2025 T2 COMP9517 Project Report

Group name : Try to be better

Group member:

Shuo Yang
Z5389396

Xicheng Peng
Z5526784

Xiangming Hou
Z5564067

Weizhao Yuan
Z5530932

Yichen Zhu
Z5610836

Abstract—**In this project, in order to identify dead trees in forest aerial images, four processing methods (two traditional + two depths) are used in this paper. The traditional methods are Watershed and MeanShift, and the deep learning methods are DeepLabv3+ResNet-50 and U-Net. Then, the channel part of data preprocessing is classified into RGB, NRG, NRG+RGB, in order to explore the best processing model and parameter Settings. Finally, the current optimal model U-Net is improved, including the adjustment of U-Net ++ and the use of augmentation strategy. The dense skip connection structure of U-Net++ is better at recovering details than the original U-Net, and the data obtained by medium augmentation is better.**

Keywords—Watershed, MeanShift, DeepLabv3+ResNet-50, styling, U-Net, U-Net ++

1. INTRODUCTION

A. Background

As an important part of the earth's ecosystem, forests achieve carbon cycling and climate regulation functions while maintaining biodiversity protection. However, dead trees have great hidden dangers in some forest areas that are prone to wildfires, which will cause irreversible harm to the natural environment, humanity and economy.

Traditional manual detection methods rely on inspectors to carry out field investigations, which have limited detection scope and high cost and low efficiency. As a subjective test, the acceptance standard could not be unified, and the timeliness is not strong enough.

The rapid development of remote sensing technology and computer vision in the 21st century makes the aerial images of unmanned aerial vehicles as data, and the automated batch analysis of computer vision achieves efficient processing while completing a more objective and unified measurement standard. The introduction of computer vision technology greatly reduces the error and ensures the safety of inspectors.

B. Task and Dataset

In this task, two traditional methods and two deep learning methods will be used to compare and explore the effect of multi-mode on recognition accuracy.

The dataset is from Kaggle's "Aerial Imagery for Dead Tree Segmentation". The sample size is 444 aerial images. The bands used are

- **RGB:** common color channels information and provide common information under visible light, which is useful for judging the details of trees, such as overall shape and texture details.
- **NRG:** sensitive to the spectral reflection of healthy plants, which can more effectively distinguish dead plants from healthy plants.

However, there are still many problems in processing the current dataset. Besides the problem that the texture of dead trees is similar to the environment (stones, soil), the shadows of trees caused by the high-altitude perspective may be recognized as dead trees in computer vision technology.

Problem statement:

The single-modal recognition leads to insufficiently precise processing of the dataset (misjudgment of shadows and necrosis, marginalization of the non-recognition of small area necrotic trees)

Compared with traditional machine learning methods based on spectral and texture features and deep learning segmentation networks (such as DeepLabv3) based on multi-channel input (RGB + NIR), this method achieves more accurate screening to improve the identification of trees covered by shadows. Through quantitative (IoU, Precision, Recall) and qualitative visualization evaluation, it is proved that multimodal fusion improves the segmentation accuracy.

2. LITERATURE REVIEW

In dead-tree segmentation, the model must accurately distinguish between adjacent tree crowns, shadows, and the background, thereby providing reliable support for forest health monitoring. This section will focus on introducing the application scenarios for these four methods and the processing of their segmentation features.

A. Morphological Processing and Watershed Background

The watershed algorithm treats a grayscale image as a topographic surface, using gradients to identify “basins” (local minima) and watershed markers. It then simulates water rising from each basin until meeting at ridge lines, which form the segmentation boundaries. This method offers highly accurate edge localization and is commonly used for segmenting complex shapes—such as extracting cell or tissue contours in biomedical images and detecting surface defects on industrial parts. However, because it is sensitive to noise and minor gradient variations, it often requires morphological preprocessing or marker-controlled strategies to prevent over-segmentation [1].

B. Kernel Density Estimation and Mean Shift Clustering

Mean Shift is an unsupervised clustering technique based on kernel density estimation. In the finite feature space of image pixels, each point is iteratively shifted toward the direction of highest local density, eventually forming clusters that correspond to segmented regions. Mean Shift can adaptively adjust both the shape and number of clusters, making it popular for image segmentation, video object tracking, and data-visualization dimensionality reduction. Its major drawbacks are high computational complexity and sensitivity to the bandwidth parameter, which must be carefully tuned based on image resolution and content[2].

C. Convolutional Neural Networks and the U-Net Architecture

Convolutional neural networks (CNNs) leverage weight sharing to extract hierarchical spatial features efficiently. The U-Net architecture builds on a classic encoder–decoder design by introducing skip connections: features from shallow encoder layers are concatenated with upsampled decoder outputs to recover fine boundary details. Originally developed for biomedical image segmentation, U-Net has become a go-to approach for tasks such as road and water-body extraction in remote sensing and other small-sample segmentation problems. Its symmetric structure enables rapid convergence and precise boundary localization even with limited training data[3].

D. Atrous Convolution and DeepLabv3

DeepLabv3 expands the receptive field without increasing computation via atrous (dilated) convolutions and incorporates an Atrous Spatial Pyramid Pooling (ASPP) module to fuse multi-scale contextual information. Built typically on deep backbones like ResNet, DeepLabv3 has become a benchmark for segmentation in complex scenes such as urban street views and remote sensing classification. Its capacity to capture information at multiple scales makes it particularly effective for targets of varying sizes against heterogeneous backgrounds[4].

E. Multispectral Fusion Background

In remote sensing and vegetation monitoring, RGB imagery is often confounded by shadows and illumination changes, making it hard to distinguish healthy versus dead vegetation. Near-infrared (NIR) channels, however, are highly sensitive to chlorophyll reflectance and remain responsive even in shadows. The Normalized Difference Vegetation Index (NDVI), first introduced by Rouse et al. (1974), quantifies vegetation health based on the ratio of visible and NIR reflectance[5]. In the deep-learning era, concatenating RGB and NIR into a multi-channel input allows end-to-end learning of joint spectral–spatial features, significantly improving model robustness and segmentation accuracy under shadow interference, multi-scale targets, and background heterogeneity.

3. Methods

A. Statistical analysis of the dataset

The image dataset used in this study contains remote sensing images in RGB and NRG formats, as well as the corresponding binary segmentation masks. Ten groups of samples were randomly selected to analyze the distribution of image attributes and target regions, and the results are as follows.

