Feature Fusion to Improve YOLOv8 for Segmenting and Classifying Aerial Images of Tree Crowns

Ziyi Sun, Bing Xue, *Fellow, IEEE*, Mengjie Zhang, *Fellow, IEEE*, and Jan Schindler

***Abstract*—Instance segmentation techniques based on convolutional neural networks (CNNs) is a vital tool for accurately identifying and segmenting individual tree crowns, which plays an essential role in environmental monitoring and forest management. In varied rural landscapes, canopy imagery often includes a mix of tiny, small, and medium tree objects scattered across diverse terrains, from standalone trees to densely clustered forest stands. This variability poses significant challenges to traditional instance segmentation methods. To achieve this, we introduce a new method named YOLOv8-FF, which incorporates a feature fusion (FF) technique based on the YOLOv8 architecture. We first design a network architecture based on YOLOv8 that is optimized for the characteristics of our dataset, enabling effective segmentation of densely distributed tiny and small tree crowns. Moreover, YOLOv8-FF incorporates a FF mechanism that includes both cross-scale and same-scale fusion methods, enhancing the model's ability to integrate information across different layers and scales, thereby improving segmentation performance. We incorporate Sparse Large Kernel Network, whose large convolution kernel can effectively extract key features, helping the model capture richer and deeper global information in the image. Experimental results on the tree crown dataset demonstrate that YOLOv8-FF outperforms several recent peer competitors, making it a promising tool for accurate and efficient tree crown instance segmentation.**

***Index Terms*—Instance segmentation, remote sensing, tree crowns, tree species, YOLOv8.**

I. Introduction

TREE crown segmentation plays an essential role in environmental monitoring and forest management [1], facilitating the implementation of more precise and effective management strategies that significantly contribute to conservation efforts and sustainable resource utilization. As artificial intelligence revolutionizes instance segmentation in remote sensing imagery, the development and adaptation of new Artificial intelligence (AI) techniques in these areas become increasingly vital for environmental monitoring, particularly for forest management.

Received 13 October 2024; revised 20 December 2024, 11 March 2025, and 15 May 2025; accepted 19 May 2025. Date of publication 5 June 2025; date of current version 23 June 2025. This work was supported by the New Zealand Ministry of Business, Innovation and Employment under Grant C09X1923 (Catalyst: Strategic Fund). *(Corresponding author: Ziyi Sun.)*

Ziyi Sun, Bing Xue, and Mengjie Zhang are with the Center for Data Science and Artificial Intelligence & School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand (e-mail: ziyi.sun@ecs.vuw.ac.nz; bing.xue@ecs.vuw.ac.nz; mengjie.zhang@ecs.vuw.ac.nz).

Jan Schindler is with Manaaki Whenua-Landcare Research, Wellington 6140, New Zealand (e-mail: schindlerj@landcareresearch.co.nz).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSTARS.2025.3576780, provided by the authors.

Digital Object Identifier 10.1109/JSTARS.2025.3576780

AI has significantly impacted computer vision, enabling machines to perceive, understand, and interpret visual information, revolutionizing various fields including image classification, object detection, and image segmentation. Instance segmentation in remote sensing imagery is also experiencing rapid advancements thanks to the emergence of artificial intelligence, particularly deep learning techniques. This task plays a crucial role across various domains such as traffic monitoring [2], marine management [3], and agriculture [4]. The main goal of instance segmentation is to identify and segment individual objects. It surpasses pixel-level classification by discerning between different object instances within the same category, assigning a unique instance ID to each object. Instance segmentation of individual tree crowns within the realm of remote sensing represents a significant and practical application, contributing to tasks like forest management [1], carbon storage estimation [5], biodiversity modeling [6], canopy closure estimation [7], ecosystem service modeling [8], and forest health description [9]. This article focuses on achieving instance segmentation of tree crowns in aerial imagery, specifically detecting and delineating individual tree crowns while classifying the tree species in a rural hill-country area in the Greater Wellington region, Aotearoa New Zealand.

Nevertheless, achieving instance segmentation of individual tree crowns in aerial imagery presents significant challenges, primarily stemming from three factors: stand density, crown characteristics, and background conditions [10]. First, dense tree stands often result in canopy overlap and shadow casting, complicating the detection of individual canopy edges. Second, the intricate characteristics of tree crowns, including variations in size, color, shape, and texture, pose difficulties for accurate segmentation, especially considering the limited detail in aerial images. This challenge is further exacerbated by the presence of numerous small objects within the images, with some tiny tree crowns occupying only a few pixels. Last, the surrounding background can impact tree crown detection and segmentation effectiveness. Particularly, short objects resembling trees, such as shrubs, weeds, and grass, may be misidentified as trees (i.e., false positives) due to the absence of height distinction in aerial imagery.

In recent years, deep convolutional neural networks (CNNs) based algorithms have improved by leaps and bounds compared to traditional algorithms in object detection and instance segmentation. The current mainstream instance segmentation algorithms are based on CNN models. The establishment of some well-known benchmark datasets such as MS COCO [56] and PASCAL VOC [11] has greatly promoted the development

of instance segmentation. A large number of algorithms have emerged, such as Mask R-CNN [12], Cascade R-CNN [14], HTC [16], PANet [21], YOLACT [25], DETR [39], [40], [41], and YOLO series [29], [30], [31], [32], [33]. However, most of these CNN-based methods are proposed for natural image scenes in benchmark datasets. Because CNN models rely heavily on the characteristics of the data, it is not advisable to directly employ existing methods for canopy instance segmentation. The above problems make it necessary to design a new CNN-based approach to addressing the instance segmentation task of individual tree crowns in aerial imagery.

In object detection and instance segmentation, the YOLO series [29], [30], [32], [33] has become a pivotal model, celebrated for its single-stage, real-time capabilities that efficiently balance speed and accuracy. This is especially beneficial in tree crown detection, where YOLO's agility enables rapid processing of forest imagery, crucial for accurately identifying individual crowns [55]. Unlike two-stage methods like mask R-CNN, which first generate region proposals before refining predictions, YOLO operates in a single-shot manner. This direct approach eliminates the need for a separate region proposal network, streamlining the segmentation process and significantly speeding up the analysis of aerial forest imagery. This rapid processing capability is vital for improving the operational efficiency of forest management and enhancing biodiversity modeling efforts.

YOLOv8 [33], as one of the latest iterations in the YOLO series, offers significant improvements in accuracy and speed, particularly for multiscale targets, making it well-suited for segmenting tree crowns. YOLOv8 incorporates specialized network head designed to detect objects of varying sizes—small, medium, and large—facilitating precise feature extraction critical for diverse object sizes. In cases, where datasets do not include all three size categories (small, medium, and large) but predominantly consist of tiny to medium-sized objects, standard models may underperform. This indicates a need for a customized method. Consequently, there may be a necessity to develop a new model tailored specifically for tasks like canopy segmentation, where object sizes vary significantly.

In response to the above analysis, the overall goal of this article is to introduce a new model designed to tackle the issue of instance segmentation and tree species classification for individual tree crowns. We propose a new method named YOLOv8-FF, which builds on the YOLOv8 framework. First, we develop a network design specifically tailored to the unique characteristics of canopy data, aiming at accurately predicting tiny, small, and medium objects. Following this, we introduce a novel feature fusion (FF) mechanism designed to enhance the representation of canopy features, ensuring richer and more precise feature integration. Last, we incorporate the Sparse Large Kernel Network (SLaK) module to improve the detection capabilities for medium objects, which are particularly challenging in dense forest imagery. In summary, the key contributions of the proposed method can be outlined follows.

1) We developed a tailored version of YOLOv8, named YOLOv8-FF, specifically designed to detect and segment individual tree crowns in aerial imagery while also classifying tree species. This model harnesses YOLOv8's robust capabilities and is further optimized to address the unique challenges of canopy data. YOLOv8-FF features a modified network structure that excludes detection head for large objects due to their rarity in canopy datasets and introduces a new head for detecting tiny objects. This refinement enables more precise detection across varying object sizes, crucial for handling the diverse range of tree crown dimensions in aerial forest imagery. Comparative analyses with leading single-stage and two-stage segmentation methods show that YOLOv8-FF outperforms its counterparts in canopy detection and mask segmentation while using fewer parameters.
2) In the feature extraction stage, we propose a new FF mechanism that incorporates both same-scale and cross-scale fusion techniques. This mechanism is designed to aggregate and enhance canopy feature information more effectively, facilitating improved segmentation accuracy and richer feature representation.
3) The SLaK module is integrated into the YOLOv8-FF framework, which enhances the model's ability to capture broader contextual information. This property is particularly beneficial for the accurate prediction of medium objects, which allows the model to better understand and interpret the spatial relationships and characteristics of objects within a larger field of view.

II. Related Work

In this section, we review the background of general instance segmentation and instance segmentation for individual tree crowns, and briefly revisit the principle of YOLOv8.

A. Instance Segmentation

Deep-learning based methods for instance segmentation are typically divided into two primary categories: two-stage methods and single-stage methods. Two-stage methods, exemplified by the Mask R-CNN [12] series, evolve from earlier two-stage object detectors like Fast R-CNN [13]. Mask R-CNN extends this approach by adding a parallel branch for mask prediction. Enhancements to this model include Cascade R-CNN [14], which boosts detection accuracy through a cascade of detectors, and Hybrid Task Cascade (HTC) [16], which incorporates a multitask, multistage cascaded architecture to improve spatial context, significantly outperforming earlier models. Another notable development is Mask Scoring R-CNN [15], which introduces a mask scoring mechanism to assess the quality of predicted instance masks. PANet [21] enhances feature information flow via a bottom-up pathway, augmenting the architecture inspired by Feature Pyramid Network (FPN) [22]. Despite their effectiveness, two-stage methods often struggle with achieving optimal processing speeds.

