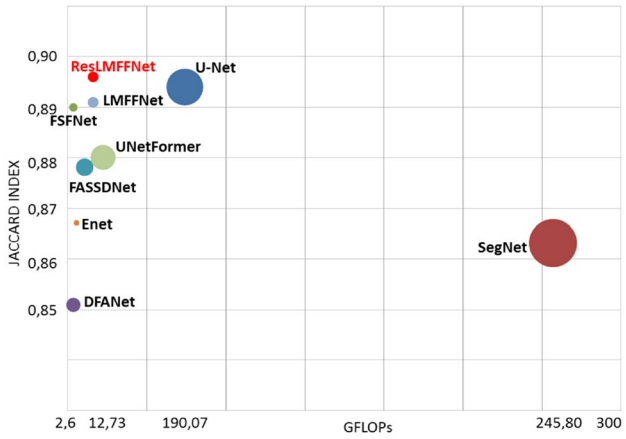


Fig. 1 Complexity-accuracy trade-off comparison on the DSTL image set in terms of Jaccard Index JI, Giga floating point operations (GFLOPs), and model parameters. The circle size indicates the number of the model parameters

2 Methods

The ResLMFFNET model, an improved version of the LMFFNET architecture, emerges to increase accuracy while preserving real-time capabilities, as depicted in Fig. 2. Similar to the LMFFNET design, the ResLMFFNET model is composed of three core components: SEM-B block, feature fusion module (FFM), and multiscale attention decoder (MAD).

The ResLMFFNET architecture achieves its novel contribution by implementing residual connections within the SEM-B blocks, as illustrated in Fig. 2. Furthermore, the accuracy is further boosted by the inclusion of a dropout layer in the decoder design. This section provides a detailed explanation of the essential components within the ResLMFFNET architecture.

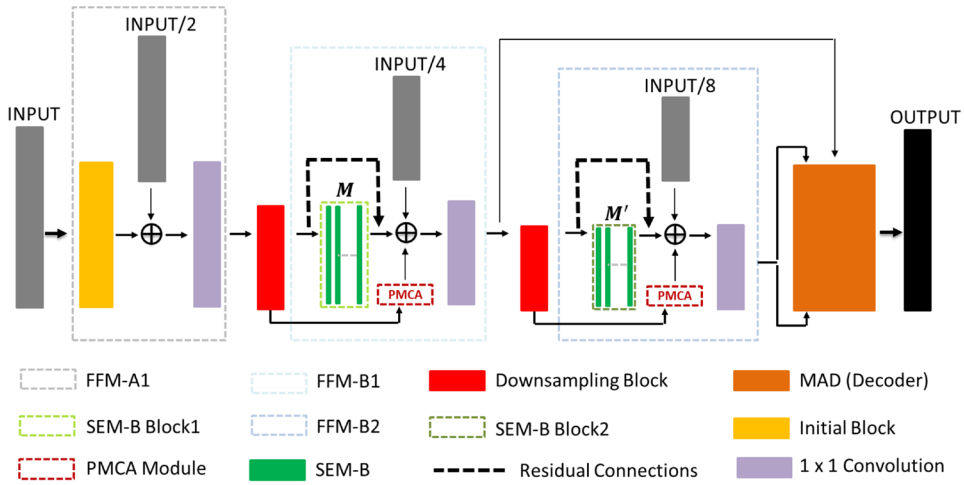


Fig. 2 ResLMFFNet architecture

2.1 SEM-B block

The SEM-B block is built upon the split-extract-merge bottleneck shown in Fig. 3. SEM-B applies a 3×3 convolution, then splits the feature map into two branches, each with 1/4 channels of the input. One branch undergoes depthwise convolution, while the other employs depthwise dilated convolution so that SEM-B effectively captures fine spatial details and larger contextual information simultaneously. Following the concatenation of the branch outputs, another 3×3 convolution is applied. This operation, leading to the original channel number, combines multi-scale features more cohesively. Finally, the output feature map is added to the input, yielding a more informative representation.

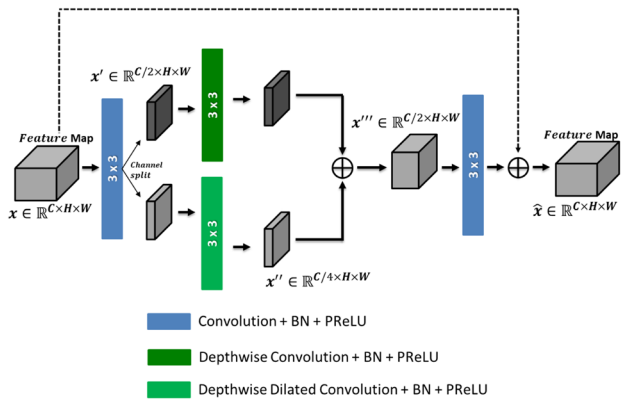


Fig. 3 SEM-B

As depicted in Fig. 2, the architecture employs a pair of distinct SEM-B blocks. The initial block is responsible for capturing shallow features, whereas the subsequent one focuses on extracting deep features. SEM-B Block1 is composed of $M(M > 0)$ SEM-Bs, while SEM-B Block2 comprises $M'\left(M' > 0\right)$ of these bottleneck units.

2.2 FFM modules

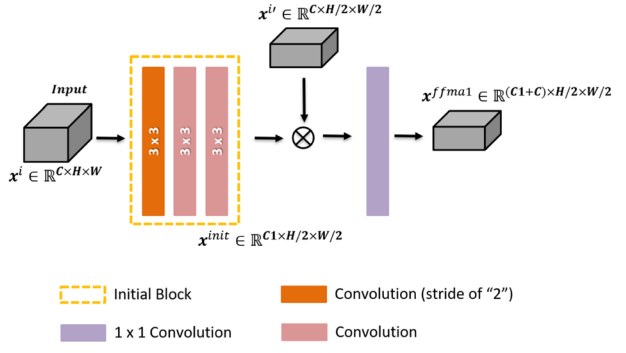
Two FFM modules, namely FFM-A and FFM-B (depicted in Fig. 4a and b, respectively), are employed to fuse multiscale features. Within these modules, pointwise convolution enables the extraction of valuable information with few parameters.