Image size and pixel properties

RGB and NRG images: The original resolution of the two formats is different, ranging from 340 to 478 pixels. Each image is of type uint8 with pixel values ranging from 0 to 255. Mask image: Same spatial resolution as RGB/NRG counterpart, single channel uint8 type, pixel value only includes **0** and **255**.

Mask pixel value distribution

In the 50 masks of the sampled data, the background (value 0) and the target area (value 255) each account for 50%, indicating that the number of target and non-target images is balanced at the same file level, as shown in Figure. 1.

Distribution of Mask pixel value.

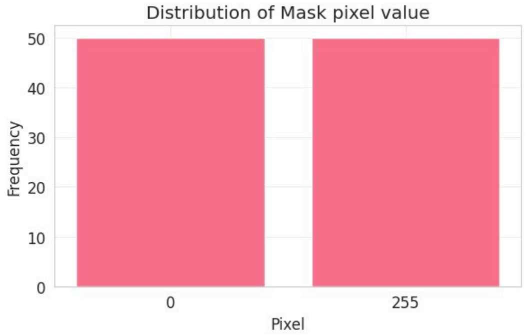


Figure. 1. Distribution of Mask pixel value

Analysis of target area proportion

It can be observed from Figure. 2 that the average pixel area ratio of the target area (dead tree crown) is 1.9%, the lowest is about 0.2%, and the highest is about 8.8%.

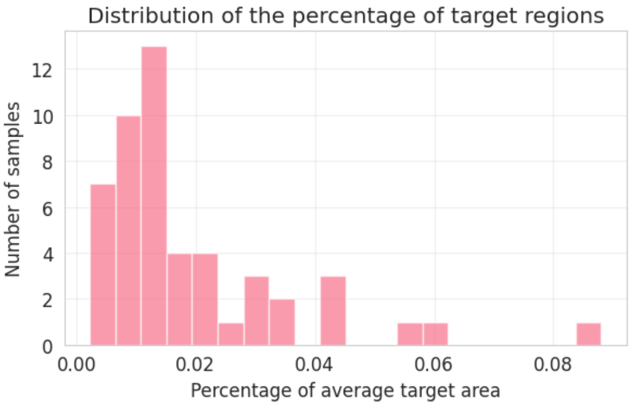


Figure. 2. Distribution of the percentage of target regions

The vast majority of samples have a target area proportion below 4%, and as shown in the histogram, the distribution is highly biased towards small proportion regions, which indicates that the target regions in the dataset are small and scattered.

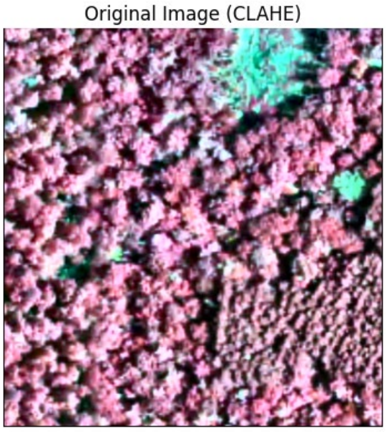


Figure. 3. Org Image

The original inspection diagram is shown in Figure. 3.

Distribution characteristics and influence

The high proportion of small objects and their dispersed morphology indicate that boundary detection is more difficult in segmentation tasks, especially for the traditional Watershed method based on pixel gradient, which is prone to over-segmentation.

The NIR band can significantly enhance the spectral difference between healthy and dead tree crowns in NRG data, which is helpful to improve the discrimination ability of the clustering algorithm in the small target background.

B. Watershed segmentation

Principle

In the dead tree segmentation task of this study, the watershed algorithm is used to effectively separate adjacent tree crowns and shadows. In this method, the input image is converted into a gray-scale terrain model, where the gray-scale value represents "height" [6]. The gradient or distance transformation is calculated to identify the potential crown area and the corresponding crown boundary in the image). Then, the process of "water filling" is simulated, where different waters meet at the boundary to form a dividing line [7].

When distinguishing dead trees from normal trees, healthy tree crowns usually have high reflectance in the NIR band, while dead trees have low reflectance in the NIR band. Therefore, through CLAHE enhancement and morphological processing, it is easier to separate the two regions and reduce over-segmentation caused by shadow or canopy connection[1].

Advantages

1) Due to the enhancement of local contrast, it is helpful for edge extraction of small objects such as tree crowns.
2) It reduces the over-segmentation problem of touching objects caused by the combination of distance variation and morphology

Disadvantages

1) Over-segmentation can still occur when the labeled points are not accurate[1].
2) Sensitive to noise, the preprocessing parameters need to be tuned precisely.

C. Mean Shift clustering segmentation

Principle

In dead tree recognition, the mean shift algorithm uses color and spatial information to perform kernel density estimation in the feature space, and iteratively moves pixels to the local maximum density position[8]. This non-parametric clustering method based on color-space distribution can automatically form different clusters without presetting the number of categories, so that the NIR bands of healthy trees and dead trees can be assigned to different cluster regions due to the differences in light reflectance[9]. Through appropriate spatial radius and color radius parameters, the fidelity of the crown boundary is further improved, and the influence caused by similar colors is relatively weakened.

Advantages

1) No need to specify the number of clusters and can adapt to objects with complex shapes[8].
2) Effectively improve boundary clarity and segmentation stability.

Disadvantages

1) Increased computational cost, especially in high-resolution images[2].
2) The results of fine-tuning parameters have an excessive impact on the fluctuation of the results and require additional optimization.

D. DeepLabv3 (NRG input)

Principle

Deep convolutional networks with atrous Spatial Pyramid Pooling (ASPP).

Advantage

1) Adaptively learns spectral-spatial features, robust to complex backgrounds and multi-scale objects[10].

Disadvantages

1) The lack of blue light band leads to poor discrimination in some cases where spectral features are close (e.g., shadows and dead canopy).

E. DeepLabv3 (6-channel RGB + NIR fusion input)

Principle

Considering the 3-channel defect, combine RGB and NIR-RGB (NRG) into 6-channel input, and change the first convolution layer of the model to in_channels=6. It enhances the ability of spectral-spatial joint representation of healthy and dead trees, and is more stable for the segmentation of shadow areas, similar spectral targets, and complex backgrounds.

Input: RGB 3 channels concatenated with NIRRGB (NRG) 3 channels, total 6 channels.

