# Using Machine Learning Methods to Estimate Spruce tree crown and DBH from Aerial Imagery

A Thesis

Submitted to the Faculty of Graduate Studies and Research

in Partial Fulfilment of the Requirements

for the Degree of

Master of Applied Science

in

Industrial Systems Engineering

University of Regina

by

Hamza Zouaghi

Regina, Saskatchewan

December, 2024

# UNIVERSITY OF REGINA

## FACULTY OF GRADUATE STUDIES AND RESEARCH

## SUPERVISORY AND EXAMINING COMMITTEE

**Hamza Zouaghi**, candidate for the degree of **Master of Applied Science in Industrial Systems Engineering,** has presented a thesis titled, ***Using machine learning methods to estimate spruce tree crown and DBH from aerial imagery,*** in an oral examination held on **December 16, 2024**. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

| | |
|---|---|
| External Examiner: | Dr. Zhanle (Gerald) Wang, Electronic Systems Engineering |
| Supervisor(s): | Dr. Wei Peng, Engineering General |
| Committee Member: | Dr. Mohammad Khondoker, Industrial Systems Engineering |
| Committee Member: | Dr. Sharfuddin Khan, Industrial Systems Engineering |
| Chair of Defense: | Dr. Xia Ji, Faculty of Education |

# Abstract

Spruce trees play a vital role in Canada's forest ecosystems, which is widely used in construction, paper production, and other industries. However, spruce trees are particularly susceptible to wildfires, which pose a major risk to both natural landscapes and human settlements. Therefore, to evaluate the spruce forest biomass volume is an important step to estimate its yield and combustibility. This paper aims to use Machine Learning (ML) approaches to estimate the biomass volume of spruce trees from aerial top-view images. Since the aerial images are only show the tree crown shapes, we set up the relationship between tree crown diameter (TCD) and Diameter of tree Breast Height (DBH), and this DBH can be further used to estimate the tree biomass volume. Here, Spruce trees top-view images were taken by a DJI Mavic 3, at an altitude of 50m above ground. We measured the actual TCD and DBH in the field. 2,155 spruce trees were labelled in our dataset according to its location in the aerial images. The actual TCD values of 2155 samples were measured with a Hypsometer device that uses ultrasound at the extremities of the tree branches, which will be further used to compare and calculate the accuracy of the TCD values that are measured from top-view images after our model training.

After experimenting with tree detection methods, we conclude that YOLO performed better than MaskRCNN by 4%. And then we proposed two methods that use YOLOs: First method, a combination of YOLOv5 bounding box to identify the trees and watershed technique to segment tree crowns from aerial images. Compared to the second method YOLOv11 that uses instance segmentation to segment the trees.

A study is conducted to showcase a relationship between TCD and DBH of the field measurements. This linear relationship can be used to estimate DBH out of TCD and then could futher calculate tree biomass volume with the estimated DBH.

We conclude that Instance Segmentation model can estimate TCD with an accuracy of 91.64% (compared to the actual TCD value), which is higher than 89.1% of another image processing model based on watershed technique method.

**Key words:** YOLOv11, YOLOv5, Watershed Technique, Identification, Segmentation, TCD, DBH, Spruce Trees, Aerial Images.

# Acknowledgements

I am very grateful to Dr. Wei Peng, my supervisor, who assisted with the tech materials and advice needed for this thesis. I also want to thank Dr. Mark Vanderwel, who shared his expertise in Tree demography and provided us with the materials needed for measuring the trees. I would like to thank my project team: Mahdi Mohemi Moshkenani, who helped us to provide details explanations and ask the right questions to solve some difficult problems. And Axl Lopez Rodriguez-Malpica, who was talented with fast coding from scratch skills.

I also acknowledge all FGSR funding and scholarship which helped me focus on my thesis and research.