The single-stage approach is introduced to enable concurrent detection and segmentation operations, thereby significantly reducing reasoning time. For example, YOLACT [25] reframes the instance segmentation task as generating a series of prototype masks and forecasting mask coefficients for individual instances. Based on the principles introduced in YOLACT, subsequent

iterations of YOLO [29] like YOLOv5 [30], YOLOv7 [32], and YOLOv8 [33] integrate both object detection and instance segmentation within a unified pipeline. SOLO methods [27], [28] contribute to instance segmentation by introducing grid-based approaches that directly predict instance masks within each grid cell. PolarMask [26] introduces a novel representation of instance masks using polar coordinates, which offers advantages in handling instances with irregular shapes or rotations. CondInst [37] directly predicts instance masks within a single unified framework by dynamically adjusting convolutional operations based on instance-specific information.

B. Instance Segmentation for Individual Tree Crowns

CNN frameworks have demonstrated promising results in the identification and segmentation of tree canopies. For instance, Sani-Mohammed et al. [43] enhanced the Mask R-CNN framework for detecting and segmenting standing dead trees within dense mixed forests. Sun et al. [44] applied Mask R-CNN with a ConvNeXt [23] backbone to segment tree crowns in urban settings. Firoze et al. [46] developed an instance segmentation framework that utilizes pixel content, shape, and self-occlusion, augmented by a graph convolutional network to handle densely packed trees. Similarly, Sun et al. [45] utilize Cascade R-CNN for counting trees in a subtropical mega city by delineating the canopies. Dersch et al. [47] combine Mask R-CNN with DETR [39] to precisely delineate individual tree crowns using multispectral imagery and high-resolution lidar data. Zhou et al. [48] implemented BlendMask [42], a hybrid of Mask R-CNN and YOLACT, for multispecies individual tree crown segmentation and classification. Straker et al. [55] introduced a new approach using YOLOv5 for segmenting individual tree crowns, tested on the autonomous AAV-based laser scanning (AAV-LS) dataset For Instance. These techniques showcase the adaptability of instance segmentation methods to the unique challenges posed by tree crown imagery. However, many of them involve complex architectures that result in slower inference speeds. Moreover, it is still a challenging task to effectively solve canopy instance segmentation in low resolution images and complex environments with densely distributed and small objects.

C. YOLOv8

The YOLO model [29] processes the entire input image to directly predict the positions and bounding boxes of objects at the output layer. In YOLO models [29], [30], [31], [32], [33], the input image is divided into multiple grids, with each grid cell responsible for predicting the position and confidence level of objects within it. The YOLOv8 model [33] adopts an anchor-free approach, estimating the center of an object directly rather than predicting its distance from a predefined anchor box. This anchor-free method reduces the number of box predictions, thus accelerating the nonmaximum suppression (NMS) process, a critical postprocessing step that eliminates overlapping predictions. YOLOv8 is designed to accommodate diverse project requirements through various models scaled to different factors, including nano, small, medium, large, and extra-large versions. The architecture of YOLOv8 comprises three main components: the backbone, neck, and head, as depicted in Fig. 1. Compared to earlier YOLO versions [29], [30], [31], [32], YOLOv8 offers enhanced inference speed without sacrificing accuracy, making it the chosen baseline model for this study. However, while YOLOv8 performs well in general object detection tasks, its effectiveness in processing canopy images, particularly in detecting tiny objects, is limited and need be improved. To address this, we develop a new method for individual tree crowns identification by improving upon the YOLOv8 framework. This method aims to refine the model's capability to handle the specific challenges posed by tree crown instance segmentation.

III. Proposed Method

In this section, the proposed method is illustrated and discussed in detail. First, the network design is described. Then, this section introduced the proposed FF mechanism and the structure of SLaK module.

A. Overview

The proposed method for instance segmentation of tree crowns is constructed based on the XLarge version of YOLOv8, i.e., YOLOv8x [33]. First, as shown in Fig. 2, YOLOv8-FF introduces a tailored network design that is optimized to handle the unique features present in canopy images. Subsequently, it incorporates a new FF mechanism designed to effectively aggregate and enhance the richness of feature information. Finally, the SLaK module is employed to replace the bottleneck module in the last C2f section of the head, which is expected to significantly improve the prediction accuracy for medium objects and overall efficiency.

Like its predecessor, YOLOv8, the YOLOv8-FF model maintains the "backbone-neck-head" architecture. The structure of this model is illustrated in Fig. 3, comprising three primary components: Backbone, Neck, and Head. The backbone serves as the foundation, extracting features from the input image through a series of convolutional layers that progressively decrease the spatial dimensions while increasing the depth, or number of channels. Key elements of the backbone include the Conv-BN-SiLU (CBS), C2f, and Spatial Pyramid Pooling Fast (SPPF) modules. The CBS module is used to extract features from the input data, which consists of a convolution layer (Conv), batch normalization (BN), and Sigmoid Linear Unit (SiLU) activation function. The C2f module is an optimized version of the C2 module, which stands for the cross stage partial (CSP) Bottleneck [49] with two convolutions. The C2f module is a faster implementation of the C2 module, which improves the execution speed of the model while maintaining similar performance. By utilizing residual connections, the C2f module enhances feature representation and improves gradient propagation in the network. SPPF is used to handle inputs of varying sizes and enhance feature extraction by pooling and aggregating features at multiple levels within the network, ensuring comprehensive information capture.

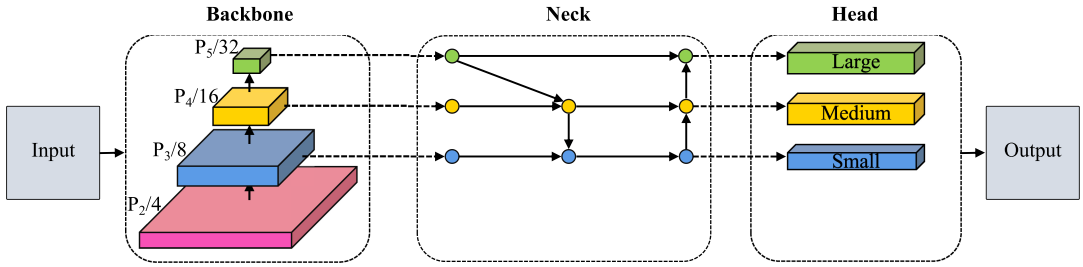


Fig. 1. Network framework of YOLOv8, which is mainly composed of three parts: backbone, neck, and head. Within the backbone, images are progressively downsampled at different levels, designated as $P_2/4$, $P_3/8$, $P_4/16$, and $P_5/32$, corresponding to downsampled sizes by factors of 4, 8, 16, and 32, respectively. YOLOv8 adopts PANet [21] as the FF network, which contains a top-down path and a bottom-up path. The two paths are displayed in the neck.

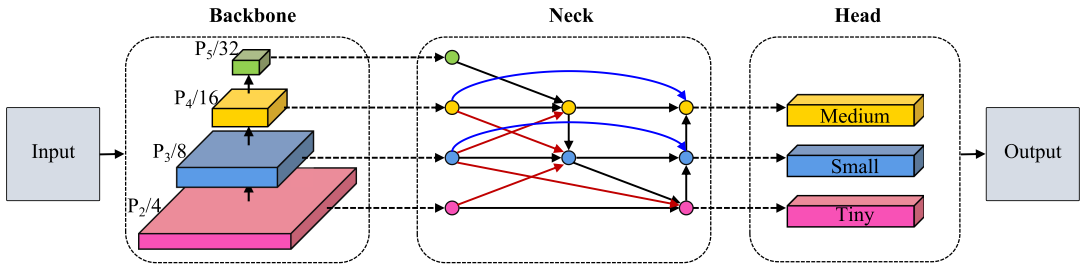


Fig. 2. Framework of the proposed method. The proposed FF mechanism consists of the blue and red connections in the neck. The blue connections aims to fuse same-scale features, and the red connections aim to fuse cross-scale features.

