Comparative Analysis of Deep Models for Dead Tree Detection in Multispectral Aerial Data

Jingyu Sun
z5525875@ad.unsw.edu.au

Mingqi Xu
z5532071@ad.unsw.edu.au

Yanbo Wang
z5603812@ad.unsw.edu.au

Yucheng Liu
z5621848@ad.unsw.edu.au

Linyang Yu
z5619437@ad.unsw.edu.au

Abstract

Monitoring forest health is critical for managing fire risks and ecological balance, particularly through the detection of standing dead trees (SDTs), which are highly flammable and indicative of environmental stress. In this study, we conduct a comparative analysis of four deep learning models—UNet, PSPNet, DeepLabV3, and Mask R-CNN—for semantic segmentation of dead trees using high-resolution multispectral aerial imagery (RGB + NIR). Using the expert-annotated DeadwoodUS2025 dataset, we train and evaluate each model under a unified protocol, applying data augmentation and adapting each architecture to support four-channel inputs. Performance is assessed via IoU, precision, recall, and F1-score metrics. Results show that PSPNet achieves the highest mIoU (0.54) and F1-score (0.65), excelling in structured scenes with clear contrast, while Mask R-CNN attains the highest recall (0.62) but suffers from false positives due to over-detection. UNet offers fast training and balanced performance, whereas DeepLabV3 presents a solid middle ground with consistent outputs across various scenarios. Through quantitative metrics and visual comparisons, we highlight each model's strengths and weaknesses, and provide practical guidelines for deploying deep segmentation models in large-scale forest health monitoring and early fire-warning systems.

Keywords—Dead Tree Detection, Multispectral Aerial Imagery, Semantic Segmentation Deep Learning Models

I. INTRODUCTION

Forest ecosystems are critical for preserving biodiversity, regulating climate, and supporting numerous species. Conversely, deteriorating forest health increases the risk of wildfires and other natural disasters—accumulated deadwood, in particular, acts as potent fuel for fast‑spreading fires. Consequently, there is an urgent need for scalable, automated methods to monitor forest health.

Conventional ground surveys are labor‑intensive, expensive, and frequently infeasible in remote or rugged terrain. In contrast, modern unmanned aerial systems and satellite platforms equipped with multispectral sensors can rapidly capture high‑resolution imagery over extensive areas. This wealth of data has driven the adoption of computer‑vision and automated image‑analysis techniques for ecological monitoring, enabling continuous, large‑scale assessment of forest conditions and early detection of fire hazards, while providing precise spatial information to guide restoration and management efforts.

Classic pixel and region based algorithms such as K‑means clustering, MeanShift, and Watershed struggle with the heterogeneous backgrounds, irregular tree geometries, and spectral ambiguities introduced by shadows and undergrowth in multispectral imagery. These methods often produce fragmented or incomplete segmentations, limiting their practical utility.

To overcome these challenges, we evaluate four convolutional‑neural‑network architectures: UNet, PSPNet, DeepLabV3, and Mask-RCNN that have demonstrated state‑of‑the‑art performance in semantic and instance segmentation. Our experiments use the DeadwoodUS2025 dataset (Ahishali et al., 2025), which comprises 444 expert‑annotated aerial images (R, G, B, NIR) from diverse U.S. forest environments, each paired with a binary mask of deadwood regions. All images are normalized and center‑cropped to a uniform spatial resolution before being fed into the models.

We train and validate each network under a consistent protocol, employing data‑augmentation techniques including random flips, rotations, and color jitter to bolster model robustness. Performance is quantified via Intersection over Union (IoU) and Dice coefficient metrics. By comparing how each architecture leverages spatial textures, spectral cues, receptive‑field expansion, and global context integration, we identify their respective strengths and limitations. Finally, we synthesize these findings into a set of best practices and deployment guidelines for deep‑learning–based deadwood segmentation in large‑scale forest monitoring and fire‑risk early‑warning systems.

II. LITERATURE REVIEW

Remote sensing technologies have increasingly enabled large-scale and automated analysis of natural environments, particularly forests. Aerial imagery, enhanced by near-infrared (NIR) and RGB spectral bands, provides rich visual cues to assess forest health. Among the key tasks in this domain is semantic segmentation, where each pixel is classified into categories such as tree, dead tree, or background. This is especially crucial for identifying

standing dead trees (SDTs), which often serve as early indicators of ecological stress and wildfire hazards. Traditional image processing techniques often fail in complex forest scenes with dense canopies, overlapping objects, and varying lighting conditions. Deep learning, particularly convolutional neural networks (CNNs), has revolutionized this task by learning spatial and contextual representations directly from raw data [5], [9], [12], [16].

A. Semantic Segmentation in Remote Sensing

Conventional forest segmentation relied heavily on handcrafted features and rule-based classifiers. Spectral vegetation indices like NDVI and EVI were used to detect vegetation stress or changes. However, such approaches often struggled to distinguish between live and dead trees, especially in high-resolution aerial imagery. With the rise of deep learning, particularly fully convolutional networks (FCNs), a paradigm shift occurred [8]. FCNs enable end-to-end training, pixel-wise prediction, and can be fine-tuned on relatively small annotated datasets [6]. In the domain of forest analysis, FCNs have been adopted for tasks such as canopy mapping, disease detection, and more recently, SDT segmentation [7], [12].

B. U-Net

U-Net [1] introduced a symmetric encoder-decoder architecture with skip connections that retain spatial details across layers. This enables high-resolution predictions, even in dense scenes with intricate boundaries. Its lightweight design and ability to converge with limited data have made it a go-to choice in biomedical and environmental applications [7]. In forest segmentation, U-Net has shown competitive performance in detecting tree crowns and delineating narrow objects like branches or dead stems [11].

C. DeepLabv3 and Atrous Convolution

DeepLabv3 [2] leverages atrous (dilated) convolution to control the receptive field without increasing computation, allowing for dense pixel labeling at multiple scales. The Atrous Spatial Pyramid Pooling (ASPP) module captures contextual information from various dilation rates simultaneously, which is valuable in forest imagery where object size and density vary across elevation and lighting conditions. Its robustness and modular design make it well-suited for large-scale semantic segmentation, including land cover analysis and dead tree detection in aerial images [2], [9], [12].