# Table of Contents

# List of Tables

# List of Figures

# List of Appendices

# List of Abbreviations

| | |
|---|---|
| DBH | Diameter at Breast Height |
| TCD | Tree Crown Diameter |
| WST | Watershed Technique |
| NMS | Non-Maximum Suppression |
| Mask RCNN | Mask Region-based Convolutional Neural Networks |

# Transparency Statement

I used ChatGPT to produce article findings (methods and accuracy). I combined these summaries to produce a draft of the literature review in Chapter 2. However, I improved the generated literature review using my own wordings to make it simple to understand and my wordings for the interesting papers that peeked my interest.

My supervisor(s) and supervisory committee have approved the use of the above technologies for the described purposes. I confirm that no AI-technologies other than those listed above have been used to prepare this thesis. I acknowledge that AI-technologies may produce output that is biased, discriminatory, incomplete, or inaccurate and that I have taken the necessary steps to address this. I acknowledge that I am solely responsible for maintaining the accuracy and academic integrity of this thesis.

# Chapter One: Introduction

## 1.1. Statement of the Problem.

Spruce trees play a vital role in Canada’s forest ecosystems and economy, serving as both an ecological keystone species and a major contributor to the forestry industry. As one of the most abundant tree species in Canada, spruce trees: black spruce (Picea mariana) and white spruce (Picea glauca) are highly valued for their versatile and durable wood, which finds extensive use in the construction of homes, furniture, and industrial applications. Additionally, spruce wood is a primary raw material in paper, which contributes significantly to Canada’s economy by supporting manufacturing industries and generating employment in rural areas.

Despite their many benefits, spruce trees face significant challenges in maintaining their health and resilience. They are particularly vulnerable to wildfires, which have become more frequent and intense in recent years due to climate change. The resinous nature of spruce trees makes them highly flammable, allowing fires to spread rapidly through spruce forests, posing a significant threat to both natural landscapes and human settlements. This problem can be avoided by controlling the forestry which represents in form of cutting trees around a high flammable risk area. In addition to wildfires, spruce trees are susceptible to pests such as the spruce beetle, further endangering forest health and reducing timber quality.

The importance of monitoring and managing spruce forests has therefore become increasingly urgent. Forestry professionals and researchers are developing strategies to predict wildfire risks, manage pest outbreaks, and promote sustainable forestry practices. Through a combination of technological advancements and conservation efforts, Canada aims to protect its spruce forests, ensuring they continue to contribute both economically and ecologically for future generations.

## 1.2. Research objective

The main objective in this thesis is to use Machine Learning (ML) approaches to estimate the biomass volume of spruce trees from aerial top-view images. There are three sub-objectives, which are:

- Setting up the relationship between tree crown diameter (TCD) and Diameter of tree Breast Height (DBH),
- Building the relationship between DBH and tree biomass volume.
- Developing an image processing model based on this YOLOv11 method to identify the TCD values from aerial images.
- Comparing the developed model to another image processing model of YOLOv5+Watershed Technique.

Accurately measuring the crown area and diameter is essential because it enables the estimation of other key parameters, such as tree height, diameter at breast height (DBH), and dry biomass volume. These parameters play a critical role in a variety of applications, including biomass estimation, carbon sequestration assessments, and forest health monitoring.

By accurately predicting the DBH from crown dimensions, forest managers can estimate timber volume, plan a sustainable harvesting practice, and monitor changes in forest structure over time. This thesis focuses on developing advanced computer vision machine learning models to enhance the precision of tree crown diameter and area calculations. By leveraging tools such as watershed technique and YOLO instance segmentation, this work aims to provide a robust framework for tree crown detection and measurement. The ultimate goal is to offer a solution that not only improves the accuracy of crown measurements but also facilitates the estimation of other critical tree metrics ensuring that these models can support forest inventory, conservation efforts, and ecosystem management effectively.

### 1.3. Structure of this Thesis

This thesis is organized into Three main chapters, each addressing a specific component of the research.