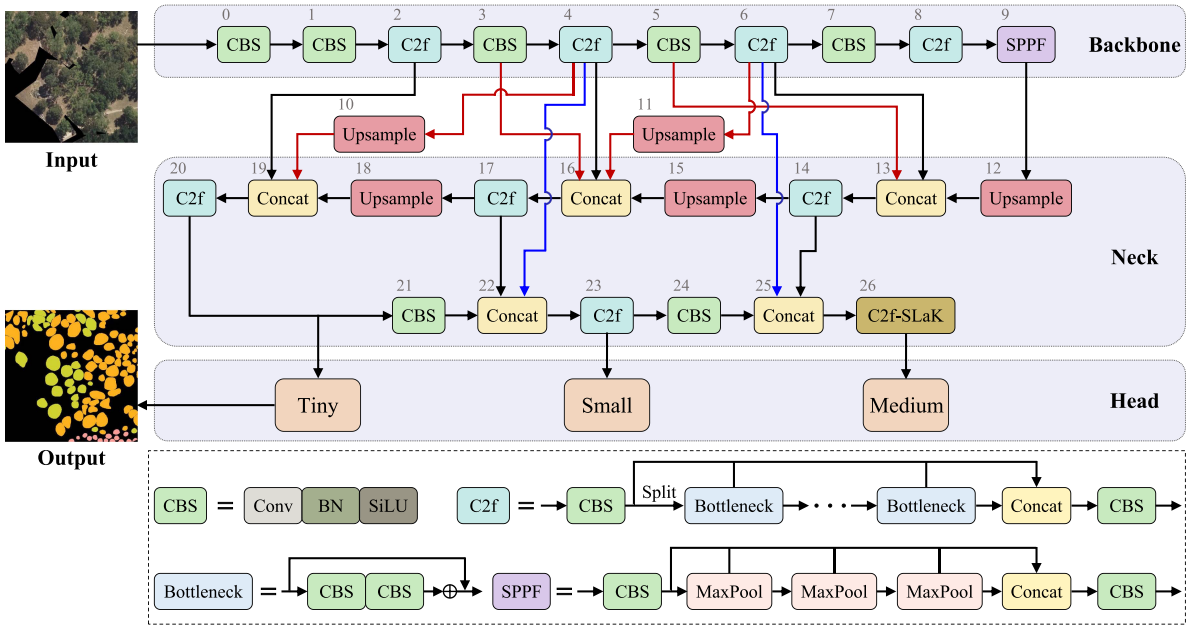


Fig. 3. Network architecture of the proposed method YOLOv8-FF. Each module is labeled numerically to indicate its sequence within the network.

Following the backbone, the neck component employs a combination of feature aggregation and concatenation techniques to merge features from various scales effectively. This integration facilitates the capture of multiscale information, which is crucial for accurate object prediction. The head of the network is tasked with generating the final predictions, including three specialized branches that handle the segmentation of tiny, small, and medium tree canopies. This part uses convolutional layers to predict bounding boxes, objectness scores, class probabilities, and masks for detected objects. In addition, the head employs a decoupled structure that separates the tasks of classification, detection, and segmentation. It incorporates anchor-free

techniques and utilizes distribution focal loss to optimize the loss function.

B. Network Design

The network architecture of YOLOv8, illustrated in Fig. 1, extracts multiscale features at different levels of the network. These features are then directed to separate detection heads, each calibrated to recognize objects at a particular scale. These processed feature maps are downsampled by factors of 32, 16, and 8, respectively, allowing the network to capture details at different resolutions. Upon analyzing tree crown characteristics in our dataset, it was observed that large objects are very rare, with the majority of crowns being medium or small. Moreover, many crowns are very small or tiny, with some occupying less than 20 pixels in a 512 $\times$ 512 image. At the smallest downsampling rate of 8 times, critical feature information for these tiny objects is often lost, underscoring the challenge of maintaining accuracy for smaller objects in aerial imagery. Our analysis concludes that canopy data might not need large object detection, and separate prediction is required for tiny tree crowns.

Given this data characteristic, our tailored network design for YOLOv8 eliminates the detection of large objects, focusing instead on enhanced detection capabilities for tiny, small, and medium objects, as shown in Fig. 2. The design adjustments directly address the predominant size categories, optimizing the model for the most frequent cases while conservatively using resources for the sparse category of large objects. This approach enhances the model's efficiency and effectiveness in segmenting and identifying tree crowns in aerial imagery tailored to the challenges posed by the dataset.

The network architecture is modified to accommodate these requirements: the large object detection segment is removed, streamlining the network for efficiency and relevance to canopy data. A detection head for tiny objects is added, positioned before the small object detection to prioritize processing of the smallest crowns. This addition necessitates the integration of an upsampling step, a concatenation, and a C2f module (layers 18–20 of Fig. 3) in the neck section before linking to the tiny object detection head. Following this, a CBS module, another concatenation, and a second C2f module (layers 21–24) are strategically placed to bridge the tiny and small object detection heads. This configuration ensures a fluid transition and effective feature processing across the different scales of tree crowns in the dataset.

C. Feature Fusion

The FF mechanism in YOLOv8 has been enhanced by adopting the PANet structure, which includes a top–down path, a bottom–up path, and lateral connections. To enhance the model's capability for generating informative features, we develop a new and lightweight FF mechanism into YOLOv8 and make it YOLOv8-FF, as depicted in Fig. 2. This enhancement is strategically located in the neck of the network, optimizing the integration of features at this crucial juncture to enhance overall model performance.

The proposed FF is divided into two main types: a cross-scale fusion and a same-scale fusion. The cross-scale FF, represented by the red connections in Fig. 2, aims to integrate features across different resolutions, which is inspired by High-Resolution Network (HRNet) [51]. It typically involves adjusting these features to a common resolution before concatenation, which is essential for effective multiscale feature integration. These cross-scale connections encourage resolution alignment through two complementary pathways: 1) when high-resolution feature maps are combined with lower-resolution counterparts, the feature maps undergo strided convolution for downsampling, ensuring semantic consistency while reducing spatial dimensions; 2) when low-resolution features need to be integrated with higher-resolution features, nearest neighbor upsampling is applied to increase spatial dimensions while preserving contextual information. Similar to HRNet's design philosophy, our approach implements this cross-scale fusion process repeatedly throughout the network as shown in Fig. 3, where multiple cross-scale connections operate at different levels of the architecture. This iterative fusion strategy creates a bidirectional information exchange that progressively refines feature representations. Through this process, semantic information from low-resolution streams enhances feature discrimination at high resolutions, while detailed spatial information from high-resolution streams improves object localization at low resolutions, ultimately leading to more robust multiscale feature representations. As indicated in Fig. 3, layer 3 is connected to layer 16, and layer 5 is connected to layer 13. The CBS module implements the downsampling operation by setting the convolution kernel to 3, the stride to 2, and the padding to 1. Low-resolution feature maps are upsampled and then fused with high-resolution feature maps. This is exemplified by the connections from layer 4 to layer 19 and from layer 6 to layer 16 through the upsampling processes outlined in layers 10 and 11.

The same-scale FF, indicated by the blue connections, leverages a strategy inspired by Bi-FPN [50]. This approach enhances the utilization of features extracted directly from the backbone, improving the network's ability to handle features at similar scales. Unlike Bi-FPN's weighted fusion method, our model employs a simpler approach by concatenating feature maps directly, which simplifies the fusion process while maintaining effectiveness. Specific implementations include connections from layer 4 to layer 22 and from layer 6 to layer 25 as detailed in Fig. 3. The integration of the proposed FF framework into YOLOv8 enhances the model's capability to process complex tree crown imagery data, by effectively managing features across different scales and resolutions.

D. SLaK

Detecting large objects aids in identifying medium objects as well, because the process involves capturing expansive spatial features crucial for accurately recognizing medium objects. The removal of the large object detection head impacts the performance of medium object detection to a certain extent. Inspired by RepLKNet [52] and SLaK [53], which prove that applying large convolutional kernels instead of a stack of small

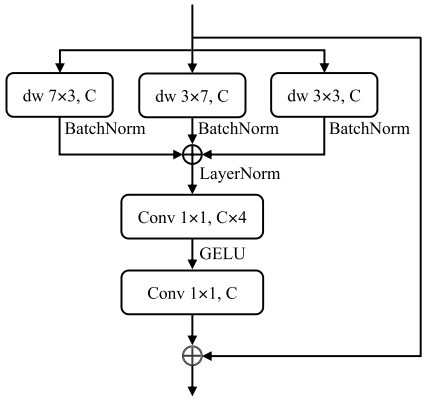


Fig. 4. Structure of SLaK in C2f-SLaK of the head. A layer is shown as kernel size and output channels. C represents the number of channels in the input feature map, dw represents the depthwise convolution.

kernels shows better performance, we incorporate the SLaK module into the C2f module, renamed as C2f-SLaK and shown in Fig. 3, in an attempt to improve the prediction performance of medium objects by increasing the convolution kernel. The structure of SLaK, as integrated into C2f-SLaK, is depicted in Fig. 4. The SLaK module combines specialized convolutional operations with Batch Normalization (BatchNorm) layers, post-layer normalization (LayerNorm), and Gaussian Error Linear Unit (GELU) [54] activation. GELU is a nonlinear activation function defined as follows:

$$\mathrm{GELU}(x) = x \cdot \Phi(x) \tag{1}$$

where $\Phi(x)$ represents the cumulative distribution function of the standard normal distribution, which can be expressed as follows:

$$\Phi(x) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right) \tag{2}$$

where *erf* is the error function. This activation function provides smooth scaling of both positive and negative inputs, improving gradient flow during training.

The structure of SLaK is built based on the ConvNeXt [23] block. SLaK modifies a large kernel by decomposing it into two parallel rectangular kernels and retains a smaller convolutional layer in parallel. This configuration helps in capturing a broader receptive field while maintaining detail resolution. The outputs from these three convolution layers, each followed by a BatchNorm layer, are summed to integrate the feature information effectively. Given that the feature map input to C2f-SLaK is derived from an image downsampled 32 times, the kernel sizes for the depthwise convolution layers are set to 7×3, 3×7, and 3×3, which are shown close-to-optimal for processing lower resolution inputs. LayerNorm, a 1×1 convolution increases the channel dimensions fourfold, followed by a GELU activation function. Another 1×1 convolution then restores the dimensions to match the original input size. To ensure stability in learning and performance, a residual connection is incorporated, allowing for the flow of original features along with enhanced features.