Advantages

1) It has both color/texture information of visible light and vegetation activity signal of near infrared, which makes the distinction between shadow and dead area more accurate[11].

F. U-Net (RGB + NIR input)

Principle

Symmetrical encoder-decoder architecture, where the encoder extracts multi-scale features and the decoder fuses the shallow high-resolution with skip connections to achieve accurate boundary recovery[12].

Input: The combination of near-infrared (NIR) band and RGB three bands to construct a four-channel input, so that the model can perceive the difference of visible light and vegetation spectral characteristics at the same time, which is conducive to distinguish healthy canopy from dead canopy.

Advantages

1) Simple structure, small number of parameters, fast training speed, suitable for limited data
2) Multi-spectral input enhances feature representation, especially in small objects or complex backgrounds[13].

G. Summary of Methods

From the input channel analysis, according to the proposed four segmentation methods -- Watershed, Mean Shift, U-Net and DeepLabv3 -- the performance of RGB and NRG input channels are compared. Through these two sets of tables, the applicable scenarios and limitations of each method are shown, which provides reference for subsequent experimental results verification, algorithm selection and optimization.

The comparison Tables 1 and 2 are shown below

Table 1.RGB input channel comparison

Comparison of RGB detection	Characteristics	Advantages	Disadvantages
Watershed	Edge detection with grayscale gradients	Sensitive to boundaries	Easy to over-segment and is disturbed by shadows
Mean Shift	Color-spatial clustering	No need to set the number of clusters	Low discrimination of color similar objects
U-Net	Encoder-decoder + skip connection	High-resolution features with clear boundaries	It's easy to overfit small data
DeepLabv3	Dilated convolution + Multi-scale features	Complex backgrounds are more robust	Computationally expensive

Table 2.NRG input channel comparison

Comparison of NRG detection	Characteristics	Advantages	Disadvantages
Watershed	Enhanced boundary detection	NIR improves contrast	Part of the boundary spectra may overlap
Mean Shift	Color-spatial clustering	The clustering is more stable	Sensitive to bandwidth parameters
U-Net	Encoder-decoder + skip connection	The preservation of the target boundary is better	easy to overfit
DeepLabv3	Dilated convolution + Multi-scale features	NIR+ Multi-scale Segmentation of complex background	Inference speed and Computationally speed slow

4. Experimental Results

A. Traditional Segmentation Methods

Watershed segmentation

1) Data preprocessing

The RGB/NRG image and the corresponding mask were resampled to 366×385, which greatly simplified the input consistency of subsequent processing

2) Preprocessing channel

BGR → LAB color space conversion;

Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to L channel to enhance the contrast;

The enhanced L channel and the original A/B channel were merged and converted back to BGR;

The enhancement result is converted to a grayscale image, and the subsequent segmentation step is expanded based on the grayscale image

3) Parameter selection

CLAHE: default clip Limit = 2.0, tile Grid Size = (8,8);

Gaussian Blur: kernel size 3×3;

Triangle Thresholding: cv2.THRESH_OTSU;

Morphological operations: Opening and dilation use a 3×3 rectangular structuring element.

Opening iter=1/2, closing iter=1

Distance transform: default Euclidean distance;

Connected component: The default 8-neighborhood marks the foreground seed.

4) Segmentation process

The gray image is segmented by Gaussian blur → Triangle threshold;

Morphological opening operation + dilation denoising;

The distance transformation was calculated to extract the "definite foreground";

Subtract to get the "unknown region";

Connected components mark foreground seeds → *cv2.watershed()* to get the final segmentation

5) Evaluation Metric

Intersection over Union (IoU) is used to measure the overlap between the predicted mask and the true mask:

$$IoU = \frac{Predicted\ value \cap True\ value}{Predicted\ value \cup True\ value}$$

When the prediction mask was binarized, the grayscale image was converted to binary according to the threshold value >127. The mIoU is obtained by averaging over the full dataset.

MeanShift segmentation

1) Data preprocessing

The RGB/NRG image and the corresponding mask were resampled to 366×385.

2) Preprocessing channel

BGR → LAB → CLAHE → BGR;

The CLAHE enhanced results are down-sampled (factor≈0.5) to accelerate the MeanShift.

It is resampled back to the original size and converted to grayscale for threshold segmentation and morphological cleaning

3) Parameter selection

MeanShift: spatial radius =20, color radius =20, maxLevel=1;

Downsampling: the scale is about 0.5.

Thresholding: Otsu automatic thresholding.

Morphological cleaning: open operation + dilation, kernel 3×3, iter=1.

4) Segmentation process

After reducing sampling *cv2. PyrMeanShiftFiltering ()* divided by smooth;

The foreground mask was extracted by resampling, grayscale → binary threshold → morphological operation.

5) Evaluation Metric

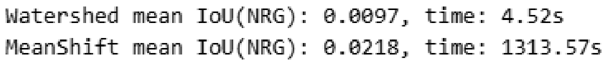
Intersection over Union (IoU) is used to measure the overlap between the predicted mask and the true mask:

$$IoU = \frac{Predicted\ value \cap True\ value}{Predicted\ value \cup True\ value}$$

When the prediction mask was binarized, the grayscale image was converted to binary according to the threshold value >127. The mIoU is obtained by averaging over the full dataset

6) Model Results Display

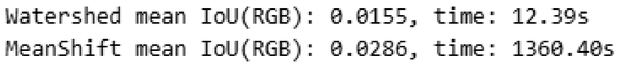
The result of code operation is shown in Figure. 4.



```
Watershed mean IoU(NRG): 0.0097, time: 4.52s
MeanShift mean IoU(NRG): 0.0218, time: 1313.57s
```

Figure. 4. Watershed & MeanShift model results (NRG)

The result of code operation is shown in Figure. 5.



```
Watershed mean IoU(RGB): 0.0155, time: 12.39s
MeanShift mean IoU(RGB): 0.0286, time: 1360.40s
```

Figure. 5. Watershed & MeanShift model results (RGB)

The optimal IoU image of Watershed in the NRG channel state is shown in Figure. 6.