Chapter Two: Literature Review provides an overview of relevant studies and concepts. It begins by discussing spruce tree allometric relationships and existing research on tree detection and measurement. The chapter identifies research gaps that this study aims to address, followed by a review of the YOLO model, which plays a central role in the proposed methodology. The chapter concludes with a summary highlighting key points from the literature.

Chapter Three: Methodology outlines the research design and approach. It covers the materials and data acquisition process, followed by detailed steps in data preparation. This chapter also explains the YOLOv5 model, the application of watershed segmentation and Non-Maximum Suppression (NMS) for contour adjustments, and methods for crown diameter estimation using GSD value. Finally, we explain the YOLOv11 model and concludes with a summary of the methodology.

Chapter Four: Results and Findings presents the outcomes of the research. It begins with the results of the spruce tree detection models, followed by the estimation of tree crown diameter (TCD). The chapter then explores Diameter at Breast Height (DBH) estimation from TCD using machine learning models. The key findings are summarized at the end of the chapter.

Chapter Five: Discussion and Conclusion provides a detailed discussion of the results and also highlights limitations of the study and suggests directions for future research.

# Chapter Two: Literature Review

Accurately estimating tree crown diameter (TCD) and other related metrics, such as the diameter at breast height (DBH), is crucial for biomass estimation. Therefore, a literature review is needed to know the allometric relationships and equations of the spruce tree. Another literature review to consider is about the crown segmentation methods which are varied from different types of datasets and segmentation models.

By analyzing existing methods and identifying research gaps, this chapter sets to push the boundaries of tree crown segmentation and measurement techniques, offering a refined approach to estimating spruce tree metrics from aerial images.

## 2.1 Spruce Tree Allometric Relationships papers

There is few research about the relationship between Spruce trees metrics. We noticed that many of the research is dated between 1993 and 2010. And we suspect that the allometric relationship may change and this is due to global warming, soil quality change, and the location of the study. However, we need to conduct this literature review to know how the forester's scientist took their measures and how they quantify it into a regression model.

The relationship between Diameter at Breast Height (DBH) and dry biomass is studied by Gower et al. (1993) mentioned in [1] was the best example that we will compare. Widlowski et al [1] made a comprehensive direct result of the allometries. Gower et al. (1993) in the paper, developed this equation, which is based on DBH, to estimate the biomass in kilograms per tree.

$$W_s = 0.105196 \cdot DBH^{2.310} \quad [\text{kg tree}^{-1}] \quad (r^2 = 0.975) \qquad \text{(Equation 2-1)}$$

$$W_{lb} = 0.011350 \cdot DBH^{2.570} \quad [\text{kg tree}^{-1}] \quad (r^2 = 0.976) \qquad \text{(Equation 2-2)}$$

$$W_{db} = 0.027669 \cdot DBH^{2.226} \quad [\text{kg tree}^{-1}] \quad (r^2 = 0.765) \qquad \text{(Equation 2-3)}$$

$$W_f = 0.029174 \cdot DBH^{2.292} \quad [\text{kg tree}^{-1}] \quad (r^2 = 0.960) \qquad \text{(Equation 2-4)}$$

where:

- $W_s$ = Stem Dry Weight
- $W_{lb}$ = Live Branch Dry Weight
- $W_{db}$ = Dead Branch Dry Weight
- $W_f$ = Foliage Dry Weight

The total tree biomass is given by:

$$W_t = W_s + W_{lb} + W_{db} + W_f \qquad \text{(Equation 2-5)}$$

Please note that we can’t verify the dry biomass ourselves in the dataset. Because of this limitation, we can rely on this equation to calculate the biomass.