The integration of the SLaK module into YOLOv8 enhances the network's performance in detecting medium objects. By employing large convolutional kernels that decompose into specialized, parallel configurations, SLaK effectively broadens the receptive field. This allows the model to capture more contextual information from the input images, thereby improving accuracy in medium object detection.

IV. Experiment Design

A. Datasets

Manaaki Whenua – Landcare Research, a Crown Research Institute based in New Zealand, has contributed a manually labelled dataset featuring aerial imagery with individual tree crowns. The aerial survey were captured from a top–down perspective in 2021, encompassing the rural hill-country of Wairarapa within the Greater Wellington region, New Zealand.[1] The imagery, provided at a 30-cm pixel resolution and presented in 3-band (RGB) orthophotos, underwent tree species labeling through field-based mapping conducted by Spiekermann et al. [60] in the specified study area. In a Geographic Information System (GIS), polygon areas representing individual tree species were manually outlined based on the tree crown edges. This meticulous process resulted in the delineation of 38 601 tree objects. Areas where tree crowns lacked complete annotation information (including individual crown boundaries and species information) were manually excluded by experts, who assigned zero values to the corresponding areas in both the imagery and the label raster data. This careful exclusion ensures that the model did not learn from inaccurate or incomplete data, thereby maintaining the integrity and accuracy of the training process.

A uniform grid with a tile size of 153.6 m, equivalent to $512\times$ 512 pixels given the 30 cm resolution of the imagery, was applied over the imagery and the study area. These tiles were then randomly allocated into a training set, comprising 26 424 objects across 473 images, a validation set, comprising 7732 objects across 118 images, and a test set, containing 4445 objects in 66 images. We train YOLOv8-FF on the training set and evaluate its performance on the validation set at the end of each epoch. Training concludes when specific termination criteria are met, such as completing the number of training epochs or not observing any improvement over 50 consecutive epochs (early stopping). The early stopping condition is determined by a weighted combination of performance metrics calculated on the training set, specifically the AP and AP_{50} metrics. Here, AP is assigned a weight of 0.9, and AP_{50} a weight of 0.1. In the testing phase, the trained model is then evaluated on the test set to assess its performance.

The dataset encompasses 28 297 tree crowns of six species categories: conifers, kānuka, willow-poplar, eucalyptus, acacia, and other natives (o.natives). The conifers category includes

[1] Sourced from the LINZ Data Service and licensed by Greater Wellington Regional Council for reuse under CC BY 4.0. Data link: https://data.linz.govt.nz/layer/105727-wellington-03m-rural-aerial-photos-2021/.

TABLE I
NUMBER OF TREE CROWNS AND INSTANCES PER CATEGORY

Class	conifers	kānuka	willow-poplar	eucalyptus	acacia	o.natives
Crowns	4,339	13,661	5,645	1,414	857	2,381
Training Instances	4,639	11,234	5,778	1,515	1,081	2,177
Validation Instances	949	4,490	1,393	272	34	594
Test Instances	438	2,052	1,081	192	227	455
Total Instances	6,026	17,776	8,252	1,979	1,342	3,226

TABLE II
NUMBER OF SMALL, MEDIUM, AND LARGE OBJECTS IN THE CANOPY DATASET

Number	Tiny	Small	Medium	Large
Training	15,055	8,728	2,636	5
Validation	4,686	2,404	642	0
Test	2,641	1,324	479	1
Total	22,382	12,456	3,757	6

various coniferous species like radiata, spruce, cedar, and Douglas fir, totaling 4339 crowns. The kānuka category comprises 13661 Kunzea spp. crowns. Willow-poplar encompasses all mapped poplar (Populus spp.) and willow (Salix spp.) canopies, amounting to 5645 crowns. The eucalyptus category includes 1414 crowns of various eucalyptus species, such as Eucalyptus globulus. The acacia category comprises 857 Acacia dealbata crowns. The natives category contains 2381 crowns of native species like totara (Podocarpus totara) and cabbage trees (Cordyline australis). Due to cropping, more instances may exist in the dataset than the counted number of tree crowns. Table I provides details on the number of tree crowns and instances per category.

In the COCO dataset, objects are categorized into small, medium, and large based on their pixel area within their mask [56]. Specifically, objects are considered small if their area is less than 32^2 pixels, medium if the area is between 32^2 and 96^2 pixels, and large if the area exceeds 96^2 pixels. In our canopy dataset, we further refine the classification of small objects: those with an area smaller than 16^2 pixels are deemed tiny, while those with an area between 16^2 and 32^2 pixels are classified as small. In terms of real-world crown size, objects are categorized as tiny if their area is less than 4.8^2 m^2, small if their area is between 4.8^2 m^2 and 9.6^2 m^2, medium if the area is between 9.6^2 m^2 and 28.8^2 m^2, and large if the area exceeds 28.8^2 m^2. A statistical analysis of the canopy dataset reveals a predominance of tiny (22382 instances), small (12456 instances), and medium (3757 instances) objects, with only six instances of large objects, as detailed in Table II.

B. Comparison Methods

To evaluate the efficacy of the proposed approach, a comparative analysis is conducted using 17 recent instance segmentation methods on a dataset of tree images. The peer competitors encompass both two-stage and single-stage approaches. Among the two-stage methods, seven prominent top–down approaches are considered: Mask R-CNN [12], [64], Cascade R-CNN [14], HTC [16], DetectoRS [17], MogaNet [18], poolformer [19], Efficientformer v2 [20], and QueryInst [36]. In addition, the single-stage category includes YOLACT [25], [65], SOLOv2 [28], CondInst [37], RTMDet [38], YOLOv5 [30], [55], YOLOv7 [32], YOLOv8 [33], [66], YOLOv9 [34], and YOLOv10 [35].

For the two-stage methods, experiments are carried out with two backbone structures, namely ResNet-101 [2] and ResNeXt-101 (64 × 4d variant[3]). These architectures are employed for Mask R-CNN, Cascade R-CNN, HTC, and DetectoRS, following the specifications outlined in their respective papers [12], [14], [16], [17]. The implementation of the eight two-stage methods, YOLACT, SOLOv2, CondInst, and RTMDet is realized using the MMDetection open-source detection toolbox [24]. Training of these nine methods is performed with pre-trained backbone weights from the ImageNet dataset [57], aligning with the MMdetection protocol. The YOLO models (YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv10) undergo training for an equivalent number of epochs as the proposed method to facilitate a comprehensive and fair evaluation.

C. Parameter Settings

In the experiments, precision, recall, F1-score, and Average Precision (AP) are selected as the main evaluation metrics [58], [59], and they are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

$$\text{AP} = \frac{1}{10}(\text{AP}_{50} + \text{AP}_{55} + \text{AP}_{60} + \cdots + \text{AP}_{95}). \tag{6}$$

where TP, FP, and FN represent the true positive, false positive, and false negative counts, respectively. A prediction whose Intersection over Union (IoU) value is greater than a predefined threshold and correctly classified is considered a TP.

The high precision indicates that when the model predicts a positive outcome, the more likely it is to be correct. The high recall rate indicates that the model is good at capturing all relevant instances of the positive class. The F1-score seeks to find a balance between precision and recall by calculating the harmonic mean of the two. In further analysis of the results, we present detailed evaluations on precision, recall, and F1-score metrics. To reduce the influence of relying on a single IoU threshold, we analyze both the maximum and average values across 10 IoU thresholds, ranging from 0.50 to 0.95 in increments of 0.05. According to MS COCO [56], AP is defined as the average AP under 10 IoU thresholds of 0.50:0.05:0.95, where AP is the area under the Precision–Recall curve.

In addition, we follow MS COCO to report the evaluation results of AP_{50}, AP_{75}, AP_{Small}, and $\text{AP}_{\text{Medium}}$ indicators. AP_{50}

[2] 101 indicates the number of convolutional layers
[3] 64× 4d means cardinality = 64 and bottleneck width = 4d.

TABLE III
COMPARISON RESULTS WITH TWO-STAGE TOP–DOWN METHODS AND SINGLE-STAGE METHODS ON THE TEST DATASET.