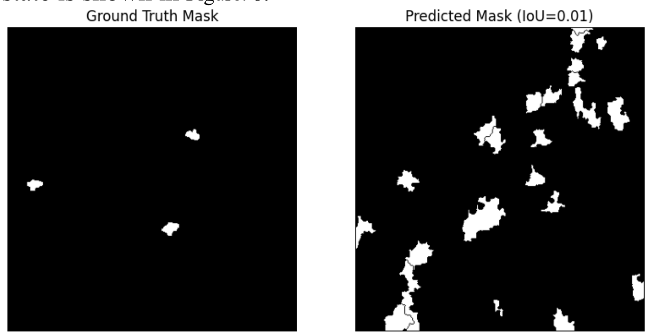


Figure. 6. Watershed IoU-NRG

The optimal IoU image of MeanShift in the NRG channel state is shown in Figure. 7.

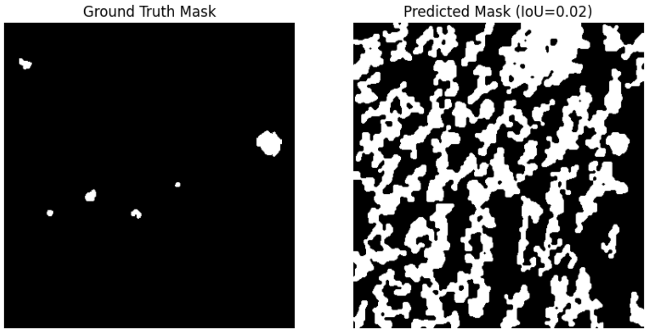


Figure. 7. MeanShift IoU-NRG

The optimal IoU image of Watershed in the RGB channel state (best performance) is shown in Figure. 8

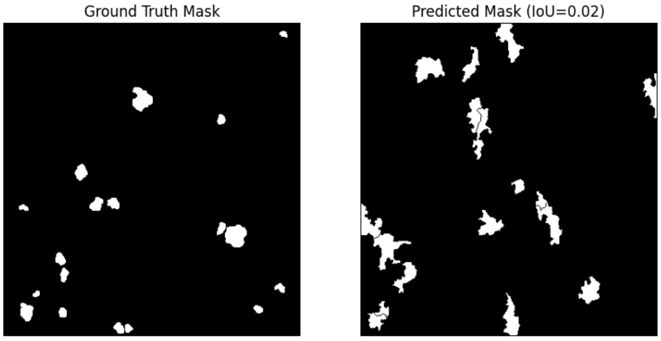


Figure. 8. Watershed IoU-RGB

The optimal IoU image of MeanShift in the RGB channel state (best performance) is shown in Figure. 9

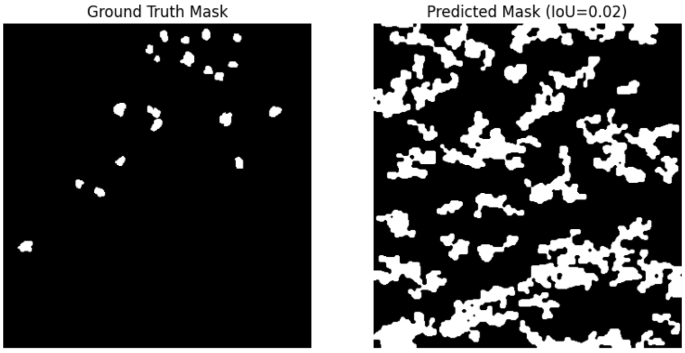


Figure. 9. MeanShift IoU-RGB

The results of the two traditional models were run and presented in Table 3.

Table 3. Watershed & MeanShift model results

Watershed & MeanShift model results	mIoU of RGB images	mIoU of NRG images
Watershed	0.0155	0.0097
MeanShift	0.0286	0.0218

The bar chart generated according to the table is more intuitive to see the processing of the traditional model, as shown in Figure. 10.

Under the same method, RGB has a slightly higher mIoU than NRG (Watershed: +0.0058; MeanShift: +0.0068), indicating that the visible light channel provides more texture information to the segmentation boundary.

MeanShift's mIoU is about 2.0× the Watershed's (NRG: 0.0218 vs.0.0097; RGB: 0.0286 vs. 0.0155), which is obviously superior in accuracy.

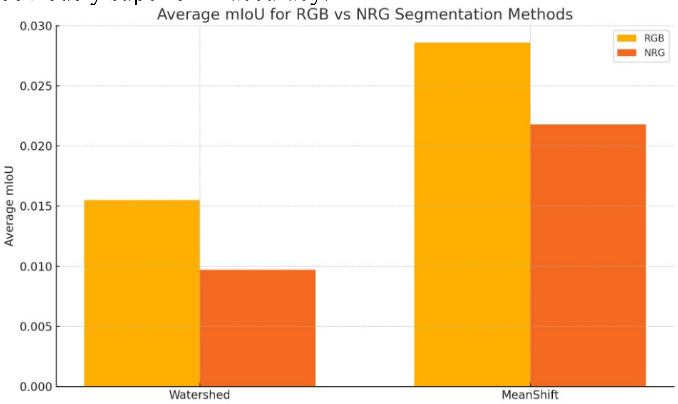


Figure. 10. Average mIoU for RGB vs NRG Segmentation

B. Deep Learning–based Methods

Deep Learning-based Method (NRG-only)

1) Network structure

Backbone: ResNet-50 (pre-trained on ImageNet)

Segmentation Head: DeepLabv3 + ASPP

Input channels: in_channels=3 (NRG channels only)

Data Preprocessing & Augmentation

Resampling: All images with the mask resized to 366×385 pixels

Normalization: channel-wise normalized by ImageNet mean/standard deviation

2) Training augmentation:

Random horizontal/vertical flip (probability 0.5)

ColorJitter (brightness/contrast/saturation/hue Jitter)

Validation flow: only resampling and normalization without any augmentation

3) Parameter setting

Optimizer: Adam with initial learning rate = 1×10^{-4}

Loss function: Binary cross-entropy

Learning rate Schedule: ReduceLROnPlateau

Training strategy: 30 epochs, batch size = 4

Dataset split: 80% training / 20% validation

4) Evaluation metrics

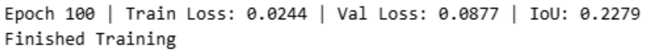
Val Loss;

mIoU: Intersection and union ratio of foreground classes

Precision/Recall: Optional for more fine-grained category analysis

5) Experimental Results

The result of code operation is shown in Figure. 11.



```
Epoch 100 | Train Loss: 0.0244 | Val Loss: 0.0877 | IoU: 0.2279
Finished Training
```