D. PSPNet for Global Context Encoding

PSPNet [3] augments FCNs by incorporating Pyramid Scene Parsing modules that extract global context through pooling at different spatial scales. This is critical in forest environments, where structural patterns of trees can be ambiguous without broader scene understanding.PSPNet disambiguates spatial layout by combining fine-scale information with coarse region-level information. It was shown to outperform baseline FCNs for segmentation of satellite imagery and has been applied to vegetation structure analysis [3], [9], [12].

E. Mask R-CNN for Instance-Level Segmentation

Mask R-CNN [4] introduces a segmentation branch to object detection networks so that object-level classification and pixel-wise prediction of a mask are possible. Unlike semantic segmentation, instance segmentation is capable of distinguishing between different instances of the same class—crucial to the counting or inspection of individual dead trees. Its Region Proposal Network (RPN) and RoIAlign layers offer spatial accuracy. Mask R-CNN is applied in forestry for detection of individual tree crowns and tree death estimation [10], [11] and is a potential candidate as an SDT detection system for sparse and cluttered forest images [4].

F. Related Work on Dead Tree Segmentation

Ahishali et al. [5] proposed ADA-Net, an attention-based contrastive learning model for domain adaptation with minimal generalization advantages under different forest conditions when small quantities of labeled data are used for training. ADA-Net offers a new state-of-the-art for SDT segmentation despite its reliance on advanced pretraining alongside adaptation strategies, which can add additional computational costs. Therefore, comparing it to lightweight models like U-Net or DeepLabv3 is still meaningful, especially when making trade-offs between accuracy and resources [5], [12].

III. METHODS

A. Data Preprocessing

The dataset consists of the RGB images, NRG (near-infrared) images and segmentation masks. For each object, three images were provided: RGB, NRG, and mask. To enhance the model's ability to extract spectral and structural information, we fused the RGB and NRG images into a 4-channel input: R, G, B, and NIR (channel 0 from NRG).

The ground truth masks are binary, where 1 indicates dead trees and 0 represents background.

Before training, all images were resized and normalized to standardize input dimensions and accelerate convergence. Data augmentation such as flipping and rotation was applied to increase model robustness.

B Task analyze and model selection

As a segmentation task, we considered several computer vision models, including YOLO, UNet, PSPNet, Mask R-CNN, DeepLabV3, ResNet, and Fast R-CNN. However, after a closer examination of the dataset, we determined that YOLO is not a suitable choice because it is primarily designed for 3-channel RGB training and the dataset lacks labeled segmentation annotations. Similarly, we excluded Fast R-CNN, as it is mainly intended for object detection and does not produce segmentation masks. Therefore, we narrowed our candidate models to UNet, PSPNet, Mask R-CNN.

B. Model Implementation

To address the semantic segmentation task of identifying dead trees from multispectral aerial imagery, we implemented and adapted four representative deep learning models: UNet, PSPNet, DeepLabV3, and Mask R-CNN. Each model was selected based on its architectural strengths and suitability for our specific context, rather than solely relying on theoretical performance in literature.

UNet: We chose UNet for its lightweight encoder-decoder structure with skip connections, which allows the preservation of spatial detail during upsampling. This is particularly effective in retaining fine-grained textures like thin branches or sparse tree canopies.

PSPNet: PSPNet's pyramid pooling module enables effective global context capture, which is critical in distinguishing dead tree clusters from complex forest backgrounds. We implemented this model to leverage its ability to interpret full-scene context at multiple scales.

DeepLabV3: DeepLabV3 supports multi-scale feature extraction via parallel atrous convolutions and has proven robust to irregular object boundaries. We utilized it to better handle variability in dead tree shape and size across terrains.

Mask R-CNN: Although primarily designed for instance segmentation, Mask R-CNN was adapted in our work due to its superior capability in generating fine object masks and separating overlapping tree regions, which are common in dense forests.

Each model was reconfigured to accept 4-channel multispectral inputs (RGB + NIR), and trained within the same pipeline for fair comparison. Minor modifications were made to input layers and decoder heads to adapt them to our specific dataset.

In addition to model architecture, we placed significant emphasis on the choice of loss functions, especially given the highly imbalanced nature of the dataset. Dead trees typically occupy only a small portion of each aerial image, making traditional loss functions less effective. To address this, we implemented Dice loss, which directly measures the overlap between predicted and actual masks. Its formulation makes it particularly well-suited for segmenting small, sparse objects, helping to stabilize training when positive samples are limited.

$$1-\frac{2\times TP}{2\times TP+FP+FN} \quad (1)$$

To further improve performance, especially in cases where background pixels dominate, we incorporated Focal loss into our training strategy. Unlike standard cross-entropy, Focal loss reduces the contribution of well-classified regions, encouraging the model to focus more on difficult or ambiguous pixels. This property is essential in our task, where subtle cues are often the only indicators of dead trees, and most regions of the image consist of visually dominant but less informative background[19].

$$-\alpha\times(1-p_t)^{\gamma}\times\log(p_t) \quad (2)$$

Lastly, we explored Tversky loss, a generalization of Dice that introduces controllable weights for false positives and false negatives. This flexibility is valuable when recall is a priority—such as in forest monitoring tasks where missing a dead tree may be more critical than a false detection. By adjusting the balance between precision and recall, we can slightly adjust the models by Tversky loss which can improve the sensitivity of detecting small, partially occluded, or visually subtle targets. These loss functions which we chose provided complementary benefits and played a critical role in achieving more reliable segmentation results at diverse forest conditions.

$$1-\frac{TP}{TP+\alpha FP+\beta FN} \quad (3)$$

In order to solve class imbalance problems and improve the segmentation of sparse targets, we prepared specific loss functions which can be suited to model structure for each model. The UNet and PSPNet models were trained by the combination of Dice loss and Binary Cross-Entropy which can balance the spatial overlap optimization with pixel-level precision. Final results proved its effectiveness in capturing fine details of dead trees and maintaining training stability. DeepLabV3 and Mask R-CNN were optimized by Focal loss and Tversky loss because they helped suppress dominant background pixels and offered flexible control of false positives and negatives. This combination enhanced the models' sensitivity to small or occluded trees and improved overall performance in complex forest environments.