Method	Backbone	Box AP	Mask AP	AP^{Mask}_{50}	AP^{Mask}_{75}	AP^{Mask}_{S}	AP^{Mask}_{M}	FLOPs	Params
Mask R-CNN [12]	ResNet-101 [64]	14.2	13.4	31.5	9.9	10.9	34.6	134.35G	62.8 M
	ResNeXt-101	15.3	14.4	33.7	10.4	12.0	35.9	174.6G	101.5M
Cascade R-CNN [14]	ResNet-101	16.0	15.2	34.8	11.3	12.7	35.2	256.4G	101.5M
	ResNeXt-101	16.2	15.2	34.7	11.1	12.7	33.8	305.6G	134.5M
HTC [16]	ResNet-101	16.3	15.2	35.6	10.9	12.5	30.5	267.9G	95.9M
	ResNeXt-101	16.0	15.1	34.2	11.8	12.7	33.0	308.2G	134.7M
DetectoRS [17]	ResNet-101	15.8	15.1	33.7	12.7	12.8	32.4	250.5G	196.6M
	ResNeXt-101	15.8	15.1	34.2	11.9	12.8	30.9	259.1G	297.9M
Mask R-CNN [12]	MogaNet-XL [18]	18.4	17.3	35.3	15.8	15.2	33.6	175.1G	101.7 M
Mask R-CNN [12]	poolformer-s36 [19]	17.9	17.0	36.5	14.1	14.3	36.4	117.9G	50.1 M
Mask R-CNN [12]	Efficientformer v2-L [20]	14.5	13.7	26.0	13.0	11.9	27.4	105.0G	45.3 M
QueryInst [36]	ResNet-101	17.3	16.5	39.0	12.6	14.5	29.2	320.9G	182.0M
YOLACT [25], [65]	ResNet-101	14.3	13.2	30.9	9.8	10.2	26.8	67.4G	53.8M
SOLOv2 [28]	ResNet-101	-	8.9	22.6	6.1	6.2	26.6	-	-
CondInst [37]	ResNet-101	19.5	18.2	41.0	14.1	15.6	32.2	105.3G	53.0M
RTMDet [38]	XLarge	18.5	16.9	36.0	14.0	13.4	36.3	810.4G	111.7M
YOLOv5 [30], [55]	XLarge	24.2	21.7	47.8	16.9	19.8	35.5	84.8G	88.3M
YOLOv7 [32]	XLarge	23.9	21.6	48.4	17.0	19.8	32.3	73.5G	72.4M
YOLOv8 [33], [66]	XLarge	26.1	23.5	48.9	19.3	21.2	38.2	110.3G	71.8M
YOLOv9 [34]	E-version	26.5	23.1	49.7	19.4	20.9	38.1	76.1G	60.4M
YOLOv10 [35]	XLarge	25.9	23.5	49.3	20.2	21.3	**40.4**	89.0G	39.6M
YOLOv8-FF	XLarge	**27.9**	**24.7**	**50.9**	**21.8**	**22.9**	37.6	196.1G	54.4M

Params refer to the number of parameters. The symbol "-" implies that the corresponding results cannot be reported using the MMDetection toolbox.
The bold numbers represent the maximum values in each column.

and AP_{75} are calculated under specific thresholds, i.e., 0.50 and 0.75. AP_{Small} and AP_{Medium} are used to evaluate the model's predictive performance for small and medium objects. Since there are almost no large objects in the canopy dataset, AP_{Large} is not included in the experiments. For a thorough comparison, we also assess the computational complexity, quantified by the number of floating-point operations (FLOPs), and the number of parameters of the models.

The experiments are conducted on Quadro RTX 6000 GPU cards and implemented using the PyTorch framework. All YOLO models are trained from scratch. The proposed method is configured with a batch size of 4 and undergoes 1500 training epochs. As no pretrained weights are used, this extended training duration ensures the weights converge and achieve good performance [61]. The remaining hyperparameters of the proposed method are set in accordance with the hyperparameter configurations of YOLOv8 [33]. Training is conducted on a single GPU, utilizing the SGD optimizer for adjustments. The initial learning rate is set at 0.01, accompanied by a weight decay of 0.0005 and a momentum of 0.937. During the training process, image augmentation techniques include HSV augmentation, image translation, image scale, image flip, and image mosaic are adopted [33]. In addition, the loss function and weight configurations for each component align with the settings established in YOLOv8 [33].

V. RESULTS AND DISCUSSIONS

In this section, we first report experimental results, both quantitative and qualitative. We then introduce a series of ablation studies for further analysis and discussions.

A. Main Results

1) Quantitative Results: Table III provides a comprehensive overview of the experimental outcomes of YOLOv8-FF alongside its selected peer competitors concerning object detection and instance segmentation performance, FLOPs, and the number of parameters (Params). FLOPs are computed for an input image size of 512 × 512. Box AP is employed to assess the performance in the object detection task, while Mask AP is utilized to evaluate instance segmentation performance.

In comparison to two-stage methods, the proposed method YOLOv8-FF as a single-stage method exhibits significant advancements across multiple evaluation metrics. Particularly, when contrasting with the best-performing two-stage method, MogaNet, YOLOv8-FF demonstrates a remarkable improvement, with a 9.5 increase in Box AP and an 7.4 increase in Mask AP, surpassing MogaNet's values of 18.4 for Box AP and 17.3 for Mask AP. This substantial enhancement underscores the efficacy of YOLOv8-FF in accurately localizing objects within tree images. Furthermore, YOLOv8-FF maintains efficiency with fewer parameters and competitive FLOPs compared to many two-stage methods.

Against single-stage methods, YOLOv8-FF outperforms YOLACT, SOLOv2, CondInst, and RTMDet across all AP metrics. In contrast to YOLO models, YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv10 have lower FLOPs, but their general performance is not as good as YOLOv8-FF. Moreover, the number of parameters of these models is higher than that of YOLOv8-FF, except for YOLOv10. Concerning the baseline model in terms of Box AP and Mask AP, YOLOv8-FF outperforms YOLOv8 by 1.8 and 1.2, respectively. YOLOv8-FF

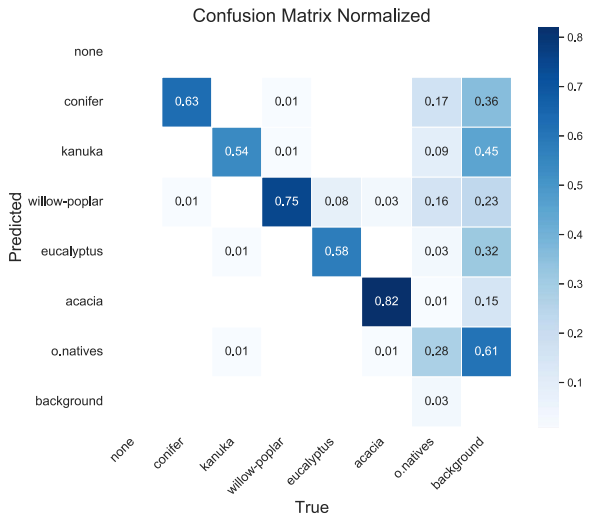


Fig. 5. Confusion matrix obtained by YOLOv8-FF.

TABLE IV
INDIVIDUAL CLASS RESULTS OF YOLOV8-FF, INCLUDING BOX AP AND A SERIES OF MASK AP SCORES

Class	Box AP	Mask AP	AP^{Mask}_{50}	AP^{Mask}_{75}	AP^{Mask}_{S}	AP^{Mask}_{M}
conifers	25.3	22.2	53.0	15.3	22.6	28.0
kānuka	14.1	12.3	26.2	10.6	11.8	43.0
willow-poplar	34.4	30.8	61.4	28.6	28.1	39.3
eucalyptus	38.5	34.4	61.2	33.2	26.7	46.4
acacia	32.8	28.7	61.3	26.2	28.6	32.2
o.natives	22.3	19.9	42.1	16.9	19.5	36.4

achieves the fourth best result in the AP^{Mask}_{M} metric, only 0.6 points behind the baseline YOLOv8 model, 0.5 points behind the YOLOv9 model, and 2.8 points behind YOLOv10. Since removing large object detection from the baseline model inevitably reduces the prediction performance for medium objects, due to the loss of broader spatial features. YOLOv8-FF slightly increases FLOPs by about 86G compared to YOLOv8, but the number of parameters of YOLOv8-FF is significantly lower than that of YOLOv8 and most single-stage methods.

2) Tree Species Results: Fig. 5 presents the confusion matrix for the prediction of different tree species achieved by YOLOv8-FF, vividly illustrating the model's ability to correctly identify and differentiate between species. We present the performance of individual class obtained by YOLOv8-FF in Table IV. YOLOv8-FF particularly excels in detecting and segmenting the eucalyptus class, achieving a Box AP of 38.5 and a Mask AP of 34.4. In contrast, the kānuka class exhibits the lowest accuracy, with a Box AP of 14.1 and a AP^{Mask}_{M} of 12.3. This class notably contains a large number of instances, reaching 17,776, and is characterized by small crown sizes and a dense distribution, as depicted in the penultimate image of Fig. 6. These factors significantly increase the complexity of instance segmentation for this class.

3) Qualitative Results: Fig. 6 displays the visual results of YOLOv7, YOLOv8, and the proposed YOLOv8-FF across six test images. These methods generally succeed in identifying most tree crowns within the images. For the objects within the blue boxes in the second image, YOLOv7 misclassifies them, YOLOv8 misses one, and YOLOv8-FF correctly detected these two objects. In the third and fourth images, YOLOv7 generates several overlapping predictions as highlighted in the blue boxes, whereas YOLOv8 and YOLOv8-FF produce more dispersed results. The fifth image reveals that all three methods have misclassification problems, and YOLOv7 misses some objects, which also reveals the challenge of this problem and the need for further research.