In Fig. 4a, the initial block applies a 3×3 convolution with a stride of 2, followed by two more 3×3 convolutions to the input image $x^i \in \mathbb{R}^{C\times H\times W}$. The output feature map of this initial block $x^{\text{init}} \in \mathbb{R}^{C1\times H/2\times W/2}$ is then concatenated with the downsampled feature map $x^{i'} \in \mathbb{R}^{C\times H/2\times W/2}$. The output of the FFM-A1 module $x^{\text{ffma1}} \in \mathbb{R}^{(C1+C)\times H/2\times W/2}$ is derived as follows:

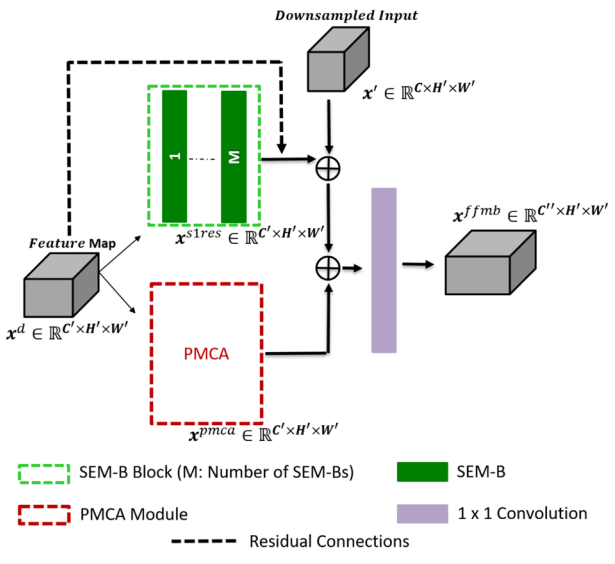
$$x^{\text{ffma1}} = f_{1\times 1\text{conv}}\left(f_{\text{concat}}\left(x^{\text{init}}, x^{i'}\right)\right), \tag{1}$$

where $f_{1\times 1\text{conv}}$ represents the pointwise convolution operation and f_{concat} denotes the concatenation operation.

The downsampling block in Fig. 5 is applied on the output of the FFM-A1 block, concatenating feature maps of 3×3 convolution (with a stride of 2) and 2×2 max pooling operations to retain more spatial information. As shown in Fig. 4b, the resulting output, $x^d \in \mathbb{R}^{C2\times H/4\times W/4}$, serves as input for



(a) FFM-A



(b) FFM-B

Fig. 4 Feature fusion modules

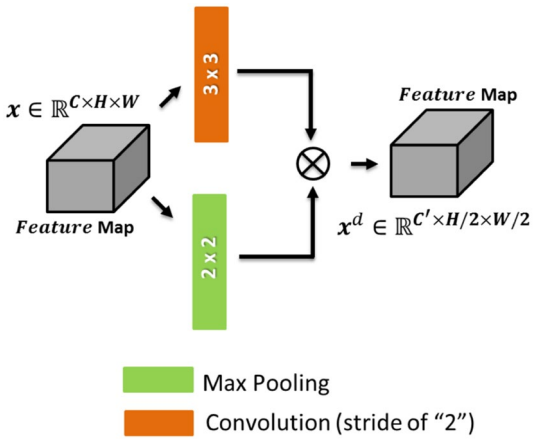


Fig. 5 Downsampling block

both the SEM-B block with M number of SEM-Bs and the partition-merge channel attention (PMCA) module. SEM-B Block1 is applied to this feature map x^d as follows:

$$x^{s1} = f_{\text{semb1}}\left(x^d\right), \tag{2}$$

where $x^{s1} \in \mathbb{R}^{C2\times H/4\times W/4}$ is the output of SEM-B Block1 and f_{semb1} represents the SEM-B Block1 operation.

The PMCA module calculates a weighted sum by applying global average pooling to the partitioned regions, then utilizing adaptively learned neural network weights, as illustrated in Fig. 6. By integrating a squeeze-and-excitation (SE) block [20], PMCA allocates more attention to the informative features. The output feature map of this module $x^{pmca1} \in \mathbb{R}^{C2\times H/4\times W/4}$ is obtained as:

$$x^{\text{pmca1}} = f_{\text{pmca}}\left(x^d\right), \tag{3}$$

where f_{pmca} represents the operations in PMCA module.

2.2.1 Residual connections

Due to the intricate details embedded in high-resolution remote-sensing images, the depth of the SEM-B blocks must be large enough to capture these nuanced differences in precision agriculture objects. However, increasing the number of SEM-B units in SEM-B blocks deepens the network, creating a problem of poor gradient flow during back-propagation. This trend leads to issues related to exploiting and vanishing gradients, which reduces the model's trainability and expressiveness, thereby decreasing its performance [21].

A novel approach to the ResLMFFNet model is to incorporate a residual connection from the input feature map x^d to the output feature map x^{s1} of the SEM-B block to mitigate this problem. Fig. 7 illustrates this approach in which the input feature map is added to the output feature map of the SEM-B block by element-wise operation.

As information flows directly through the SEM-B blocks, residual connections facilitate the capture of intricate details in remote-sensing images with high spatial resolution and prevent vanishing/exploding gradients [22]. Moreover, matrix addition in residual connections does not add learnable parameters. Thus, ResLMFFNet uses residual connections rather than dense connections or attention mechanisms. Referencing Fig. 4b, the output feature map

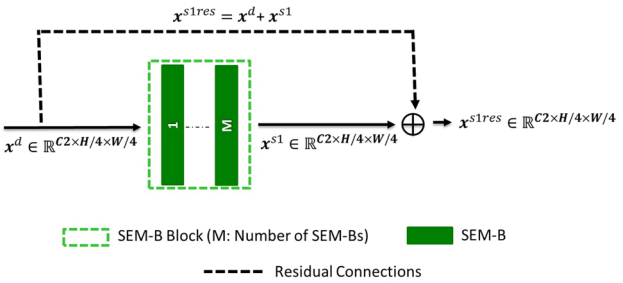
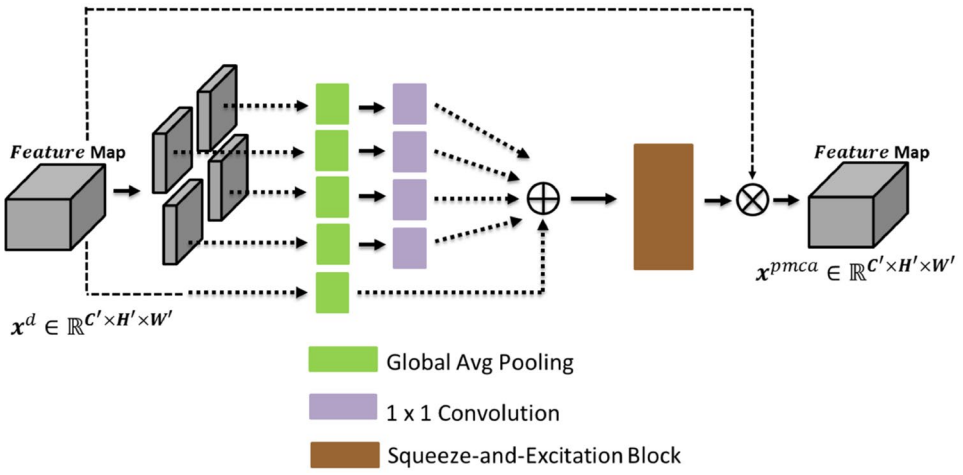


Fig. 7 Residual connection in SEM-B block

Fig. 6 PMCA module



x^{s1} from SEM-B Block1 is updated using a residual connection to obtain $x^{s1res} \in \mathbb{R}^{C2\times H/4\times W/4}$:

$$x^{s1res} = x^{s1} + x^{d}. \tag{4}$$

In the LMFFNet architecture, the output of the PMCA module x^{pmca1}, the downsampled input $x^{i''} \in \mathbb{R}^{C\times H/4\times W/4}$, and the output of SEM-B Block1 x^{s1} are concatenated. In ResLMFFNet, this concatenation includes x^{s1res} instead of x^{s1}. Using pointwise convolution, the FFM-B1 block produces the output $x^{ffmb1} \in \mathbb{R}^{(C3+C)\times H/4\times W/4}$ as follows:

$$x^{ffmb1} = f_{1\times 1conv}\left(f_{concat}\left(x^{s1res}, x^{pmca1}, x^{i''}\right)\right). \tag{5}$$

Two FFM-B blocks at different levels are utilized to fuse shallow and abstract features. Using a residual connection allows for deeper network with unchanged trainable parameters, especially beneficial for preserving important features in objects of different scales. This connection involves a simple element-wise summation, avoiding parameter increase and causing only a slight inference speed rise.