C. Unet

The model we use is the vanilla U-Net, a straightforward but reliable choice for pixel-level segmentation. It follows the familiar "U" shape: an encoder that shrinks the feature maps, a bottleneck at the bottom, and a decoder that brings everything back up to the original resolution. As the image moves down the left side, each block applies two 3 × 3 convolutions with ReLU, then halves the spatial size, letting the network learn richer and richer semantics. The deepest layer holds the most abstract view of the scene.

On the way up, transposed convolutions double the resolution at every step, and each decoder block is fed the matching feature map from the encoder. Those skip connections merge coarse context with fine detail, so the model recovers sharp edges and small structures that would otherwise vanish. A final 1 × 1 convolution turns the last feature map into the desired number of output channels, and a sigmoid (or soft-max for multi-class jobs) converts raw scores to probabilities.

U-Net accepts any reasonable channel count—four channels in our RGB + NIR case—and returns a mask of the same height and width as the input. The network is light on parameters, easy to train, and has a proven track record in medical imaging, remote sensing, and other domains where clean, boundary-aware masks matter.

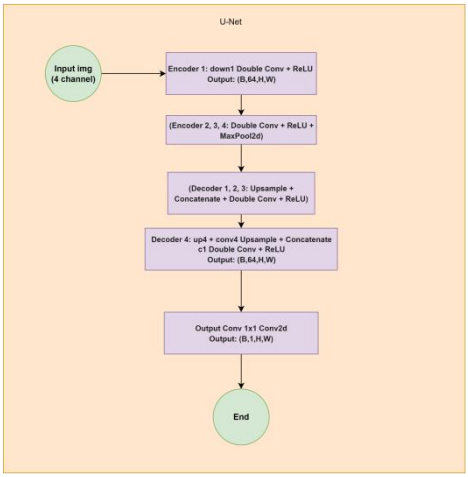

figure 1: U-Net Architecture with 4-Channel Input

D. PSPNet

The model we use is the Pyramid Scene Parsing Network (PSPNet), a robust and innovative choice for pixel-level segmentation that builds on advanced contextual understanding. It follows a distinctive structure: a deep convolutional backbone that extracts hierarchical features, a pyramid pooling module at the core, and a final upsampling stage that restores the original resolution. As the image passes through the backbone—typically a ResNet—the network applies a series of convolutional layers, progressively reducing spatial dimensions while capturing increasingly complex semantics, culminating in a deep feature representation.

At the heart of PSPNet lies the pyramid pooling module, which aggregates global context by pooling features at multiple scales (e.g., 1×1, 2×2, 3×3, 6×6 grids). This multi-level pooling enriches the model with a broad view of the scene, capturing both local details and global relationships. On the way up, the upsampled features are fused with the backbone's high-resolution maps through skip connections, blending coarse contextual information with fine-grained details to ensure precise boundaries and structures are preserved. A final 1×1 convolution transforms the enriched feature map into the desired number of output channels, and a softmax (for multi-class tasks) converts the scores into probability maps.

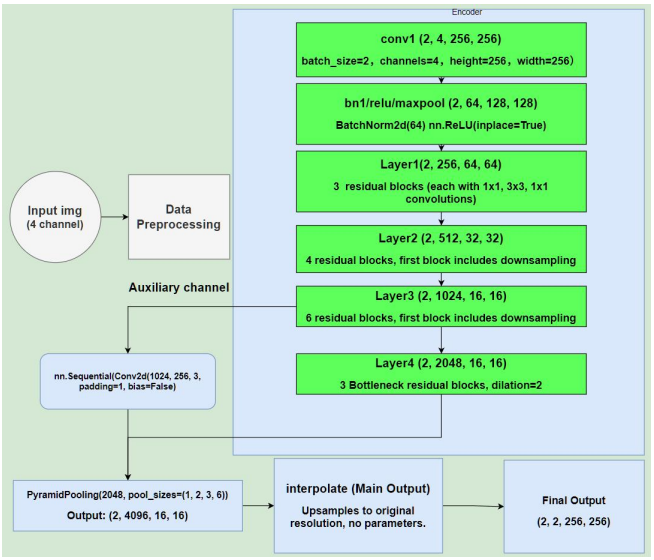

figure 2: Resnet50 + PSPNet constructure

PSPNet handles any reasonable channel count—four channels in our RGB + NIR case—and produces a mask matching the input's height and width. The network, while parameter-rich due to its deep backbone, is highly effective for tasks requiring strong contextual awareness, with a strong track record in scene parsing, remote sensing, and urban segmentation where understanding the full scene context is key.

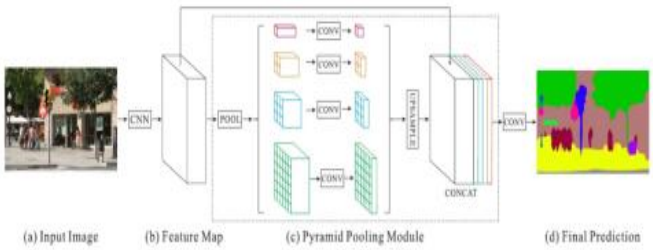

figure 3: Pyramid architecture

E. DeepLabV3

We choose DeepLab V3 as our segmentation backbone because it delivers state-of-the-art accuracy with modest computational overhead and thanks to its Atrous Spatial Pyramid Pooling (ASPP) head which captures context at multiple scales when an essential trait when small target pixels are scattered across vast background areas in remote-sensing images.