TABLE V
ACCURACY ASSESSMENTS OF YOLOV7 ON SIX TEST IMAGES

Image	1	2	3	4	5	6
Num of canopies	59	44	105/35/19/2	94/16	126/14/42	59/7/2
Num of TPs	49	38	75/16/14/2	81/15	30/2/4	39/2/1
Num of FPs	20	7	39/13/4/0	33/3	32/0/9	30/3/4
Num of FNs	10	6	30/19/5/0	13/1	96/12/38	20/5/1
Recall	83.1	86.4	72.7	90.0	15.9	48.2
Precision	71.0	84.4	74.7	77.2	59.7	38.8
F1-score	76.6	85.4	73.5	83.1	23.8	40.9

TABLE VI
ACCURACY ASSESSMENTS OF YOLOV8 ON SIX TEST IMAGES

Image	1	2	3	4	5	6
Num of canopies	59	44	105/35/19/2	94/16	126/14/42	59/7/2
Num of TPs	45	36	75/17/15/1	71/15	25/2/6	39/3/0
Num of FPs	20	8	27/12/4/0	16/0	30/0/13	36/4/0
Num of FNs	14	8	30/18/4/1	23/1	101/12/36	20/4/2
Recall	76.3	81.8	62.2	84.6	16.1	36.3
Precision	69.2	81.8	77.8	90.8	59.0	31.6
F1-score	72.6	81.8	69.1	87.6	25.3	33.8

TABLE VII
ACCURACY ASSESSMENTS OF YOLOV8-FF ON SIX TEST IMAGES

Image	1	2	3	4	5	6
Num of canopies	59	44	105/35/19/2	94/16	126/14/42	59/7/2
Num of TPs	47	37	74/14/13/1	78/15	26/2/7	35/3/1
Num of FPs	14	6	19/7/4/0	11/ 1	21/0/15	28/0/1
Num of FNs	12	7	31/21/6/1	16/1	100/12/35	24/4/1
Recall	79.7	84.1	57.2	88.4	17.2	50.7
Precision	77.0	86.0	80.7	90.7	62.4	68.5
F1-score	78.3	85.1	67.0	89.5	27.0	58.3

Tables V–VII detail the accuracy evaluations of YOLOv7, YOLOv8, and YOLOv8-FF, respectively, using six test images. These tables show a variety of performance metrics such as the numbers of canopies, TPs, FPs, FNs, along with calculated Recall, Precision, and F1-score, all determined under an IoU threshold of 0.5. In images 3-6, the columns "Num of canopies," "Num of TPs," "Num of FPs," and "Num of FNs" contain multiple numerical values (e.g., 105/35/19/2), representing the counts of various categories in each image. YOLOv8-FF generally exhibits superior precision, recall, and F1-scores compared to YOLOv7 and YOLOv8, particularly excelling in images 5 and 6. Although YOLOv7 has a higher recall rate on some images, it produces many overlapping predictions, resulting in lower precision than YOLOv8-FF on all images. Moreover, YOLOv8-FF outperforms YOLOv8 in recall rate, except for image 3, but YOLOv8-FF has higher precision.

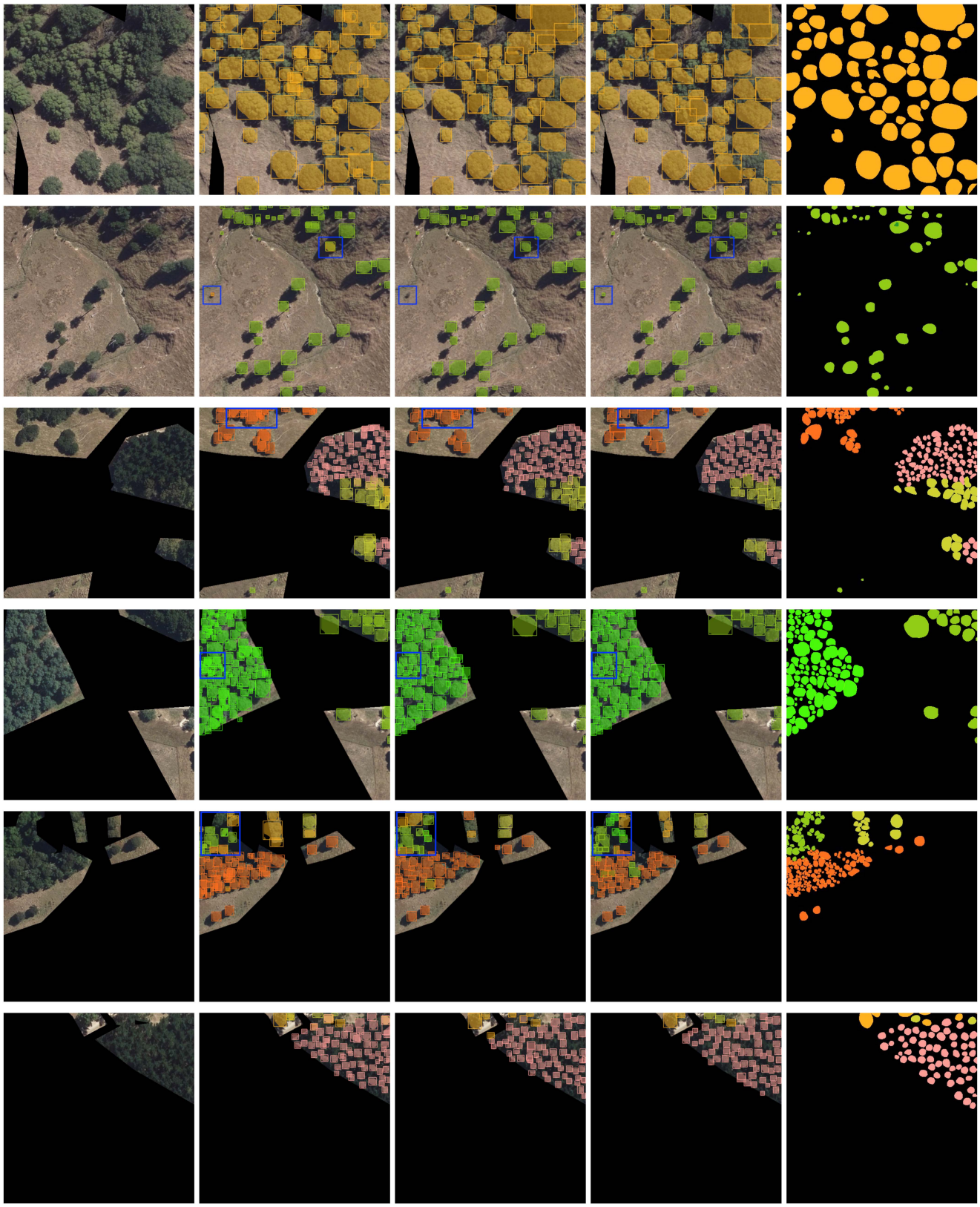


Fig. 6. Comparisons of the visual results of YOLOv7, YOLOv8, and the proposed method on the test dataset. The category information of the prediction results can be judged by color: conifers, kānuka, willow-poplar, eucalyptus, acacia, and o.natives. All tree species are included in these six images. (a) Input. (b) YOLOv7. (c) YOLOv8. (d) YOLOv8-FF. (e) Ground truth.

TABLE VIII
ABLATION OF DESIGNED MODULES. P, R, AND F1 REFER TO PRECISION, RECALL, AND F1-SCORE.

Method	Box AP	Mask AP	AP^{Mask}_{50}	AP^{Mask}_{75}	AP^{Mask}_{S}	AP^{Mask}_{M}	Max P	Max R	Max F1	Avg P	Avg R	Avg F1	FLOPs	Params
YOLOv8 [33]	26.1	23.5	48.9	19.3	21.2	38.2	70.4	51.4	58.9	41.8	26.3	31.8	110.3G	71.8M
+ p234	26.3	23.5	49.0	19.4	21.4	34.1	69.0	51.8	58.5	40.1	26.2	31.2	195.8G	56.2M
+ p234 + SS FF	26.7	23.5	49.2	19.6	21.8	34.3	70.4	51.4	58.9	41.8	26.3	31.8	196.7G	56.7M
+ p234 + CS FF	27.0	23.9	50.1	20.7	22.0	**38.4**	67.5	**53.6**	59.0	41.0	**27.3**	32.3	198.4G	57.0M
+ p234 + FF	27.2	24.1	49.6	20.5	22.7	36.4	67.5	**53.6**	59.0	41.0	**27.3**	32.3	199.2G	57.5M
+ p234 + FF + SLaK	**27.9**	**24.7**	**50.9**	**21.8**	**22.9**	37.6	**70.7**	53.1	**60.1**	**43.0**	**27.3**	**33.0**	196.1G	54.4M

The bold numbers represent the maximum values in each column.

Despite the promising results of YOLOv8-FF, variations in performance across different images highlight challenges and potential limitations in handling diverse environmental conditions or varying canopy densities. The notable number of false negatives for the "kānuka" category in image 5, across all three models, underscores the complexity of segmenting densely distributed, small tree crowns. This emphasizes the ongoing challenges in crown instance segmentation, as explored in this study.

B. Further Analysis

In this section, we examine the impacts of the individual contributions made in this article, namely, the redesigned network structure, the FF method, and the SLaK module, respectively. The ablation results are displayed in Table III. Besides evaluating AP metrics, we also provide data on precision, recall, and F1-score metrics. To minimize the impact of a single IoU threshold, we compare both the maximum and average values across ten thresholds ranging from 0.50 to 0.95 in increments of 0.05. Typically, the highest values are achieved at the IoU threshold of 0.5, as this setting allows more predictions to be considered TPs compared to higher thresholds.

1) Network Design: The modified network architecture segments the canopy instance segmentation task by focusing on tiny, small, and medium objects. To assess the validity of this approach, comparative experiments were conducted by altering the network architecture of the baseline model, YOLOv8, to observe its impact on performance. Throughout the ablation experiments in this study, a consistent training setting was maintained. The results, as shown in Table III, indicate that the new network design, referred to as "+ p234," improves performance in some metrics. This model achieves a 0.2 increase in Box AP and a 0.2 rise in $AP_S{}^{Mask}$. However, removing the large object detection component has led to a decline in $AP_M{}^{Mask}$, suggesting the role of large object detection in aiding medium object identification.