Figure. 11. Deeplabv3 -NRG model results (NRG)

Deep Learning-based Method (NRG+RGB)

DeepLabv3+ResNet-50 multi-modal fusion model is used

1) Network Architecture

Backbone: Pretrained ResNet-50

Segmentation Head: DeepLabv3 with Atrous Spatial Pyramid Pooling (ASPP)

Input Adaptation: The first convolution is modified from *in_channels*=3 to *in_channels*=6 to accept fused RGB+NRG inputs

2) Data Preprocessing & Augmentation

Resizing: All images and masks are resized to 366×385 pixels

Normalization: Applied using ImageNet mean and standard deviation

Training Augmentations:

Random horizontal and vertical flips (p=0.5)

Color jitter (brightness, contrast, saturation, and hue adjustments)

Only resizing and normalization (no augmentations)

3) Parameter Settings

Optimizer: Adam with initial learning rate = 1×10^{-4}

Loss Function: Binary cross-entropy combined

Learning Rate Scheduler: ReduceLROnPlateau

Training Regime: 30-50 epochs, batch size = 4

Dataset Split: 80% training / 20% validation

4) Evaluation Metrics

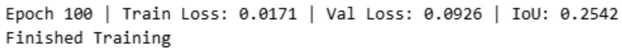
Mean Intersection over Union (mIoU): Computed for the foreground class

Precision and Recall: Optional, for more detailed class-specific analysis

5) Experimental Results

The NRG-only variant trained from epoch 25 to 100 showed persistently low mIoU.

The result of code operation is shown in Figure. 12.



```
Epoch 100 | Train Loss: 0.0171 | Val Loss: 0.0926 | IoU: 0.2542
Finished Training
```

Figure. 12. Deeplabv3 -NRG+RGB model results

The multimodal model (RGB+NRG) trained for 100 epochs achieved substantially higher mIoU compared both to the traditional methods and the single-modality baseline, with markedly improved alignment of extracted tree crowns and shadows

6) Model Results Display

The optimization results of upgrading from three channels (NRG) to six channels (RGB+NRG) are presented in Table 4.

Table 4. Deeplabv3 -NRG & Deeplabv3 -NRG+RGB

Deeplabv3 -NRG & Deeplabv3 -NRG+RGB	Val Loss	mIoU
Deeplabv3 -NRG	0.0877	0.2279
Deeplabv3 -NRG+RGB	0.0926	0.2542

As shown in Figure. 13, it can see even more performance improvement after upgrading.

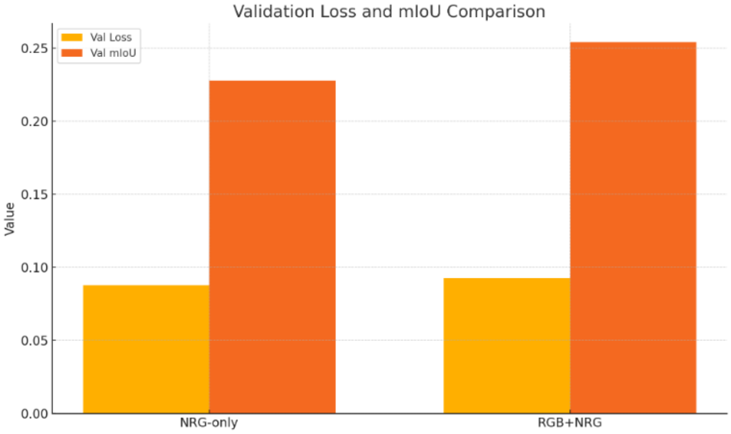


Figure. 13. Validation Loss and mIoU Comparison

By extending the input from single-channel NRG to multi-modal RGB+NRG, the mIoU can be increased from 0.2279 to 0.2542, and the gain is about 0.0263 (about 11.5% improvement), which fully demonstrates the complementary effect of visible light information on crown detail segmentation.

The validation loss of the multi-modal version is slightly higher than that of the single channel (0.0926 vs. 0.0877), possibly due to more high-frequency variations brought by the fusion of RGB channels, the training process is more difficult to fully converge, but the overall segmentation quality (mIoU) still benefits more.

The slight increase in loss does not prevent the model from learning richer features, and the multimodal deep model performs better on the overall boundary identification.

The single-channel model converges faster and has a lower validation loss, but lacks color and texture cues, resulting in a limited mIoU of ~0.23.

As shown in Figure. 14, it presents the actual ground annotation and the predicted annotation under the combination of NRG.

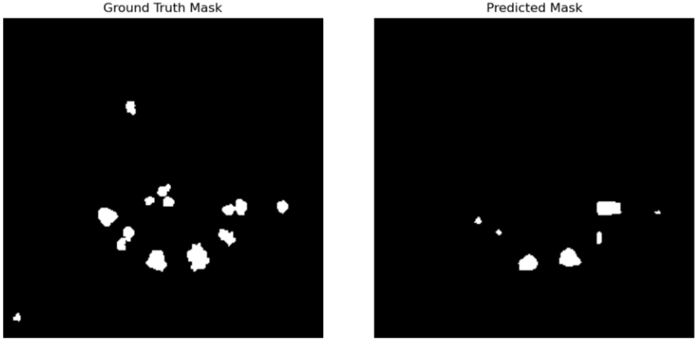


Figure. 14. DeepLabv3+ResNet-50 Prediction Comparison (NRG)

As shown in Figure. 15, it presents the actual ground annotation and the predicted annotation under the combination of (RGB+NRG)

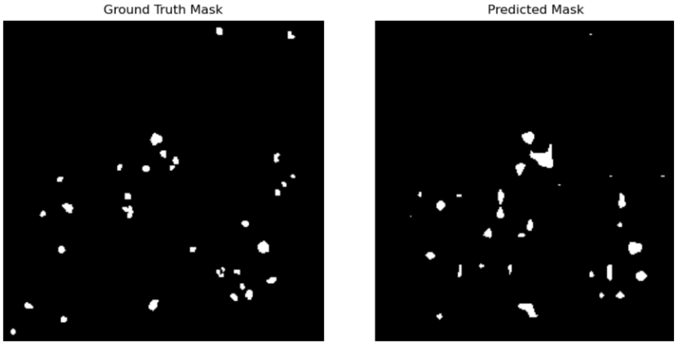


Figure. 15. DeepLabv3+ResNet-50 Prediction Comparison (NRG+RGB)