We consider the CBAM to improve our model because as each channel of a feature map is considered as a feature detector , channel attention focuses on 'what' is meaningful given an input image. (Woo & Park, 2018, #)in which we insert a Convolutional Block Attention Module (CBAM) at the conv4_x stage. We place attention only here to reduce the influence of our attention model in earlier layers which mostly encode low-level textures that in our highly imbalanced dataset belong almost entirely to the background, so adding CBAM too soon would risk highlighting noise instead of signal. Operating on deeper, semantically richer features allows CBAM to spotlight genuine deadwood cues while suppressing irrelevant patterns. and we also added a resblock in the previous layer

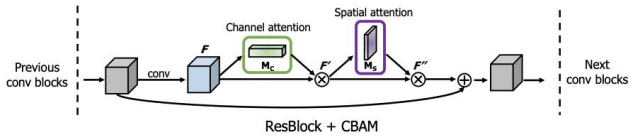

figure 4: Structure of ResBlock with CBAM

The backbone output then feeds the ASPP block. By running several parallel atrous (dilated) convolutions with different rates (e.g., 6, 12, 18) and adding a global-average-pooling branch, ASPP gathers fine details and image-level context in one representation[14],[17]. This design compensates for the sparse effective receptive field that can arise at large dilation rates and ensures robust detection of objects at many scales.

Finally, a 1 × 1 classifier converts the ASPP features into pixel-wise logits, and the model is trained end-to-end. In combination, late-stage attention and multi-scale context give the network the precision to delineate small targets while remaining resilient to overwhelming background clutter.

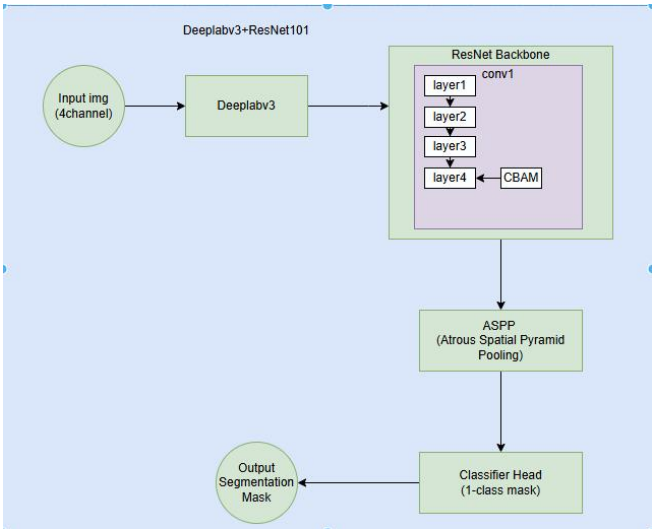

figure 5: DeepLab3 + ResNet101 constructure

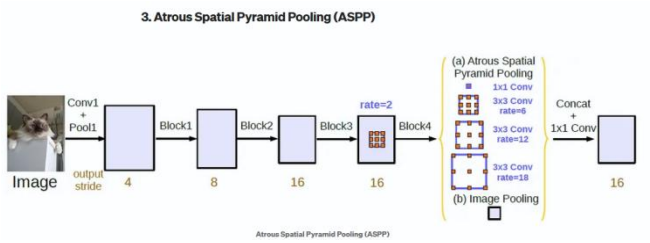

figure 6: ASPP architecture

F. Mask R-CNN

Our team chose the Mask R-CNN model because of its well-established two-stage approach which seamlessly combines object detection and instance segmentation. Specifically, we used a ResNet101 + Feature Pyramid Network (FPN) backbone to process four-channel input (RGB + NIR) and extract rich multi-scale feature maps. The FPN fuses the outputs of different ResNet stages into a unified pyramid which preserves both fine details and high-level context, crucial for accurately localizing redundant regions of varying sizes. The ResNet-FPN backbone provides excellent multi-scale feature extraction, making it highly effective for both Faster R-CNN and Mask R-CNN [13], [18]. The Region Proposal Network (RPN) then slides anchors across the pyramid to generate region proposals, and ROIAlign precisely crops these proposals that preserve spatial fidelity which avoids the quantization issues of earlier ROI pooling methods(figure7). Finally, two specialized heads process each ROI:

Detection Head: optimizes bounding box coordinates and predicts class labels.

Mask Head: generates high-resolution, per-instance segmentation masks.

This architecture delivers strong performance across a wide range of scales and produces pixel-accurate masks which make it well-suited for our deadwood detection task.

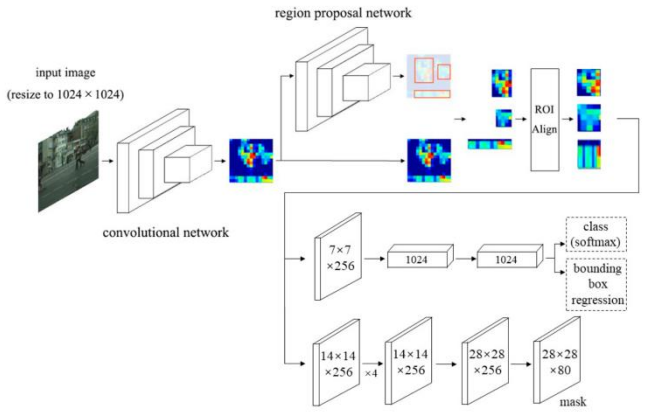

figure 7: ROIAlign network architecture

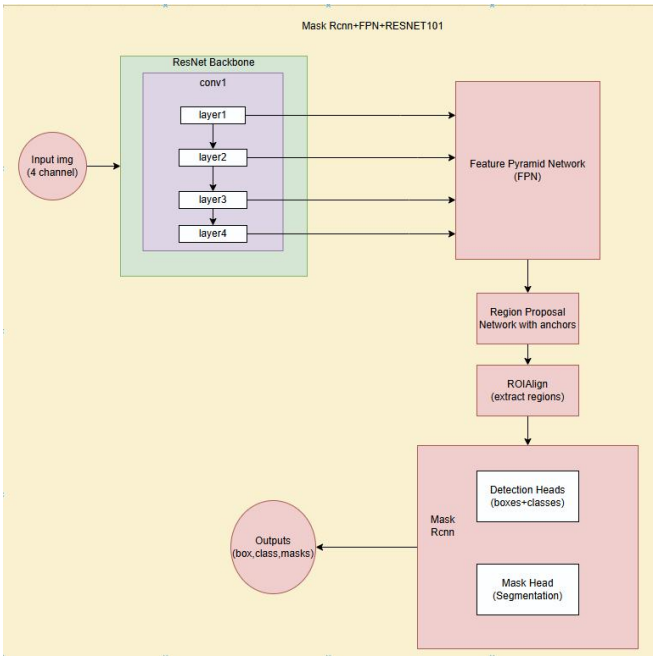

figure 8: Rcnn Mask with FPN Resnet backbone

G. Evaluation Metrics

To evaluate model performance for the segmentation of dead trees from aerial multispectral imagery, we employed four commonly used evaluation metrics:

IoU: Measures the overlap between predicted segmentation and ground truth, widely used in semantic segmentation tasks.

Precision: Indicates the proportion of predicted dead tree pixels that are truly dead trees.

Recall: Reflects the proportion of actual dead tree pixels that were correctly identified.

F1-score: The harmonic mean of precision and recall, balancing both false positives and false negatives.

These metrics were computed on the validation set at the end of each epoch to monitor the model's learning progress. Evaluation curves (e.g., Precision-Recall, IoU trends) were also plotted to visualize convergence behavior and stability.

IV. EXPERIMENTAL RESULTS

During the experiment, Our team members developed and trained all models in Python 3.10+ and PyTorch 2.6+ configure environments. For UNet, PSPNet, DeepLabV3, and Mask R-CNN deep learning models, we uniformly used the Adam optimizer with an initial learning rate of 1e–4 and a batch size of 8. When validation loss failed to decrease learning rate decay and early stopping were automatically triggered to prevent overfitting. Each model was trained for 100 epochs, and performance was evaluated on the validation set after several epochs, with the best weights saved.

During the testing phase, the optimal weights were loaded into the local environment and the final segmentation mask was generated using a combination of sliding window slicing and full-image prediction. The high-resolution image was first segmented into overlapping patches for separate predictions, followed by a single inference on the entire image. Finally, the predicted probabilities at the same pixel location were averaged and thresholded, balancing local details with global semantics.

In the performance evaluation, in addition to the average intersection over union (IoU), pixel-level accuracy (Precision), recall (Recall), and F1-Score were calculated to comprehensively reflect the model's ability to detect dead wood areas and the number of missed detections. We also plotted a confusion matrix to deeply analyze the false detection and missed detection patterns of each model in different scenarios (such as dense branches and leaves, shadows, etc.). Through a comprehensive comparison of quantitative indicators and qualitative visualization results, the advantages and limitations of the four networks in terms of accuracy, robustness, and instance segmentation capabilities were clarified, providing feasible technical suggestions for the subsequent model selection and deployment in large-scale forest health monitoring systems.

The final experimental results are shown in the table below:

TABLE I: Metrics Comparison

	mIoU	Precision	Recall	F1-score
UNet	**0.37**	**0.52**	**0.57**	**0.54**
PSPNet	**0.54**	**0.73**	**0.39**	**0.65**
DeepLab V3	**0.40**	**0.61**	**0.54**	**0.57**
Mask-RCNN	**0.34**	**0.43**	**0.62**	**0.50**

Among the four models, UNet comes in as the most lightweight option. It achieves an mIoU of 0.37, precision of 0.52, and recall of 0.57, resulting in an F1 score of 0.54. Its skip connections help retain low-level details, which explains the relatively high recall. However, due to its straightforward structure and lack of a multi-scale module, precision remains modest, and its segmentation boundaries are sometimes soft. UNet is ideal when working with limited hardware or during early-stage prototyping, where fast iteration matters more than top-tier accuracy.

PSPNet delivers the strongest performance in terms of segmentation accuracy. It tops the table with an mIoU of 0.54, precision of 0.73, and an F1 score of 0.65. These gains come mainly from its pyramid pooling module, which effectively captures global context and filters out noisy pixels. However, the same pooling mechanism tends to suppress smaller or less prominent features, causing its recall to drop to 0.39, the lowest among the group. When in scenarios where false positives are costly this trade-off makes PSPNet ideal for field validation tasks, but when in scenarios where requiring detection of fine-grained targets, it may underperform.

DeepLab V3 offers balanced performance with mIoU of 0.40, precision of 0.61, recall of 0.54, and F1 score of 0.57. This model uses the amitrous convolutions and the ASPP module to effectively balance local detail and global context. In practical situation, DeepLab V3 is suitable for general-purpose segmentation tasks because it can detects main structures well and unable over-firing on irrelevant regions.When we need reliable results and don't want with intensive hyperparameters adjusting or hardware demands, DeepLab V3 is ideal choice for that tasks

Mask R-CNN focuses on recall, which it achieves well at 0.62, the highest in the group. This is thanks to its region proposal network and feature pyramid structure, they cast a wide net across the image. But this structure also has disadvantages, first aggressive detection leads to more false positives which pulls this model precision down to 0.43, with mIoU of 0.34 and F1 score of 0.50. Otherwise, the two-stage architecture increases processing time and memory usage. According to these features, Mask R-CNN is best suited for safety-critical applications where missing a detection is more problematic than generating extra ones.

In summarize, UNet is simple and the most efficient option, with minimal resource usage it can offer decent recall and speed, this feature make UNet suitable for quick prototyping or low-end hardware. PSPNet achieves the highest mIoU and precision, Because of global context awareness, if avoiding false positives is critical, this model is best choice. DeepLab V3 stands out for its balanced performance, it can maintain reliable results with little tuning, it's suitable for general-purpose use. Mask R-CNN leads in recall due to its region proposal mechanism but suffers from low precision and higher computational cost. The choice ultimately depends on whether your task prioritizes catching every object or avoiding extra detections.

To more directly demonstrate the performance of each model in real-world deadwood identification, we further present some of the more successful outputs of each model in this task. These outputs allow us to compare the performance differences and strengths and weaknesses of each algorithm model.