2.3 MAD decoder

The attention-based MAD decoder architecture is presented in Fig. 8, designed to recover multi-scale spatial details. The output $x^{ffmb1} \in \mathbb{R}^{(C3+C)\times H/4\times W/4}$ of the FFM-B1 block, at a quarter scale of the input, undergoes a pointwise convolution. Consequently, this process yields the output feature map $x^{ffmb1MAD} \in \mathbb{R}^{C5\times H/4\times W/4}$ with $C5$ channels as follows:

$$x^{ffmb1MAD} = f_{1\times 1conv}\left(x^{ffmb1}\right). \tag{6}$$

The output $x^{ffmb2} \in \mathbb{R}^{(C4+C)\times H/8\times W/8}$ from the FFM-B2 block, which is at 1/8 scale of the input, undergoes pointwise convolution, reaching to $C6$ number of channels. Moreover, this feature map is doubled in size using upsampling, leading to $x^{ffmb2MAD} \in \mathbb{R}^{C6\times H/4\times W/4}$ as:

$$x^{ffmb2MAD} = f_{up}\left(f_{1\times 1conv}\left(x^{ffmb2}\right)\right), \tag{7}$$

where f_{up} represents the upsampling operation performed with bilinear interpolation. To capture more multi-scale spatial information, the feature maps $x^{ffmb1MAD}$ and $x^{ffmb2MAD}$ are concatenated and subjected to a 3×3 depthwise separable convolution. This process refines the combined multi-scale information effectively. The resulting feature map is then passed through a sigmoid activation function to produce the multi-scale attention map $M^{MAM} \in \mathbb{R}^{C\times H/4\times W/4}$ as follows:

$$\begin{aligned} M^{MAM'} &= \left(f_{concat}\left(x^{ffmb1MAD}, x^{ffmb2MAD}\right)\right. \\ M^{MAM} &= \delta\left(f_{dwconv}\left(M^{MAM'}\right)\right), \end{aligned} \tag{8}$$

where f_{dwconv} represents depthwise separable convolution operation and δ shows the sigmoid activation function.

2.3.1 Dropout

A dropout layer is incorporated into the decoder part of the ResLMFFNet architecture to enhance its generalization capability. The FFM-B2 block's output $x^{ffmb2} \in \mathbb{R}^{(C4+C)\times H/8\times W/8}$ is reused in a second branch beyond its role in creating the M^{MAM} attention map. While the original LMFFNet design applies 3×3

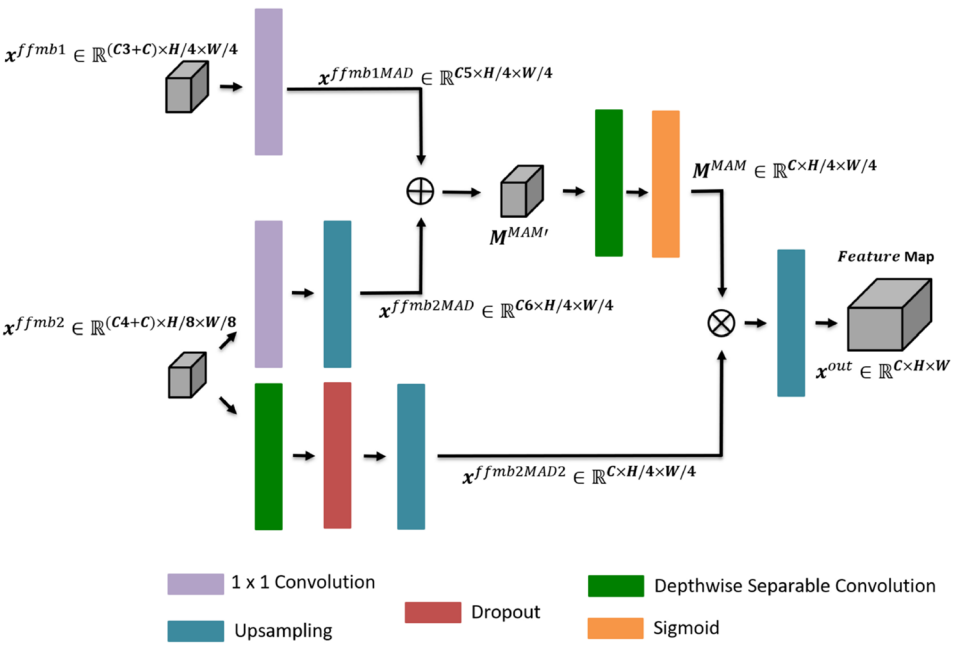


Fig. 8 Decoder of ResLMFFNet—MAD

depthwise separable convolution and upsampling to this feature map x^{ffmb2}, the ResLMFFNet design (as depicted in Fig. 8) employs a dropout layer with rate of 0.5 immediately after a 3×3 depthwise separable convolution, followed by upsampling. This process yields the $x^{\text{ffmb2MAD2}} \in \mathbb{R}^{C\times H/4\times W/4}$ feature map as:

$$x^{\text{ffmb2MAD2}} = f_{\text{up}}\left(f_{\text{drop}}\left(f_{\text{dwconv}}\left(x^{\text{ffmb2}}\right)\right)\right), \tag{9}$$

where f_{drop} represents the dropout layer of 0.5 rate. The ResLMFFNet architecture fuses the attention map M^{MAM} from the first branch and the feature map $x^{\text{ffmb2MAD2}}$ from the second branch using pointwise multiplication. The output $x^{\text{out}} \in \mathbb{R}^{C\times H\times W}$ is acquired through upsampling after the pointwise multiplication to reach the original input size as:

$$x^{\text{out}} = f_{\text{up}}\left(M^{\text{MAM}} \odot x^{\text{ffmb2MAD2}}\right), \tag{10}$$

where $\odot$ is the pointwise multiplication operation.

3 Experimental results

This section introduces the image sets, the evaluation metrics, and the implementation details. Subsequently, comprehensive experiments assess the real-time semantic segmentation performance of the ResLMFFNet architecture across various precision agriculture applications.