2) FF: An outstanding point of YOLOv8-FF is that the FF stragety can flexibly aggregate cross-scale and same-scale features at the same time, which improves the multiscale detection capability of the model. A comparison of architecture performance for FF mechanisms that include the cross-scale FF (CS FF) and the same-scale FF (SS FF) alone, as well as the FF mechanism that include both, is shown in Table III. Basically, the proposed FF method leads to better performance. Specifically, comparing the model "+p234," The addition of either cross-scale or same-scale FF brings significant performance improvements. It is worth noting that both methods enhance the model's prediction of medium objects, making the models increase by 0.2 and 4.3 on $AP_M{}^{Mask}$. When combining the cross-scale and the same-scale FF in the model, it brings greater improvement compared to one alone. Moreover, the proposed FF is lightweight and bring about a slight increase in FLOPs and the number of parameters.

3) SLaK: SLak is integrated into the network to improve the model's feature extraction ability for medium objects by expanding the receptive field. Integrating both FF mechanisms individually already enhances the AP^{Mask}_{M} metric, although when these mechanisms are combined together, the metric decreases slightly, by approximately 2.0, compared to using CS FF alone. The addition of SLaK to the model further improves its performance, particularly increasing the $AP_M{}^{Mask}$ by 1.2, indicating a positive impact on the detection capabilities for medium objects. Moreover, SLaK retains the advantages of large convolutional kernels while reducing the overall parameter count by 3.1M, thus improving the model's computational efficiency.

These findings demonstrate the effectiveness of our proposed method, YOLOv8-FF, which leverages structural modifications and advanced modules to achieve superior performance in segmenting and classifying tree crowns in complex images.

C. Discussions

The study's dataset required masking certain areas where tree crowns lacked essential labels (individual crown masks and species labels). Including these unlabeled regions in training would risk the model developing incorrect associations and patterns, which could diminish prediction accuracy on new data.

The introduction of YOLOv8-FF presents a meaningful enhancement in the instance segmentation task of tree canopies, especially suitable for complex forest environments. This tailored network architecture integrates FF techniques and the SLaK, improving the model's ability to handle multiscale data, particularly for tiny, small, and medium objects. These improvements not only increase the accuracy and effectiveness of remote sensing tools for forest management and biodiversity studies, but also support advanced applications like species classification within segmented tree crowns, which is crucial for carbon storage estimation and forest health monitoring.

Unlike the adjustments in [67], [68], which include tiny heads for detecting smaller objects, YOLOv8-FF uniquely omits the detection head for large objects, building upon this basis. This considerable departure from typical YOLO adaptations—traditionally designed to enhance capabilities across all sizes—is deliberately based on our dataset's unique composition, which primarily consists of tiny to medium canopies. This specific adaptation proves particularly effective for our dataset's needs, offering a focused and effective solution for remote sensing challenges and showcasing how targeted changes based on data traits can lead to substantial improvements. Moreover, Yi et al. [68] and Wang et al. [69] employed Bi-FPN within the YOLOv8 and YOLOv5 frameworks for remote sensing detection, focusing exclusively on same-scale FF. In contrast, YOLOv8-FF expands upon this by incorporating both same-scale and cross-scale fusion, introducing a new concept that enhances feature extraction capabilities across various scales. This adaptation also offers fresh insights for detection and segmentation tasks in the field of remote sensing. While previous methods, such as [70] applied RepLKNet [52], have explored the use of large kernel convolutions in YOLO models, the application of the SLaK structure in YOLO models for tree crown instance segmentation remains relatively unexplored. YOLOv8-FF introduces a new strategy to employing large kernel convolutions within the YOLO architecture, specifically tailored to enhance the segmentation of tree canopies. By customizing YOLOv8-FF to the specific traits of the dataset, the model's technical capabilities are enhanced, setting the stage for future advancements in environmental monitoring and forest management using deep CNN models.

D. Limitations

The YOLOv8-FF model, while effective in detecting and segmenting various tree species, still faces several challenges. First, the YOLOv8-FF model encounters challenges when dealing with densely populated and small tree crowns, such as those of the kānuka species. These conditions, characterized by overlapping and closely situated canopies, complicate the identification process and degrade performance. In addition, the proposed method records a slightly lower $\mathrm{AP_M}^{Mask}$ metric compared to the baseline YOLOv8 method. Despite some enhancements achieved by incorporating the SLaK structure, this issue reveals a trade-off between model specialization and overall accuracy. These limitations emphasize the need for continued research in the future work to enhance the robustness and adaptability of the YOLOv8-FF model.

VI. Conclusion

The primary objective of this article was to enhance the performance of YOLOv8 for the instance segmentation of individual tree crowns in aerial imagery. This goal has been successfully achieved with the development of the YOLOv8-FF method, which incorporates a redesigned network structure, an innovative FF method, and the integration of the SLaK module. By rethinking the network architecture to focus more effectively on tiny, small, and medium objects, the model has shown improved accuracy in recognizing tree crowns in aerial images. The FF method introduced in YOLOv8-FF effectively combines cross-scale and same-scale feature integrations, enhancing the model's ability to capture detailed and relevant features across different scales. In addition, the incorporation of the SLaK module expands the model's receptive field, particularly improving the detection of medium objects by utilizing larger convolutional kernels while maintaining lower model complexity. Experimental evaluations on the tree crown dataset, in comparison with 17 contemporary models, demonstrate that YOLOv8-FF consistently outperforms its competitors across most metrics, including various AP metrics, precision, recall, and F1-score. Notably, YOLOv8-FF achieves these results with fewer parameters compared to the baseline YOLOv8 model.

Looking forward, while YOLOv8-FF significantly advances the task of instance segmentation for tree crowns, challenges remain, particularly in segmenting objects with unclear boundaries such as overlapping or closely situated multispecies canopies. Future work will explore more sophisticated convolutional operations and enhanced data augmentation techniques to further improve the accuracy and efficiency of the model. In addition, the impact of removing detection head for large trees or objects could be investigated to evaluate how this modification affects performance. These advancements will aim to address the current limitations in canopy instance segmentation in aerial imagery.

Acknowledgment

The authors would like to thank Dr. Raphael Spiekermann for provisioning of the tree species classification data and Stella Belliss and Theresa Banning for hand-annotating the training labels.

References

[1] R. I. Spiekermann, F. van Zadelhoff, J. Schindler, H. G. Smith, C. Phillips, and M. Schwarz, "Comparing physical and statistical landslide susceptibility models at the scale of individual trees," *Geomorphology*, vol. 440, 2023, Art. no. 108870.

[2] L. Mou and X. Zhu, "Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6699–6711, Nov. 2018.

[3] Z. Ren, Y. Tang, Z. He, L. Tian, Y. Yang, and W. Zhang, "Ship detection in high-resolution optical remote sensing images aided by saliency information," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5623616.

[4] S. Jin et al., "Separating the structural components of maize for field phenotyping using terrestrial LiDAR data and deep convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2644–2658, Apr. 2020.

[5] A. Fujimoto et al., "An end to end process development for uav-sfm based forest monitoring: Individual tree detection, species classification and carbon dynamics simulation," *Forests*, vol. 10, no. 8, pp. 1–27, 2019.

[6] N. Saarinen et al., "Assessing biodiversity in boreal forests with uav-based photogrammetric point clouds and hyperspectral imaging," *Remote Sens.*, vol. 10, no. 2, pp. 1–22, 2018.

[7] G. Brūmelis, I. Dauškane, D. Elferts, L. Strode, T. Krama, and I. Krams, "Estimates of tree canopy closure and basal area as proxies for tree crown volume at a stand scale," *Forests*, vol. 11, no. 11, 2020, Art. no. 1180.

[8] S. Livesley, E. G. McPherson, and C. Calfapietra, "The urban forest and ecosystem services: Impacts on urban water, heat, and pollution cycles at the tree, street, and city scale," *J. Environ. Qual.*, vol. 45, no. 1, pp. 119–124, 2016.

[9] I. Shendryk, M. Broich, M. G. Tulbure, A. McGrath, D. Keith, and S. V. Alexandrov, "Mapping individual tree health using full-waveform airborne laser scans and imaging spectroscopy: A case study for a floodplain eucalypt forest," *Remote Sens. Environ.*, vol. 187, pp. 202–217, 2016.

[10] H. Zhao, J. Morgenroth, G. Pearse, and J. Schindler, "A systematic review of individual tree crown detection and delineation with convolutional neural networks (CNN)," *Curr. For. Rep.*, vol. 9, pp. 149–170, 2023.

[11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards realtime object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 39, pp. 1137–1149.

[14] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.

[15] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6409–6418.

[16] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4974–4983.

[17] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10213–10224.

[18] S. Li et al., "MogaNet: Multi-order gated aggregation network," in *Proc. Int. Conf. Learn. Representations*, 2024, pp. 1–35.

[19] W. Yu et al., "Metaformer is actually what you need for vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10819–10829.

[20] Y. Li et al., "Rethinking vision transformers for mobilenet size and speed," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16889–16900.

[21] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[23] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11976–11986.

[24] K. Chen et al., "Mmdetection: Open MMlab detection toolbox and benchmark," pp. 1–13, 2019, *arXiv: 1906.07155*.