After comparing the visual outputs of the four segmentation models, we can observe that each exhibits its own tendencies in detecting deadwood and generating masks. UNet, thanks to its encoder-decoder structure with skip connections, performs relatively well in preserving object shapes, especially smaller ones, though its predicted

edges tend to be somewhat fuzzy. PSPNet, with its pyramid pooling module, seems better at separating objects from the background and capturing broader context, but it occasionally misses fine-grained targets. DeepLab V3, using dilated convolutions to widen its receptive field, offers strong results on more complete structures and often scores high on IoU, yet it may overlook fragmented or occluded regions. Mask R-CNN stands out for its precise localization and object boundary matching, making it a good fit when instance-level distinction is required. In conclusion, these images reflect each model's different architectural biases that some focus more on detail, others on global context or object-level accuracy. So we should consider these differences when we select a model for specific tasks.

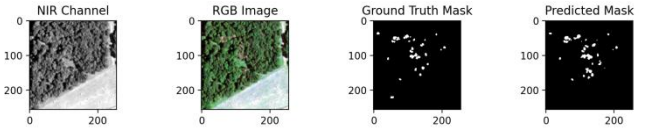

Fig. 9. UNet Best Example Output

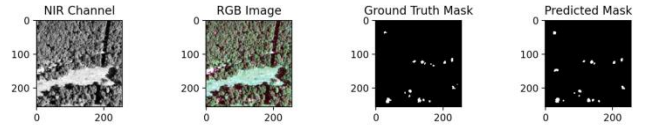

Fig. 10. PSPnet Best Example Output

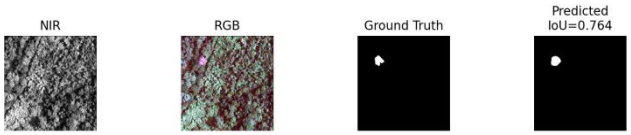

Fig. 11. DeepLabV3 Best Example Output

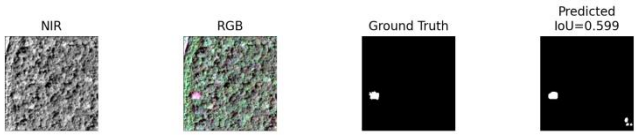

Fig. 12. Mask-RCNN Best Example Output

After comparing the worst-case predictions from the four models, we find that these images reveal several limitations common across these segmentation models. First, when sparse or small-sized objects all models struggle significantly that often cause either complete omission or excessive false positives. This reveals that downsampling operations and lack of fine-detail refinement modules contribute to the loss of small targets. Second, certain predictions achieve an Intersection over Union (IoU) of zero, indicating total failure to identify any ground truth regions. This may be attributed to complex backgrounds, poor contrast, or insufficient contextual learning. Moreover, several predicted masks highlight regions with high confidence that do not correspond to any annotated objects, suggesting misinterpretation of background textures as targets. In cases where NIR or RGB inputs are noisy or vary from training conditions, models like DeepLabv3 fail to generalize, emphasizing their sensitivity to input distribution shifts. Overall, these failure cases point to weaknesses in detecting fine structures under challenging conditions, highlighting the need for enhanced spatial attention mechanisms, multi-scale feature integration, and more diverse training data to improve robustness and generalization.

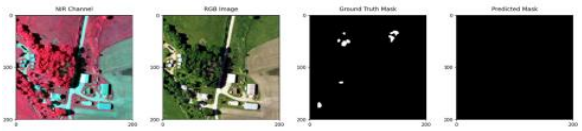

Fig. 13. PSPNet Worst Example Output

Fig. 14. DeepLabV3 Worst Example Output

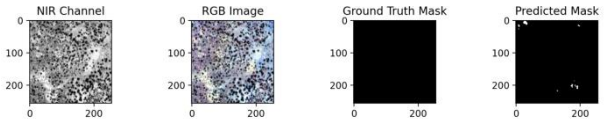

Fig. 15. UNet Worst Example Output

Fig. 16. Mask-RCNN Worst Example Output

V. DISCUSSION

A. Comparative Performance of Segmentation Models

PSPNet still leads the metric board with an mIoU of 0.54 and a precision of 0.73. Its pyramid-pooling block absorbs global context, wiping away stray pixels across the frame. The trade-off shows up in recall: at just 0.39, many very small or low-contrast trees disappear in the pooling stage. PSPNet therefore fits situations where a few misses are tolerable but extra false alarms would bog down field crews.

Mask R-CNN sits at the opposite end. Dense region proposals push recall to 0.62—hardly any tree slips past—but bright roofs, bare soil, and glare are swept up too, dropping precision to 0.43 and mIoU to 0.34. A quick area-or-shape filter removes part of this noise, yet the model remains computationally heavy if GPU hours are tight.

DeepLab-V3 falls squarely in the middle. With atrous convolutions, an ASPP head, and a Tversky loss already in place, it handles detail and context in equal measure, ending up with balanced numbers (mIoU 0.40, precision 0.61, recall 0.54). In practice it captures most trunks, skips a few slender branches, and is less dazzled by glare than Mask R-CNN. When both error types matter and time for tuning is short, DeepLab-V3 is the safe default.

U-Net remains the lightweight baseline. Skip links preserve edges well enough to hit a recall of 0.57, but without an explicit multi-scale head its precision stalls at 0.52 and mIoU at 0.37. For rapid prototyping or GPU-limited deployments, U-Net still earns its keep—especially if a small post-filter tidies soft boundaries.

Looking ahead, a handful of low-cost tweaks could lift all four models. Lowering the output stride or adding a light super-resolution layer should help every network recover

sub-pixel trees. Feeding the learner more "glare-but-no-tree" tiles will train it to ignore shiny clutter. Mask R-CNN will tighten up with finer anchors; PSPNet and DeepLab-V3 can benefit from tuning the Tversky α / β weights or switching to a focal-Tversky variant that emphasises hard pixels without sacrificing stability. Finally, a simple blob filter—dropping regions below a minimum area or with implausible aspect ratios—removes most stray detections. An ensemble that merges PSPNet's sharp masks with Mask R-CNN's wide net could raise the overall F1 without a full retrain, giving survey teams fewer misses and fewer false calls on the next pass.