3.1 Image sets

The experiments employ three remote-sensing image sets. One set comprises images obtained from satellite-based systems, while the other two consist of images acquired through UAV sensing systems. This section explains each of these image sets.

3.1.1 DSTL satellite imagery feature detection image set

The DSTL Kaggle [2] image set comprises 25 satellite images, each capturing a region of 1000 m × 1000 m. An example image is presented in Fig. 9a, accompanied by the corresponding ground truth displayed in Fig. 9b for ten labeled classes. This study uses images with a spatial resolution of 1.24 m as input for real-time binary semantic segmentation of crop regions. The depicted light green pixels in Fig. 9b represent crops. Ground truth annotations are created by describing the target classes with polygons in GeoJSON, followed by normalizing geo-coordinates within specific ranges to obscure satellite image locations.

3.1.2 RIT-18 (The Hamlin State Beach Park) aerial image set

The RIT-18 [23] image set includes aerial images taken via an octocopter. The training image (Fig. 9c) has a 9393 × 5642 pixel size with a high spatial resolution (0.047 m). This study uses the RIT-18 image set for real-time binary semantic segmentation of trees. Ground truth (Fig. 9d) for eighteen labeled classes shows tree pixels in blue. Ground truth annotations are created by manually delineating the target classes within each orthomosaic image utilizing ENVI software.

3.1.3 Wheat Yellow-Rust aerial image set

The Wheat Yellow-Rust [24] image set is a collection of aerial images captured by the DJI Matrice 100 (M100) quadcopter. The training image indicated in Fig. 9e possesses dimensions of 1336 × 2991 pixels and a spatial resolution of 0.013 ms. This study performs real-time binary semantic segmentation of wheat yellow-rust disease. Affected regions, caused by the controlled introduction of yellow rust inoculum in 2 m × 2 m regions, are highlighted in blue within the ground truth representation in Fig. 9f. Ground truth

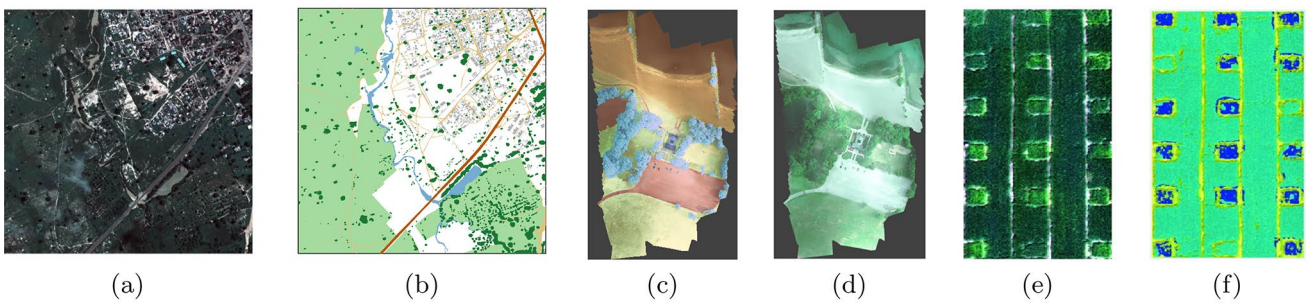


Fig. 9 Image set Illustrations. **a** An example original image from the DSTL image set. **b** The corresponding ground truth image from the DSTL image set. **c** Original training image from the RIT-18 image set. **d** The corresponding ground truth image from the RIT-18 image set. **e** Original training image from the Wheat Yellow Rust image set. **f** The corresponding ground truth image from the Wheat Yellow Rust image set

annotations are created by labeling target objects in each image using the MATLAB ImageLabeler tool.

3.2 Evaluation metric

The Jaccard Index, also called the intersection over union (IoU), is a metric utilized in experiments to evaluate the performance of real-time semantic segmentation models. A binary classification task involves calculating overlapping pixels of the prediction and the mask divided by the total number of pixels, as follows:

$$\text{Jaccard Index} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{11}$$

where TP denotes correctly predicted pixels, FP represents incorrectly predicted pixels, and FN corresponds to missed pixels in the prediction.

Additionally, F_1 score is used as a complementary metric for showing the performance of the proposed model. F_1 score combines precision and recall by calculating harmonic series as:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{12}$$

where the precision and the recall are calculated as follows:

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{13}$$

3.2.1 Implementation details

The experiments involve training semantic segmentation architectures with the adaptive moment estimation (Adam) algorithm on the NVIDIA Quadro RTX 5000 GPU while utilizing the PyTorch framework. A manual hyperparameter tuning process is adapted separately for each image set to find the best-performing values based on Jaccard Index measurements.

The mini-batch size is 8, and the number of epochs is 70. Weight initialization follows the Xavier uniform method, while the chosen loss function is binary cross-entropy with logits. For the DSTL and RIT-18 image sets, an initial learning rate of 10^{-4} is adopted and decreased by 9% every five iterations. The Wheat Yellow-rust image set employs an initial learning rate of 5×10^{-5}, which undergoes a reduction of 9% every ten iterations.

The images are partitioned into 224×224 image patches, resulting in 5985 patches from the DSTL set, 1778 patches from the RIT-18 set, and 1299 patches from the Wheat Yellow-rust set. These patches are then assigned to training (72%), testing (20%), and validation (8%). The validation process utilizes 5-fold cross-validation.

The experiments are conducted using RGB and normalized difference vegetation index (NDVI) [24] images to demonstrate the generalization capacity of ResLMFFNET architecture. By normalizing the difference between near infrared and red reflectance, NDVI provides information on healthy green plants.

3.3 Results

Table 1 outlines a comparative analysis between the ResLMFFNET model and state-of-the-art real-time semantic segmentation architectures using the RIT-18, DSTL and Wheat Yellow-Rust image sets. The comparison examines inference speed, computational complexity and memory requirement. Inference speed is measured using frames per

Table 1 Performance comparison of real-time semantic segmentation architectures estimated on the RIT-18, DSTL, and Wheat Yellow-Rust image sets

Architectures	GFLOPs	Params (M)	GPU memory requirement for inference (MB)	RIT-18 image set	DSTL image set	Wheat Yellow-Rust image set
				FPS	FPS	FPS
U-Net	190.07	14.79	64.09	20.54	20.69	20.48
SegNet	245.80	29.44	120.74	18.37	18.48	18.44
FSFNet	2.60	0.81	9.27	135.59	141.19	140.04
DFANet	2.73	2.18	14.79	50.79	46.01	48.12
FASSDNet	8.65	2.84	17.15	98.16	106.67	97.56
Enet	4.35	0.36	7.68	60.21	61.95	60.61
UNetFormer	17.95	11.71	51.95	106.66	106.31	104.92
LMFFNet	12.73	1.34	11.41	99.38	109.09	99,84
ResLMFFNet (proposed)	**12.73**	**1.34**	**18.01**	**98.91**	**108.55**	**97.86**

All experiments are performed with an input size of $3 \times 224 \times 224$ on a Quadro RTX 5000 GPU Card