C. U-Net Methods

1) Network structure

Backbone Encoder: ResNet-34 pretrained on ImageNet

Decoder: standard U-Net upsampling path with skip connections

Channel configuration:

RGB-only (3 channels)

NRG-only (3 channels)

RGB+NIR (4 channels)

RGB+NRG (6 channels)

Enhanced variant: U-NET ++ (unet_plus_plus) was also tested on the same channel configuration

2) Data Preprocessing & Augmentation

Resampling: All images and masks were resized to 366×385 pixels

Normalized: channel-wise normalized by ImageNet mean and standard deviation

Training augmentation:

Random horizontal/vertical flip (p=0.5)

Color dithering (Brightness/Contrast/Saturation/Hue)

Validation flow: only resampling and normalization, no geometry or color enhancement

3) Parameter setting

Optimizer: Adam with an initial learning rate of 1×10^{-4}

Loss function: Binary cross-entropy + Dice loss

Learning rate Schedule: ReduceLROnPlateau (patience=8, factor=0.5)

Training strategy:

num_epochs = 40;

U-Net fixed 50 rounds;

U-net ++ to convergence, batch size=8

Dataset split: 80% training / 20% validation

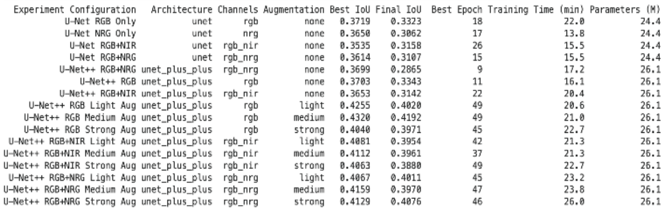
4) Evaluation metrics

mIoU: Intersection and union ratio of foreground classes

Precision/Recall: Optional for more fine-grained category analysis

5) Experimental Results

The result of code operation is shown in Figure. 16.



```
📊 Complete Experiment Results Comprehensive Report
==================================================================================
   Experiment Configuration   Architecture Channels Augmentation Best IoU Final IoU  Best Epoch Training Time (min) Parameters (M)
            U-Net RGB Only            unet      rgb         none   0.3719    0.3323          18                22.0           24.4
            U-Net NRG Only            unet      nrg         none   0.3650    0.3062          17                13.8           24.4
             U-Net RGB+NIR            unet  rgb_nir         none   0.3535    0.3158          26                15.5           24.4
             U-Net RGB+NRG            unet  rgb_nrg         none   0.3614    0.3107          15                15.5           24.4
          U-Net++ RGB+NRG unet_plus_plus  rgb_nrg         none   0.3699    0.2865           9                17.2           26.1
              U-Net++ RGB unet_plus_plus      rgb         none   0.3703    0.3343          11                16.1           26.1
          U-Net++ RGB+NIR unet_plus_plus  rgb_nir         none   0.3653    0.3142          22                20.4           26.1
    U-Net++ RGB Light Aug unet_plus_plus      rgb        light   0.4255    0.4020          49                20.6           26.1
   U-Net++ RGB Medium Aug unet_plus_plus      rgb       medium   0.4320    0.4192          49                21.0           26.1
   U-Net++ RGB Strong Aug unet_plus_plus      rgb       strong   0.4040    0.3971          45                22.7           26.1
U-Net++ RGB+NIR Light Aug unet_plus_plus  rgb_nir        light   0.4081    0.3954          42                21.3           26.1
U-Net++ RGB+NIR Medium Aug unet_plus_plus  rgb_nir       medium   0.4112    0.3961          37                21.3           26.1
U-Net++ RGB+NIR Strong Aug unet_plus_plus  rgb_nir       strong   0.4063    0.3880          49                22.7           26.1
 U-Net++ RGB+NRG Light Aug unet_plus_plus  rgb_nrg        light   0.4067    0.4011          45                23.2           26.1
U-Net++ RGB+NRG Medium Aug unet_plus_plus  rgb_nrg       medium   0.4159    0.3970          47                23.8           26.1
U-Net++ RGB+NRG Strong Aug unet_plus_plus  rgb_nrg       strong   0.4129    0.4076          46                26.0           26.1
```

Figure. 16. U-Net mIoU Comparison

6) Model Results Display

After screening the data, the data used as control experiments were collated into tabular output Table 5.

Table 5. U-Net mIoU & Time results

U-Net mIoU & Time results	Training time (min)	Best IoU	Final IoU
U-Net RGB Only	22.0	0.3719	0.3323
U-Net NRG Only	13.8	0.3650	0.3062
U-Net RGB+NIR	15.5	0.3535	0.3158
U-Net RGB+NRG	15.5	0.3614	0.3107
U-Net++ RGB (Light Aug)	20.6	0.4255	0.4020
U-Net++ RGB (Medium Aug)	21.0	0.4320	0.4192
U-Net++ RGB (Strong Aug)	22.7	0.4040	0.3971

Models using the RGB channels have all been pre-trained on ImageNet, which is why their performance is comparable to (or even better than) that of multi-channel models.

Under the same channel configuration (RGB-only), U-Net++ outperforms the original U-Net by approximately **5.3%** in best IoU (0.4255 vs. 0.3719), at the cost of only ~1.4 minutes additional training time, demonstrating a clear accuracy gain.

For the vanilla U-Net, **RGB-only** (0.3719) slightly surpasses **NRG-only** (0.3650), while multimodal fusion (RGB+NIR, RGB+NRG) fails to exceed the RGB-only baseline—indicating that, for this architecture, visible-spectrum information is most critical.

Medium Augmentation achieves the highest best IoU of **0.4320**, representing a **1.5%** improvement over Light Augmentation (0.4255) and a **6.9%** improvement over Strong Augmentation (0.4040).

Although Strong Augmentation increases sample diversity, it also introduces excessive distortion—such as image blurring and cropping artifacts—leading to reduced performance.

The configuration **U-Net++ (RGB-only, Medium Augmentation)** delivers the best final IoU of **0.4192** after **21.0 min** of training, representing the optimal trade-off between resource consumption and segmentation accuracy.

By contrast, the original U-Net completes training in **15–22 min** but yields a maximum final IoU of only **0.3323**, which may be insufficient for applications requiring high-precision segmentation.