B. *Model Discussion*

a) Mask-RCNN:

we compared the performance metric of each model and we went into deeper analyze of each hyperparameter influence on each model performance across the mode we conclude the most sensitive parameter for each model and look into the advantage and limitation of each model

Mask R-CNN exhibits notable advantages in instance-level segmentation tasks, particularly when applied to minority-class object detection such as standing dead trees. One of its key strengths lies in its instance-level prediction mechanism. Instead of assigning semantic labels to each pixel, Mask R-CNN outputs separate binary masks for each object instance, enabling more accurate detection of small, isolated targets. Furthermore, its Region Proposal Network (RPN) component effectively narrows the search space by focusing on regions likely to contain objects, which significantly improves recall—especially for small-scale or low-contrast features. Additionally, the backbone network, integrated with a Feature Pyramid Network (FPN), facilitates multi-scale feature extraction. This enhances the model's ability to preserve spatial details and detect objects across a variety of scales.

However, these strengths are accompanied by several limitations. Mask R-CNN has a relatively high computational cost due to its multi-stage architecture, which includes RPN, ROIAlign, classification heads, and mask heads. Training and inference are more resource-intensive than in typical semantic segmentation networks. Moreover, as the model is inherently designed for instance segmentation, converting its outputs into semantic masks requires an additional processing step to merge overlapping instances. This step can introduce ambiguity at object boundaries and potentially degrade final segmentation accuracy.

Mask R-CNN also demonstrated considerable sensitivity to certain hyperparameters in our experiments. Specifically, increasing the input image resolution led to improved feature retention in the FPN, while reducing the predicted mask pixel size allowed the RPN to produce more accurate region proposals. These two modifications jointly resulted in an approximate IoU improvement of 0.1. This highlights the model's strong dependence on spatial detail preservation, as both RPN and FPN performance directly benefit from higher-resolution inputs and finer mask granularity.

b) DeepLabV3:

DeepLabV3 operates as a dense semantic segmentation model that performs per-pixel classification across the entire image, making it naturally aligned with the goals of semantic segmentation tasks such as standing dead tree detection. One of its architectural highlights is the use of Atrous Spatial Pyramid Pooling (ASPP), which captures contextual information at multiple scales without excessive downsampling. This design can improve its ability to detect small and spatially sparse objects within complex backgrounds, because it can balance global awareness and local detail..

On the other hand, DeepLabV3 exhibits limitations related to its dependence on spatial resolution. Although ASPP enables broader context aggregation, it still relies on dilated convolutions which are applied to downsampled feature maps. Small or thin objects may be lost entirely during downsampling if resolution is lower which leads to reduced recall. This makes the model particularly sensitive to the input image size, as smaller objects risk becoming indistinguishable from the background. Additionally, unlike architectures such as UNet, which utilize skip connections, or Mask R-CNN, which applies ROI-based refinement, DeepLabV3 lacks dedicated refinement modules which often results in blurred or imprecise object boundaries, especially when the scenes with intricate textures or objects of irregular shape. Consequently, the model may produce lower IoU scores in datasets containing small, fine-grained structures.

Our experiments further confirmed that DeepLabV3's segmentation accuracy is highly dependent on the resolution of the input data. Higher-resolution inputs tend to preserve critical spatial details, thereby improving small-object detection and overall IoU performance. However, this improvement comes at the cost of increased memory usage and longer training times. In practice, this trade-off necessitates careful balancing between input size, batch size, and hardware constraints, particularly when working with imbalanced datasets where minority classes are defined by subtle or small visual cues.

c) Unet:

In this project, we focus on the implementation and enhancement of the standard U-Net segmentation model for remote sensing applications. U-Net's symmetric U-shaped architecture, consisting of an encoder, bottleneck, and decoder, has consistently shown strong performance for pixel-level segmentation. The encoder progressively extracts high-level semantic features through double convolutions and pooling, while the bottleneck further enriches this information. The decoder then fuses deep semantic information with shallow spatial details via upsampling and skip connections, effectively restoring spatial resolution and improving segmentation boundary accuracy. This intuitive design, along with support for multi-channel inputs, makes U-Net highly adaptable to complex data such as multispectral remote sensing imagery, as well as robust to limited data scenarios.

Recognizing the rapid progress in segmentation methods, our project compares and benchmarks U-Net against a range of modern architectures, including DeepLab, PSPNet , and Mask R-CNN, as well as more recent approaches that address domain adaptation and multimodal data fusion, such as ADA-Net and work by Audebert et al.. These comparisons highlight both the strengths and the limitations of U-Net in challenging settings like small object segmentation and heterogeneous landscape analysis.

To further improve segmentation accuracy and model robustness, we introduce several engineering optimizations tailored for remote sensing. Specifically, we expand the input to four channels (RGB+NIR), apply standardized preprocessing steps, and employ synchronized data augmentation to enhance model generalization. Our loss strategy combines multiple objectives—such as BCE+Dice, Focal, and Tversky losses—to better handle class imbalance and uncertain boundaries. The training system also included the functions of real-time metric monitoring, flexible hardware adaptation, and automatic selection of the best-performing model. This system ensures both training reliability and reproducibility.

Finally, because transformer-based models are beginning to display exceptional performance in vision tasks, we are starting to find opportunities to integrate these advances into future segmentation pipelines for large-scale remote sensing analysis. Overall, our optimized U-Net framework such as U-Net ++ which delivers notable improvements in accuracy, stability, and practical usability and underscore its value for real-world applications in remote sensing image segmentation.

d) PSPNet:

PSPNet has demonstrated promising performance in semantic segmentation tasks, especially when dealing with scenarios that require understanding the global structure of a scene. For instance, we found it quite effective in identifying standing dead trees across different forest environments. One of the main strengths of PSPNet is its Pyramid Pooling Module (PPM), which captures contextual information at multiple spatial scales (1×1, 2×2, 3×3, and 6×6). By combining local and global features, the model becomes more robust to variations in object scale and complex background textures. In our experiments, this design helped the network handle diverse spatial patterns more effectively. Additionally, the use of an auxiliary loss during training improved gradient flow and made convergence easier, especially for deeper networks.