The bold entities show the best experimental results in each metric

Table 2 Tree semantic segmentation test results in terms of Jaccard Index (IoU) and F_1 score for the different architectures with RIT-18 image set

Architectures	RGB images		NDVI images	
	IoU	F_1	IoU	F_1
U-Net	0.860 ± 0.285	0.887 ± 0.243	0.841 ± 0.306	0.878 ± 0.269
SegNet	0.852 ± 0.293	0.883 ± 0.270	0.835 ± 0.310	0.860 ± 0.294
FSFNet	0.844 ± 0.307	0.868 ± 0.290	0.772 ± 0.335	0.814 ± 0.307
DFANet	0.807 ± 0.342	0.833 ± 0.323	0.796 ± 0.322	0.825 ± 0.323
FASSDNet	0.847 ± 0.292	0.876 ± 0.269	0.756 ± 0.374	0.786 ± 0.353
Enet	0.859 ± 0.281	0.885 ± 0.258	0.801 ± 0.344	0.825 ± 0.329
UNetFormer	0.870 ± 0.260	0.899 ± 0.232	0.842 ± 0.296	0.871 ± 0.272
LMFFNet	0.874 ± 0.270	0.895 ± 0.249	0.835 ± 0.298	0.865 ± 0.274
ResLMFFNet (proposed)	**0.884 ± 0.252**	**0.913 ± 0.214**	**0.856 ± 0.278**	**0.885 ± 0.257**

The bold entities show the best experimental results in each metric

second (FPS), while computational complexity is evaluated based on metrics including learnable parameters, floating-point operations per second (FLOPs), and Gigaflops (GFLOPs). Function "torch.cuda.max_memory_allocated()" calculates the maximum GPU requirement for inference. Notably, the ResLMFFNET model retains identical GFLOPs and trainable parameter values as the LMFFNET, with only negligible variations observed in the FPS and memory requirement values.

The tree semantic segmentation test results are shown in Table 2, measured as Jaccard Index (IoU) and F_1 score. ResLMFFNET outperforms other architectures. Compared with LMFFNET, the proposed architecture enhances the Jaccard index for RGB by about 1% and NDVI by 2.1%, all while

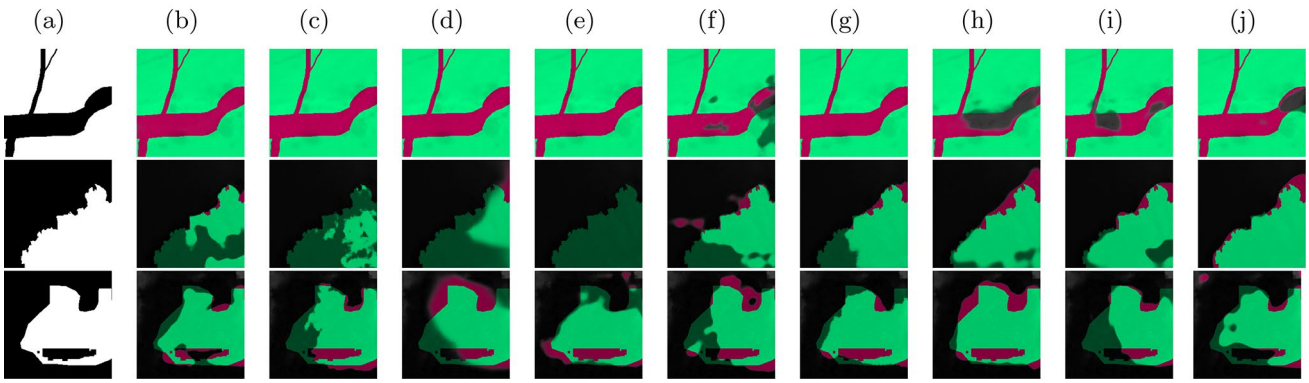


Fig. 10 Real-time semantic segmentation test results. Light green represents a hit, dark green represents a miss, and red represents a false alarm. First row shows crop predictions, second row shows tree predictions and third row shows wheat-yellow rust predictions **a** ground-truth masks. **b** U-Net. **c** SegNet. **d** FSFNet. **e** DFANet. **f** FASSDNet. **g** ENet. **h** UNetFormer. **i** LMFFNet. **j** ResLMFFNet

Table 3 Crop semantic segmentation test results in terms of Jaccard Index (IoU) and F_1 score for the different architectures with DSTL image set

Architectures	RGB images		NDVI images	
	IoU	F_1	IoU	F_1
U-Net	0.894 ± 0.237	0.904 ± 0.240	0.857 ± 0.285	0.883 ± 0.263
SegNet	0.863 ± 0.311	0.876 ± 0.302	0.874 ± 0.281	0.893 ± 0.265
FSFNet	0.890 ± 0.263	0.908 ± 0.246	0.863 ± 0.310	0.877 ± 0.300
DFANet	0.851 ± 0.324	0.864 ± 0.314	0.829 ± 0.337	0.847 ± 0.325
FASSDNet	0.878 ± 0.290	0.892 ± 0.278	0.869 ± 0.296	0.884 ± 0.284
Enet	0.867 ± 0.306	0.880 ± 0.296	0.866 ± 0.311	0.877 ± 0.304
UNetFormer	0.880 ± 0.283	0.896 ± 0.268	0.865 ± 0.307	0.879 ± 0.297
LMFFNet	0.891 ± 0.261	0.909 ± 0.242	0.874 ± 0.283	0.893 ± 0.266
ResLMFFNet (proposed)	**0.896 ± 0.256**	**0.913 ± 0.247**	**0.888 ± 0.272**	**0.904 ± 0.259**

The bold entities show the best experimental results in each metric

maintaining comparable inference speed and computational complexity.

Table 3 shows semantic segmentation test results for the crop target object within the DSTL satellite image set. ResLMFFNET outperforms LMFFNET by achieving approximately 0.5% higher Jaccard index values for RGB and 1.4% for NDVI in segmenting large-scale crop objects.

Table 4 displays semantic segmentation test results for the Wheat Yellow-Rust aerial image set. ResLMFFNET surpasses other architectures, achieving notable improvements of approximately 11.2% for RGB and 4.6% for NDVI in the Jaccard index compared to LMFFNET. This enhancement is remarkable, considering the challenging image set with limited training data. In addition, the tree class from the DSTL image set has limited labeled data. Therefore, Table 5 lists only test results for real-time models that converge on limited training samples. The ResLMFFNet model is superior to other models and achieves improvements of 2.5% in RGB images and 3.6% in NDVI images compared to the LMFFNet model.

Figure 10 illustrates the visual comparison of prediction results from different models using sample images alongside their corresponding ground truth masks. Specifically, light green represents hit pixels, dark green denotes missed pixels, and red indicates false alarm pixels. Three lines display the prediction results for crop, tree, and wheat yellow-rust objects. The ResLMFFNet architecture, illustrated in Fig. 10 (i), demonstrates reduced false alarms and miss pixels for target objects of varying scales. These visual results indicate that the ResLMFFNet architecture improves segmentation accuracy while retaining real-time inference speed.