5. DISCUSSION

The previous section shows the experimental results and performance characteristics of two traditional unsupervised methods (Watershed, MeanShift), two deep learning methods (DeepLabv3+ResNet-50 unimodal/multi-modal), and two U-Net family methods (U-Net/U-Net++). This section discusses and compares the results obtained from the experiment.

A. Traditional methods vs. Deep Learning methods

The optimal IoU of the operation results of these methods was screened out and made into a bar chart for comparison, as shown in Figure. 17.

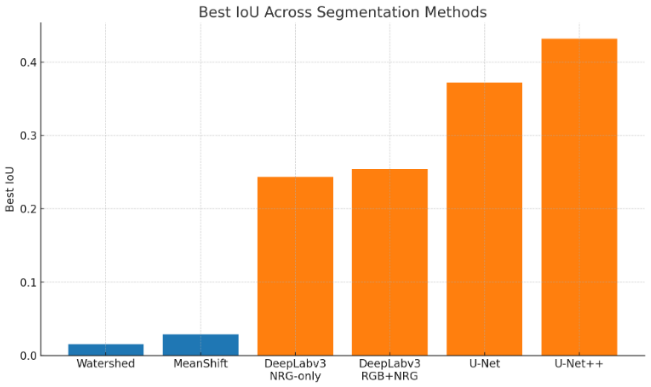


Figure. 17. Best IoU Across Segmentation Methods

Accuracy (mIoU) comparison

The highest mIoU of the traditional method is only 0.0286 (MeanShift) on RGB and even lower (0.0218) on NRG. This shows that traditional methods still have shortcomings in dealing with the task of image information segmentation.

The deep learning method (DeepLabv3 multi-modal) improves the mIoU to 0.2542, and the accuracy is improved by about 9-12 times.

The medium augmentation for U-Net++ is even higher, lifting the data to 0. 4320, an accuracy improvement of about 20 times.

Efficiency comparison

The traditional method runs completely on CPU, and the processing time for a single image is ~ 0.5-1 s. It does not need GPU participation, and can be deployed quickly.

Deep models take longer to train and require GPU memory allocation. The average inference time is 0.1-0.2s per page, which is suitable for real-time application scenarios.

Deployment cost

The traditional method does not need model parameters and only relies on OpenCV.

The number of deep model parameters increases from 24 M (DeepLabv3 NRG-only) to 26 M (multi-modal, U-Net++), which requires video memory and storage resources.

B. Input channel Effects

Figure. 18. is plotted according to the data by comparing the single-mode & multi-mode with the corresponding methods.

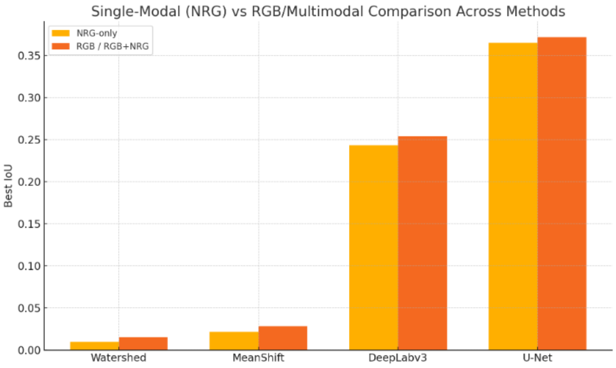


Figure. 18. Single-Modal (NRG) vs RGB/Multimodal

Unimodal vs. multi-modal

The traditional method does not use multi-modal, only the data recording and comparison of NRG and RGB are carried out, but it can also be clearly found that the processing of RGB will be slightly improved under the traditional method.

DeepLabv3 is promoted from single-channel NRG-only (mIoU = 0.2279) to RGB+NRG (mIoU = 0.2542), with a gain of about 0.0263.

The U-Net series also achieves better performance than the single mode under the RGB+NRG channel combination, indicating that the visible light and near infrared information are highly complementary.

NRG-only relative weakness

Relying only on the near-infrared channel will lose the texture and color information in RGB, resulting in insufficient boundary judgment and segmentation.

C. Data augmentation strategies

Analyze the augmentation strategy of U-Net, which currently performs best overall

No augmentation vs augmentation

1) None

Do only resampling and normalization without any geometric or color transformations.

Effect: The model only sees the original samples, which makes it the weakest to generalize.

2) Light

Random horizontal/vertical flip (p=0.5)

Effect: Only the most basic geometric diversity is added, which prevents the model from being overly sensitive to mirror invariance.

3) Medium

Light + small rotation (±15°) + brightness/contrast jitter;

Effect: While preserving the shape and structure of the target, we introduce moderate geometric and color changes, which greatly improves the robustness of the model to changes in scene illumination, pose, etc.

4) Strong

Medium + random cropping + Gaussian blur;

Effect: Most aggressive augmentations, which have the most diversity, but introduce too much distortion (blurring of edges, loss of key information from cropping), which makes the training noisy.

The bar chart of IoU data corresponding to the four augmentation is shown in Figure. 19.

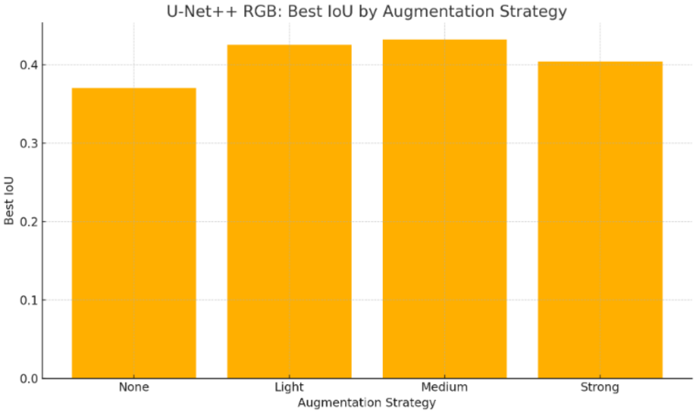


Figure. 19. U-Net++ RGB: Best IoU by Augmentation strategy

In U-Net++, the "Medium augmentation " strategy (random flip + color jitter + small rotation) improves the highest mIoU from 0.3699 without augmentation to 0.4320, which is a significant gain.

Compared with "Light augmentation", "Medium" strategy includes both geometric transformations such as rotation and flip, as well as color perturbation, but does not make large-scale adjustments to the original image, which improves the learning breadth of the model, and does not lose important details of the target.