However, PSPNet also comes with some drawbacks. Due to the large pooling operations in the PPM, some fine-grained spatial details can be lost, which often leads to blurry or inaccurate object boundaries—particularly for objects with irregular shapes. This limitation was evident in our results, especially when compared with models like U-Net that preserve more spatial information using skip connections. Also, PSPNet is relatively computationally expensive. Its multi-scale pooling and fusion layers require more memory and processing power, which might not be ideal for real-time applications or resource-limited environments.

During training, we also observed that PSPNet is quite sensitive to hyperparameter settings. The learning rate, in particular, had a strong impact on training stability. An inappropriate value could lead to unstable convergence or noisy segmentation outputs. To address this, we experimented with adjusting the pooling sizes in the PPM to match the typical scale of dead trees in our dataset and applied a cosine annealing learning rate scheduler. As a result, we saw about a 0.05 improvement in mIoU on the validation set. This suggests that PSPNet's performance depends heavily on both the choice of hyperparameters and the alignment between pooling scales and object characteristics.

Overall, PSPNet is a good fit for applications that prioritize global scene understanding, such as landscape analysis, urban planning, or environmental monitoring. That said, for tasks requiring precise boundary segmentation or real-time performance, lighter models like U-Net or DeepLab might be more practical. As for its sensitivity to learning rate, using adaptive scheduling strategies and tuning the model based on input resolution and object size can help stabilize training and improve performance.

VI. CONCLUSION

In this project, we implemented and compared four mainstream deep learning models—UNet, PSPNet, DeepLabV3, and Mask R-CNN—for semantic segmentation of dead tree areas using aerial multispectral imagery. Based on commonly used evaluation metrics such as IoU, precision, recall, and F1-score, each model demonstrated varying levels of segmentation performance.

UNet showed strong baseline performance and achieved good results on average, likely due to its skip connections preserving spatial resolution. PSPNet's ability to capture global contextual features contributed to its stable performance across different regions. DeepLabV3 benefited from multi-scale feature extraction and was relatively robust. Mask R-CNN, while originally designed for instance segmentation, still provided competitive pixel-level results after adaptation.

However, given that our current evaluation is limited to standard quantitative metrics, deeper insights into failure patterns—such as errors under occlusion, varying lighting, or texture confusion—could not be thoroughly investigated. Moreover, limitations of the dataset, including class imbalance, limited annotations, and potential image noise, may have affected the reliability of the metrics.

In future work, we plan to expand our analysis by incorporating qualitative evaluations (e.g., visualizing prediction masks), assessing computational efficiency, and conducting ablation studies. Additionally, integrating domain-specific data augmentation techniques or transformer-based architectures may further enhance segmentation performance[20] for forest health monitoring.

VII. REFERENCES

[1] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In MICCAI. https://arxiv.org/abs/1505.04597

[2] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. (2017). DeepLab: Semantic image segmentation with deep convolutional nets, Watrous convolution, and fully connected CRFs. IEEE TPAMI, 40(4), 834–848.

[3] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In CVPR.

[4] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. In ICCV.

[5] Ahishali, M., Rahman, A. U., Heinaro, E., & Junttila, S. (2025). ADA-Net: Attention-guided domain adaptation with contrastive learning for standing dead tree segmentation. arXiv:2504.04271.

[6] Zhang, Y., Du, B., & Zhang, L. (2016). A Coarse-to-Fine Framework for Remote Sensing Image Segmentation With Sparse Representation and Conditional Random Field. IEEE TGRS, 54(7), 4094–4107.

[7] Kampffmeyer, M., Salberg, A. B., & Jenssen, R. (2016). Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks. In CVPR.

[8] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In CVPR.

[9] Audebert, N., Le Saux, B., & Lefevre, S. (2018). Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS Journal of Photogrammetry and Remote Sensing, 140, 20–32.

[10] Silva, C. A., Hudak, A. T., Vierling, L. A., et al. (2016). Imputation of Individual Longleaf Pine (Pinus palustris Mill.) Tree Attributes from Field and LiDAR Data. Canadian Journal of Remote Sensing, 42(5), 554–573.

[11] Dalponte, M., Ørka, H. O., Ene, L. T., et al. (2014). Tree Crown Delineation and Tree Species Classification in Boreal Forests Using Airborne Laser Scanning. IEEE TGRS, 52(9), 5297–5310.

[12] Ma, L., Liu, Y., Zhang, X., et al. (2019). Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing, 152, 166–177.

[13] T.-Y. Lin et al., “Feature Pyramid Networks for Object Detection,” in *Proc. CVPR*, 2017.

[14] K. He et al., “Mask R-CNN,” in Proc. ICCV, 2017.

[15] Woo, S., & Park, J. (2018). CBAM: Convolutional Block Attention Module. Computer Vision and Pattern Recognition (cs.CV).

[16] Zhang, C., & Xu, W. (2018). "Deep Learning in Remote Sensing Applications: A Meta-Analysis and Review." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(11), 4296–4310.

[17] Li, Y., Chen, Y., Wang, N., & Zhang, Z. (2019). "Scene-Aware Attention Network for Remote Sensing Image Segmentation." IEEE Transactions on Geoscience and Remote Sensing, 57(7), 5070–5083.

[18] Marmanis, D., Datcu, M., Esch, T., & Stilla, U. (2016). "Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks." IEEE Geoscience and Remote Sensing Letters, 13(1), 105–109.

[19] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2020). "Focal Loss for Dense Object Detection." IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 318–327.

[20] Yuan, Y. & Wang, J. (2021). "Multimodal Remote Sensing Image Segmentation: A Survey." Information Fusion, 73, 34–54.