Figure 11 shows the accuracy curves of the ResLMFFNet and LMFFNet architectures obtained through training using the RIT-18, DSTL and Wheat Yellow-Rust image sets. According to the fluctuations, the LMFFNet architecture exhibits an unstable training process, probably due to problems like vanishing/exploding gradients. Training becomes more stable with the proposed ResLMFFNet by smoothing fluctuations, as reflected in Fig. 11a, c and e. Therefore, ResLMFFNet can overcome possible vanishing/exploding gradients, thus improving overall segmentation performance.

3.3.1 Ablation study

The first ablation study investigates how the dropout layer in the MAD decoder affects performance. Table 6 reveals that ResLMFFNet and LMFFNet perform better when the dropout rate is 0.5. With a dropout rate of

Table 4 Wheat Yellow-Rust semantic segmentation test results in terms of Jaccard Index (IoU) and F_1 score for the different architectures with UAV image set

Architectures	RGB images		NDVI images	
	IoU	F_1	IoU	F_1
U-Net	0.521 ± 0.294	0.647 ± 0.333	0.502 ± 0.355	0.582 ± 0.353
SegNet	0.545 ± 0.347	0.620 ± 0.372	0.469 ± 0.366	0.545 ± 0.378
FSFNet	0.510 ± 0.309	0.613 ± 0.304	0.407 ± 0.333	0.499 ± 0.343
DFANet	0.489 ± 0.321	0.587 ± 0.320	0.448 ± 0.323	0.546 ± 0.330
FASSDNet	0.538 ± 0.318	0.633 ± 0.326	0.540 ± 0.307	0.643 ± 0.301
Enet	0.524 ± 0.327	0.617 ± 0.334	0.421 ± 0.352	0.500 ± 0.375
UNetFormer	0.664 ± 0.241	0.769 ± 0.203	0.515 ± 0.320	0.615 ± 0.313
LMFFNet	0.569 ± 0.297	0.671 ± 0.291	0.574 ± 0.282	0.682 ± 0.269
ResLMFFNet (proposed)	**0.681 ± 0.240**	**0.781 ± 0.208**	**0.620 ± 0.226**	**0.740 ± 0.185**

The bold entities show the best experimental results in each metric

Table 5 Tree semantic segmentation test results in terms of Jaccard Index (IoU) and F_1 score for the different architectures with DSTL image set

Architectures	RGB images		NDVI images	
	IoU	F_1	IoU	F_1
DFANet	0.359 ± 0.233	0.485 ± 0.256	0.308 ± 0.253	0.418 ± 0.282
FASSDNet	0.471 ± 0.224	0.605 ± 0.234	0.382 ± 0.248	0.506 ± 0.264
Enet	0.466 ± 0.247	0.592 ± 0.261	0.373 ± 0.265	0.489 ± 0.286
UNetFormer	0.473 ± 0.207	0.612 ± 0.213	0.370 ± 0.242	0.495 ± 0.259
LMFFNet	0.478 ± 0.216	0.614 ± 0.225	0.391 ± 0.243	0.518 ± 0.261
ResLMFFNet (proposed)	**0.503 ± 0.220**	**0.638 ± 0.222**	**0.427 ± 0.260**	**0.550 ± 0.274**

The bold entities show the best experimental results in each metric

0.7, the LMFFNet model exhibits subpar performance, whereas, with a rate of 0.3, the model's performance does not improve from the baseline. Since there is no overfitting in the ResLMFFNet model, as shown in Fig. 11, ResLMFFNet appears robust to various dropout rates. The optimal dropout rate, however, remains 0.5 based on experimental results.

Table 7 shows experimental results using various M and N parameter values corresponding to the number of SEM-Bs in SEM-B blocks. Increasing the depth of SEM-B blocks within the LMFFNet model correlates

Table 6 Ablation experiment results on dropout layer

Architectures	Params (M)	RIT-18 image set		DSTL image set		Wheat Yellow-Rust image set	
		IoU	F_1	IoU	F_1	IoU	F_1
LMFFNet	1.40	0.874 ± 0.270	0.895 ± 0.249	0.891 ± 0.261	0.909 ± 0.242	0.569 ± 0.297	0.671 ± 0.291
LMFFNet_dropout ($p = 0.5$)	1.40	0.880 ± 0.265	0.901 ± 0.247	0.893 ± 0.258	0.911 ± 0.240	0.607 ± 0.275	0.712 ± 0.254
LMFFNet_dropout ($p = 0.3$)	1.40	0.874 ± 0.266	0.898 ± 0.248	0.875 ± 0.266	0.899 ± 0.241	0.548 ± 0.308	0.648 ± 0.303
LMFFNet_dropout ($p = 0.7$)	1.40	0.868 ± 0.278	0.891 ± 0.258	0.869 ± 0.286	0.889 ± 0.267	0.490 ± 0.279	0.609 ± 0.264
ResLMFFNet_dropout(p=0.5)	1,40	**0.884 ± 0.252**	**0.913 ± 0.214**	**0.896 ± 0.256**	**0.913 ± 0.247**	**0.681 ± 0.240**	**0.781 ± 0.208**
ResLMFFNet_dropout ($p = 0.3$)	1.40	0.883 ± 0.247	0.912 ± 0.227	0.895 ± 0.254	0.913 ± 0.237	0.613 ± 0.296	0.715 ± 0.280
ResLMFFNet_dropout ($p = 0.7$)	1.40	0.882 ± 0.259	0.905 ± 0.239	0.894 ± 0.286	0.912 ± 0.268	0.608 ± 0.322	0.712 ± 0.332

Tree, crop and wheat yellow-rust semantic segmentation test results are given in terms of Jaccard Index (IoU) and F_1 score for RIT-18, DSTL and Wheat Yellow-Rust image sets

The bold entities show the best experimental results in each metric

Table 7 Ablation experiment results on different M and N parameter values

Architectures	GFLOPs	Params(M)	RIT-18 image set		DSTL image set		Wheat Yellow-Rust image set	
			IoU	F_1	IoU	F_1	IoU	F_1
LMFFNet ($M = 3, N = 8$)	12.73	1.34	**0.874 ± 0.270**	0.895 ± 0.249	**0.891 ± 0.261**	0.909 ± 0.242	**0.569 ± 0.297**	0.671 ± 0.291
LMFFNet ($M = 3, N = 12$)	15.76	1.82	0.872 ± 0.270	0.891 ± 0.251	0.875 ± 0.275	0.896 ± 0.254	0.429 ± 0.324	0.526 ± 0.335
LMFFNet ($M = 5, N = 8$)	14.27	1.40	0.874 ± 0.237	0.895 ± 0.239	0.886 ± 0.262	0.906 ± 0.242	0.560 ± 0.279	0.667 ± 0.264
ResLMFFNet ($M = 3, N = 8$)	12.73	1.34	0.884 ± 0.252	0.913 ± 0.214	0.896 ± 0.256	0.913 ± 0.247	**0.681 ± 0.240**	0.781 ± 0.208
ResLMFFNet ($M = 3, N = 12$)	15.76	1.82	**0.885 ± 0.253**	0.908 ± 0.234	**0.897 ± 0.259**	0.912 ± 0.246	0.624 ± 0.260	0.717 ± 0.255
ResLMFFNet ($M = 5, N = 8$)	14.27	1.40	0.880 ± 0.262	0.904 ± 0.242	0.895 ± 0.255	0.913 ± 0.238	0.598 ± 0.266	0.707 ± 0.254