"Intensity boosting" fails to find a balance between noise and excessive deformation, and performs slightly worse than moderate boosting, suggesting that excessive boosting may introduce irrelevant perturbations and ensure that the learned features are meaningful.

Augmentation category selection

According to the comparison and analysis, the data processing results corresponding to **medium** intensification are the best. It is suggested that geometric flipping and mild color jitter are preferred while preserving shape information.

D. Comparison of network architectures

U-Net vs. U-Net++

U-Net++ outperforms U-Net ontology across all channels and augmentation configurations, with an average mIoU improvement of 3-6%.

Dense jump connections and deeper feature fusion are helpful for detail recovery, especially for more accurate crown edge segmentation.

E. Qualitative misclassification analysis

Shadow and background: Traditional methods are easy to misclassify shadows as foreground. The deep model has a certain ability to correct them, but there is still edge blur.

Adjacent tree separation: unsupervised methods are difficult to segment the adjacent tree combination region. U-net ++ performs best at watershed boundaries.

Over-fragmentation: In the case of Strong Augmentation, some models show over-fragmentation.

6. Conclusion

A. Effective methods & Strategies

Deep learning methods significantly outperform traditional methods

The mIoU of DeepLabv3+ResNet-50 multi-modal (RGB+NRG) is increased to 0.2542, which is more than 10 times higher than that of Watershed/MeanShift.

U-net ++ (RGB-only + Medium Aug) mIoU reaches 0.4320, which is the best of all methods.

Multi-channel fusion gives full play to complementary advantages

Compared with NRG-only, RGB+NRG improves DeepLabv3 by about 0.026, and U-Net improves U-Net by about 0.006.

Moderate Data Augmentation Improves generalization

Medium augmentation (flip + small rotation + color jitter) generally gives 3-6% IoU gain in U-Net/U-Net++.

B. Suboptimal Methods & Strategies

Traditional segmentation method

The Watershed/MeanShift algorithm performs poorly in complex tree crowns and shadows (mIoU < 0.03), and it is difficult to extract semantic boundaries.

Over-enhancement (Strong Aug)

Intensity augmentation introduces too much distortion (blurring, crop loss) and IoU drops.

Unimodal NRG-only

Only near-infrared information unable to capture the details of crown texture and shape, and the performance of all depth models in NRG-only is lower than that in RGB-only.

C. Limitations

Dataset size and diversity are limited

From the data set, only remote sensing images in a single region or season unable to guarantee the generalization of the model.

Real-time vs. accuracy tradeoff

U-net ++ has the best performance, but it has a large number of model parameters (26M) and high deployment cost.

Insufficient post-processing

Lack of fine-grained edge repair and detection, resulting in a small number of classification errors.

D. Future Work

Data Augmentation & Multi-Source Fusion

Extend the current best-performing models and training strategies to multiple, diverse datasets to ensure robust generalization.

Model Lightweighting & Acceleration

Port the architecture to lightweight backbones such as Mobile-Net or Ghost-Net, and apply pruning and quantization techniques to improve deployment efficiency on resource-constrained devices.

For each channel configuration, we will design three additional augmentation pipelines— “Light,” “Medium,” and “Strong” (including random cropping, contrast stretching, and Mixup)—and, under standard testing conditions on the given datasets, compare the model’s generalization performance across different datasets for each strategy.

Augmentation Strategy Optimization

Further tune augmentation parameters and introduce additional transformations on top of the current medium-strength channel to discover even more effective augmentation strategies.

By fine-tuning the parameters of the “medium” augmentation and introducing advanced transformations such as Cutout and HSV jitter, we further improve mIoU and accelerate overall convergence while keeping the original training settings intact. This provides a generalizable and reproducible augmentation strategy across different datasets and model architectures.

Semi-Supervised & Self-Supervised Learning

Leverage unlabeled or weakly labeled data through contrastive learning or generative adversarial methods, enabling the model to adapt more effectively to new environments without extensive manual annotation.

In remote-sensing segmentation tasks with extremely scarce labels, incorporating self-supervised pre-training on unlabeled data can substantially improve mIoU; moreover, this improvement remains consistent in label-scarce scenarios, demonstrating both generality and reproducibility.

Post-Processing & Fine-Grained Refinement

Integrate advanced post-processing techniques—such as conditional random fields (CRF), graph-based segmentation, and edge-refinement algorithms—to sharpen object boundaries, reduce fragmentation, and enhance overall segmentation quality.

REFERENCES

[1] J. Beucher and C. Meyer, “The morphological approach to segmentation: the watershed transformation,” in Mathematical Morphology in Image Processing, 1992, pp. 433–481.

[2] J. Fukunaga and L. D. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” IEEE Trans. Inf. Theory, vol. 21, no. 1, pp. 32–40, Jan. 1975.

[3] Ö. Çiçek, A. Abdulkadir, S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI), Athens, Greece, Oct. 2016, pp. 424–432.

[4] Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2016.

[5] J. W. Rouse, R. H. Haas, J. A. Schell, and D. W. Deering, “Monitoring vegetation systems in the Great Plains with ERTS,” in Third Earth Resources Technology Satellite-1 Symp., NASA SP-351 I, 1974, pp. 309–317.

[6] L. Vincent and P. Soille, “Watersheds in digital spaces: an efficient algorithm based on immersion simulations,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, no. 6, pp. 583–598, Jun. 1991.

[7] J. B. T. M. Roerdink and A. Meijster, “The watershed transform: Definitions, algorithms and parallelization strategies,” Fundam. Informaticae, vol. 41, no. 1–2, pp. 187–228, 2000.

[8] Comaniciu, D. & Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), pp.603–619.

[9] Cheng, Y., 1995. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), pp.790–799.

[10] L. Chen et al., “Rethinking Atrous Convolution for Semantic Image Segmentation,” in Proc. ECCV, 2018, pp. 18–34.

[11] S. Zhao, H. Zhu, and Y. Li, “Semantic Segmentation of Remote Sensing Imagery with DeepLabv3 and Multi-Scale Context,” Remote Sens., vol. 12, no. 5, p. 812, 2020.

[12] Ronneberger, O., Fischer, P. & Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI*, pp. 234–241.

[13] Yang, Q. et al., 2024. Improved U-Net based remote sensing segmentation with attention modules SimAM/CBAM. *arXiv* (Aug 2024).