[25] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9157–9166.

[26] E. Xie et al., "PolarMask: Single shot instance segmentation with polar representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12193–12202.

[27] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "Solo: Segmenting objects by locations," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 649–665.

[28] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic and fast instance segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 17721–17732.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[30] G. Jocher et al., "ultralytics/yolov5: V7. 0-YOLOv5 sota realtime instance segmentation," Zenodo (2022).

[31] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," pp. 1–17, 2022, *arXiv:2209.02976*.

[32] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7464–7475.

[33] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[34] C.-Y. Wang, I. H. Yeh, and H.-Y. M. Liao, "YOLOv 9: Learning what you want to learn using programmable gradient information," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 1–21.

[35] A. Wang et al., "YOLOv 10: Real-time end-to-end object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 37, pp. 107984–108011.

[36] Y. Fang et al., "Instances as queries," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6910–6919.

[37] Z. Tian, C. Shen, and C. Chen, "Conditional convolutions for instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 282–298.

[38] C. Lyu et al., "Rtmdet: An empirical study of designing real-time object detectors," pp. 1–15, 2022, *arXiv:2212.07784*.

[39] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[40] H. Zhang et al., "DINO: Detr with improved denoising anchor boxes for end-to-end object detection," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–19.

[41] F. Li et al., "Mask DINO: Towards a unified transformer-based framework for object detection and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3041–3050.

[42] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8573–8581.

[43] A. Sani-Mohammed, W. Yao, and M. Heurich, "Instance segmentation of standing dead trees in dense forest from aerial imagery using deep learning," *ISPRS Open J. Photogramm. Remote Sens.*, vol. 6, 2022, Art. no. 100024.

[44] Z. Sun, B. Xue, M. Zhang, and J. Schindler, "An improved mask R-CNN for instance segmentation of tree crowns in aerial imagery," in *Proc. Int. Conf. Image Vis. Comput. New Zeal.*, 2023, pp. 1–6.

[45] Y. Sun, Z. Li, H. He, L. Guo, X. Zhang, and Q. Xin, "Counting trees in a subtropical mega city using the instance segmentation method," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 106, 2022, Art. no. 102662.

[46] A. Firoze, C. Wingren, R. A. Yeh, B. Benes, and D. Aliaga, "Tree instance segmentation with temporal contour graph," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2193–2202.

[47] S. Dersch, A. Schöttl, P. Krzystek, and M. Heurich, "Towards complete tree crown delineation by instance segmentation with Mask R–CNN and DETR using UAV-based multispectral imagery and lidar data," *ISPRS Open J. Photogramm. Remote Sens.*, vol. 8, 2023, Art. no. 100037.

[48] J. Zhou et al., "Multispecies individual tree crown extraction and classification based on BlendMask and high-resolution UAV images," *J. Appl. Remote Sens.*, vol. 17, no. 1, 2023, Art. no. 16503.

[49] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. workshops*, 2020, pp. 390–391.

[50] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10781–10790.

[51] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[52] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11963–11975.

[53] S. Liu et al., "More convNets in the 2020s: Scaling up kernels beyond 51x51 using sparsity," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–23.

[54] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," pp. 1–10, 2016, *arXiv:1606.08415*.

[55] A. Straker et al., "Instance segmentation of individual tree crowns with YOLOv5: A comparison of approaches using the forinstance benchmark LiDAR dataset," *ISPRS Open J. Photogramm. Remote Sens.*, vol. 9, 2023, Art. no. 100045.

[56] T.-Y. Lin et al., "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[58] M.-T. Pham, L. Courtrai, C. Friguet, S. Lefévre, and A. Baussard, "YOLO-Fine: One-stage detector of small objects under various backgrounds in remote sensing images," *Remote Sens.*, vol. 12, no. 15, 2020, Art. no. 2501.

[59] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *Proc. Int. Conf. Syst., Signals Image Process.*, 2020, pp. 237–242.

[60] R. I. Spiekermann, S. McColl, I. Fuller, J. Dymond, L. Burkitt, and H. G. Smith, "Quantifying the influence of individual trees on slope stability at landscape scale," *J. Environ. Manag.*, vol. 286, 2021, Art. no. 112194.

[61] A. Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. Int. Conf., Mach. Learn.*, 2019, pp. 242–252.

[62] M. Beloiu, L. Heinzmann, N. Rehush, A. Gessler, and V. C. Griess, "Individual tree-crown detection and species identification in heterogeneous forests using aerial RGB imagery and deep learning," *Remote Sens.*, vol. 15, no. 5, 2023, Art. no. 1463.
[63] A. I. Pleoianu, M. S. Stupariu, I. andric, I. Pătru-Stupariu, and L. Drăgu, "Individual tree-crown detection and species classification in very high-resolution remote sensing imagery using a deep learning ensemble model," *Remote Sens.*, vol. 12, no. 15, 2020, Art. no. 2426.
[64] A. J. Chadwick, N. C. Coops, C. W. Bater, L. A. Martens, and B. White, "Transferability of a Mask R–CNN model for the delineation and classification of two species of regenerating tree crowns to untrained sites," *Sci. Remote Sens.*, vol. 9, 2024, Art. no. 100109.
[65] J. Mo et al., "Deep learning-based instance segmentation method of litchi canopy from UAV-acquired images," *Remote Sens.*, vol. 13, no. 19, 2021, Art. no. 3919.
[66] Y. J. La, D. Seo, D. Kang, M. Kim, T. W. Yoo, and I. S. Oh, "Deep learning-based segmentation of intertwined fruit trees for agricultural tasks," *Agriculture*, vol. 13, no. 11, 2023, Art. no. 2097.
[67] W. Liu, K. Quijano, and M. M. Crawford, "YOLOv5-Tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 8085–8094, Sep. 2022.
[68] H. Yi, B. Liu, B. Zhao, and E. Liu, "Small object detection algorithm based on omproved YOLOv8 for remote sensing," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 1734–1747, Dec. 2024.
[69] L. Wang, J. Cai, T. Wang, J. Zhao, T. R. Gadekallu, and K. Fang, "Pine wilt disease detection based on UAV remote sensing with an improved YOLO model," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 19230–19242, Oct. 2024.
[70] K. Xia et al., "Mixed receptive fields augmented YOLO with multi-path spatial pyramid pooling for steel surface defect detection," *Sensors*, vol. 23, no. 11, 2023, Art. no. 5114.

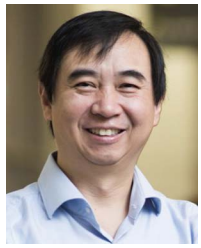
Ziyi Sun received the B.E. and M.E. degrees in computer science and technology from the Shandong University of Finance and Economics, Shandong, China, in 2018 and 2021, respectively. She is currently working toward the Ph.D. degree in computer science with the Victoria University of Wellington, Wellington, New Zealand.

Her research interests include deep learning, computer vision, and image segmentation.

Bing Xue (Fellow, IEEE) received the B.Sc. degree in information management and information systems from the Henan University of Economics and Law, Zhengzhou, China, in 2007, the M.Sc. degree in management from Shenzhen University, Shenzhen, China, in 2010, and the Ph.D. degree in computer science from VUW, New Zealand, in 2014.

She is currently Professor of Artificial Intelligence, Deputy Head of School with the School of Engineering and Computer Science, Deputy Director of Center for Data Science and Artificial Intelligence with VUW. She has more than 400 papers published in fully refereed international journals and conferences and her research focuses mainly on evolutionary computation, machine learning, classification, symbolic regression, feature selection, evolving deep NNs, image analysis, transfer learning, multiobjective machine learning.

Prof. Xue has been organizing many international conferences, such as General Chair of PRICAI 2025, Conference Chair of IEEE CEC 2024 and EuroGP 2024. She has also served as an Associate Editor of several international journals, such as IEEE TEVC, IEEE TAI, IEEE CIM, and ACM TELO. She is a Fellow of Engineering New Zealand.



Mengjie Zhang (Fellow, IEEE) received the Ph.D. degree in computer science from RMIT University, Melbourne, VIC, Australia, in 2000.

He is currently a Professor of Computer Science, the Director of Centre for Data Science and Artificial Intelligence, Victoria University of Wellington, New Zealand. His current research interests include evolutionary machine learning, genetic programming, image analysis, feature selection and reduction, job shop scheduling, and evolutionary deep learning and transfer learning. He has published over 800 research papers in refereed international journals and conferences.

Prof. Zhang is a Fellow of the Royal Society of New Zealand, a Fellow of Engineering New Zealand, and an IEEE Distinguished Lecturer.

Jan Schindler received the B.Sc. degree in physical geography from the University of Munich, Munich, Germany, in 2010 and the M.Sc. degree in Earth observation from KU Leuven, Leuven, Belgium, in 2012. and the Ph.D. degree in atmospheric physics from the University of Mainz, Max Planck Institute for Chemistry, Germany, in 2017.

He is currently a Senior Remote Sensing Researcher with Manaaki Whenua - Landcare Research, New Zealand. His work focuses on using remote sensing and machine learning techniques to improve land cover and vegetation mapping across spatial scales from 2D multi-spectral imagery and 3D point clouds from aerial and mobile laser scanning. He also co-supervises postgraduate students at the University of Canterbury in Christchurch, New Zealand, and Victoria University of Wellington, New Zealand.