Tree, crop and wheat yellow-rust semantic segmentation test results are given in terms of Jaccard Index (IoU) and F_1 score for RIT-18, DSTL and Wheat Yellow-Rust image sets

The bold entities show the best experimental results in each metric

Table 8 Ablation experiment results on different L_2-norm rates

Architectures	RIT-18 image set		DSTL image set		Wheat Yellow-Rust image set	
	IoU	F_1	IoU	F_1	IoU	F_1
ResLMFFNet (L_2-norm with 10^{-3})	0.883 ± 0.255	0.906 ± 0.236	0.896 ± 0.259	0.912 ± 0.242	0.599 ± 0.251	0.714 ± 0.225
ResLMFFNet (L_2-norm with 10^{-4})	0.874 ± 0.267	0.898 ± 0.248	0.896 ± 0.256	0.913 ± 0.240	0.564 ± 0.266	0.680 ± 0.243
ResLMFFNet	**0.884 ± 0.252**	**0.913 ± 0.214**	**0.896 ± 0.256**	**0.913 ± 0.247**	**0.681 ± 0.240**	**0.781 ± 0.208**

Tree, crop and wheat yellow-rust semantic segmentation test results are given in terms of Jaccard Index (IoU) and F_1 score for RIT-18, DSTL and Wheat Yellow-Rust image sets

The bold entities show the best experimental results in each metric

Table 9 Ablation experiment results on data augmentation of scaling within [0.95 1.05] range

Architectures	RIT-18 image set		DSTL image set		Wheat Yellow-Rust image set	
	IoU	F_1	IoU	F_1	IoU	F_1
ResLMFFNet (scale = (0.95, 1.05))	0.883 ± 0.256	0.906 ± 0.238	0.892 ± 0.265	0.908 ± 0.254	0.631 ± 0.255	0.740 ± 0.217
ResLMFFNet	**0.884 ± 0.252**	**0.913 ± 0.214**	**0.896 ± 0.256**	**0.913 ± 0.247**	**0.681 ± 0.240**	**0.781 ± 0.208**

Tree, crop and wheat yellow-rust semantic segmentation test results are given in terms of Jaccard Index (IoU) and F_1 score for RIT-18, DSTL and Wheat Yellow-Rust image sets

The bold entities show the best experimental results in each metric

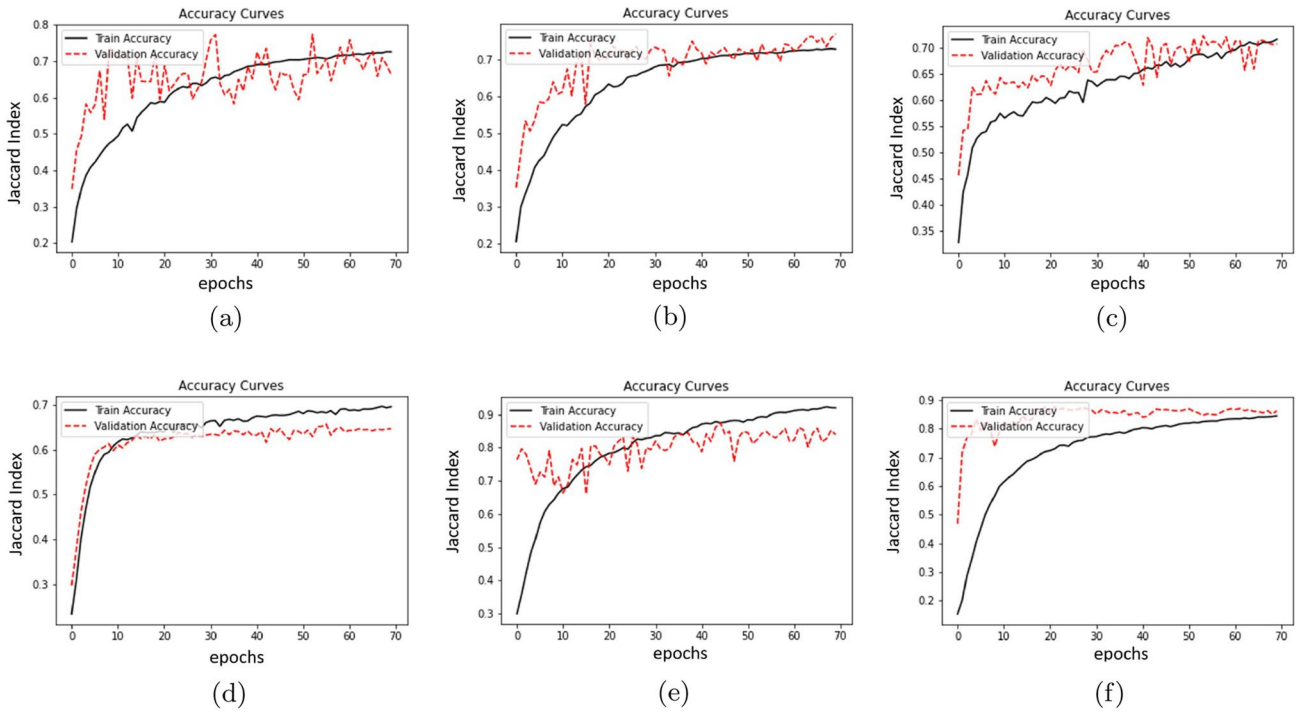


Fig. 11 Accuracy curves of training and validation sets in the training stage. **a** LMFFNet using RIT-18 image set. **b** ResLMFFNet using RIT-18 image set. **c** LMFFNet using DSTL image set. **d** ResLMFFNet using DSTL image set. **e** LMFFNet using Wheat Yellow-Rust image set. **f** ResLMFFNet using Wheat Yellow-Rust image set

with a decline in performance, a phenomenon already noted in the LMFFNet study [19]. This trend highlights the challenge of poor gradient flow inherent in deeper blocks, as evidenced by the accuracy curves depicted in Fig. 11. The ResLMFFNet model offers a solution by introducing residual connections to better leverage the potential of deeper SEM-B blocks. Notably, ResLMFFNet demonstrates performance improvements with increased M and N values. These results from the ablation study confirm that the ResLMFFNet model enhances gradient flow within deeper SEM-B blocks, helps preserve high-level features, and thereby boosts overall performance.

Based on the findings from the ablation study presented in Table 8, L_2-norm regularization does not significantly affect overall performance. Accuracy curves in Fig. 11 show that the dropout layer used in the ResLMFFNet architecture already provides sufficient regularization and eliminates overfitting.

Scale transformation is selected as a data augmentation method to distinguish detail and global content features. The region may be scaled down or up by up to 5%, yet Table 9 results indicate no notable performance enhancement.

3.3.2 Limitations

The proposed model outperforms other architectures on all image sets, yet some failure modes may affect



(a) Mask (b) Prediction

Fig. 12 Segmentation results for ResLMFFNet in the complex detailed and occluded tree objects

performance. The ResLMFFNet model produces inaccurate predictions, particularly for images prone to occlusion or containing complex details within target objects (Fig. 12).

To effectively address the occlusion problem, the literature employs a convolutional block attention module (CBAM) [25]. This module prioritizes the region of interest by weighting features in both spatial and channel dimensions. In the context of the ResLMFFNet model, enhancing occluded tree features could be future research by integrating CBAM into the decoder's input maps sourced from various scales in the encoder. Using CBAM to extract global information from fine-grained features might reduce interference from background and occluded trees.

4 Conclusions

This study introduces ResLMFFNet, an improved version of the LMFFNet model. Its design promises to overcome the challenge of balancing high accuracy with fast inference speed for various precision agriculture tasks. By incorporating residual connections into SEM-B blocks and a dropout layer in the MAD decoder structure, ResLMFFNet outperforms LMFFNet in terms of the Jaccard index without changing model parameters and significantly affecting inference time. Extensive experiments demonstrate its superiority over state-of-the-art architectures for real-time segmentation of crops, trees, and wheat yellow-rust. ResLMFFNet helps to preserve low-level features, improves generalization capability and solves the possible problem of vanishing/exploding gradients. Therefore, the proposed model supports real-time precision agriculture applications with high accuracy and fast inference time. Future work can explore optimizing and quantizing the ResLMFFNet model for deployment on embedded systems such as the Jetson TX series mounted on quadcopters.

Author Contributions In adherence to the guidelines outlined in the Instructions for Authors, the paper follows a single-author model, with the sole author performing all operations related to the manuscript.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability Source code is available on GitHub: https://github.com/iremulku/Semantic-Segmentation-in-Precision-Agriculture.

Declaration

Conflict of interest The authors declare that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Jinya, S., Zhu, X., Li, S., Chen, W.-H.: Ai meets uavs: a survey on ai empowered uav perception systems for precision agriculture. Neurocomputing **518**, 242–270 (2023)
2. Ulku, I., Akagündüz, E., Ghamisi, P.: Deep semantic segmentation of trees using multispectral images. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **15**, 7589–7604 (2022)
3. Schürholz, D., Castellanos-Galindo, G.A., Casella, E., Mejía-Rentería, J.C., Chennu, A.: Seeing the forest for the trees: mapping cover and counting trees from aerial images of a mangrove forest using artificial intelligence. Remote Sens. **150**(13), 3334 (2023)
4. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
5. Sa, I., Chen, Z., Popović, M., Khanna, R., Liebisch, F., Nieto, J., Siegwart, R.: weednet: dense semantic weed classification using multispectral images and mav for smart farming. IEEE Robot. Autom. Lett. **30**(1), 588–595 (2017)
6. Deng, J., Zhong, Z., Huang, H., Lan, Y., Han, Y., Zhang, Y.: ightweight semantic segmentation network for real-time weed mapping using unmanned aerial vehicles. Appl. Sci. **100**(20), 7132 (2020)
7. Gao, J., Liao, W., Nuyttens, D., Lootens, P., Xue, W., Alexandersson, E., Pieters, J.: Cross-domain transfer learning for weed segmentation and mapping in precision farming using ground and uav images. Expert Syst. Appl. **246**, 122980 (2024)
8. Milioto, A., Lottes, P., Stachniss, C.: Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2229–2235. IEEE (2018)

9. Qi, F., Wang, Y., Tang, Z., Chen, S.: Real-time and effective detection of agricultural pest using an improved yolov5 network. J. Real-Time Image Proc. **200**(2), 33 (2023)
10. Yang, B., Yang, S., Wang, P., Wang, H., Jiang, J., Ni, R., Yang, C.: Frpnet: an improved faster-resnet with paspp for real-time semantic segmentation in the unstructured field scene. Comput. Electron. Agric. **217**, 108623 (2024)
11. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **390**(12), 2481–2495 (2017)
12. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: a deep neural network architecture for real-time semantic segmentation (2016). arXiv:1606.02147
13. Wang, Y., Zhou, Q., Liu, J., Xiong, J., Gao, G., Xiaofu, W., Latecki, L.J.: Lednet: a lightweight encoder-decoder network for real-time semantic segmentation. In: IEEE International Conference on Image Processing (ICIP), pp. 1860–1864. IEEE (2019)
14. Kim, M., Park, B., Chi, S.: Accelerator-aware fast spatial feature network for real-time semantic segmentation. IEEE Access **8**, 226524–226537 (2020)
15. Wang, Y., Zhou, Q., Xiong, J., Xiaofu, W., Jin, X.: Esnet: an efficient symmetric network for real-time semantic segmentation. In: Conference on Pattern Recognition and Computer Vision, pp. 41–52. Springer (2019)
16. Li, H., Xiong, P., Fan, H., Sun, J.: Dfanet: deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9522–9531 (2019)
17. Rosas-Arias, L., Benitez-Garcia, G., Portillo-Portillo, J., Olivares-Mercado, J., Sanchez-Perez, G., Yanai, K.: Fassd-net: fast and accurate real-time semantic segmentation for embedded systems. IEEE Trans. Intell. Transp. Syst. **230**(9), 14349–14360 (2021)
18. Wang, L., Li, R., Zhang, C., Fang, S., Duan, C., Meng, X., Atkinson, P.M.: Unetformer: a unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. ISPRS J. Photogramm. Remote. Sens. **190**, 196–214 (2022)
19. Shi, M., Shen, J., Yi, Q., Weng, J., Huang, Z., Luo, A., Zhou, Y.: Lmffnet: a well-balanced lightweight network for fast and accurate semantic segmentation. IEEE Trans. Neural Netw. Learn. Syst. (2022)
20. Jie, H., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
21. Jaiswal, A., Wang, P., Chen, T., Rousseau, J., Ding, Y., Wang, Z.: Old can be gold: better gradient flow can make vanilla-gcns great again. Adv. Neural. Inf. Process. Syst. **35**, 7561–7574 (2022)
22. 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, pp. 770–778 (2016)
23. Kemker, R., Salvaggio, C., Kanan, C.: Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. ISPRS J. Photogramm. Remote. Sens. **145**, 60–77 (2018)
24. Jinya, S., Yi, D., Baofeng, S., Mi, Z., Liu, C., Xiaoping, H., Xiangming, X., Guo, L., Chen, W.-H.: Aerial visual perception in smart farming: field study of wheat yellow rust monitoring. IEEE Trans. Ind. Inf. **170**(3), 2242–2249 (2020)
25. Wang, Y., Qin, Y., Cui, J.: Occlusion robust wheat ear counting algorithm based on deep learning. Front. Plant Sci. **12**, 645899 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.