On the other side, we verify if the relationship between crown diameter and DBH that we obtained matches the equation developed by Nagel et al. (2002) mentioned in [1] below:

$$C_r = (0.6122 + 0.0536 \cdot DBH) \quad [\text{m}] \qquad \text{(Equation 2-6)}$$

where:

- $C_r$ = Crown diameter
- $DBH$ = Diameter at Breast Height

This equation can be compared to our results and verify if our measurements are true or not. Although, these equations came from Norway Spruce trees, we believe that White and blue Spruce share similar features. However, the results later on in chapter 4 shows that they are different and need different allometric approach.

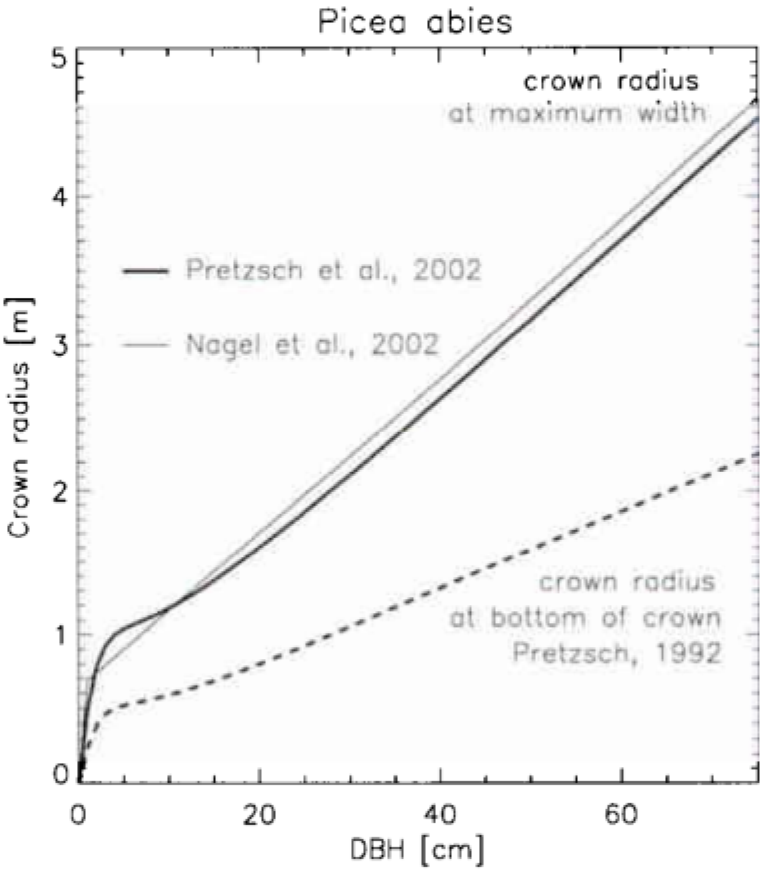


**Figure 1**: Explains the relationship between DBH and crown radius Cr for Norway spruce.

The figure above will be used to compare our results and see if we are in the right direction or not. This will increase the credibility of our measurements of DBH and Tree Crown Diameter TCD. The trend should be similar to this graph. We choose this paper, mainly because we can use the previous findings and see if the trees are affected by weather, soil, location of the studied spruce trees.

There are some other papers that tackle this issue concerning spruce trees. Sharma RP et al. [2] shows that crown dimensions correlate with tree growth and are used as predictors in growth models. The crown-to-bole diameter ratio (CDBDR), which compares crown width to DBH, was modeled using data from Norway spruce and European beech stands, achieving R² values of 0.73

(spruce) and 0.78 (beech) with a spatially explicit mixed-effects model, outperforming the non-spatial model with $R^2$ values of 0.71 (spruce) and 0.76 (beech). F Colin et al. [3] proposes a new method for assessing wood quality using regional inventory data and includes a wood quality simulation software that requires detailed input on tree branchiness and morphology. The study focuses on 117 Norway spruce trees sampled in northeastern France and constructs models to predict key crown characteristics, such as crown position, size, insertion angle, and branch whorls, based on DBH, total height, age, and growth unit position along the stem. Filipescu CN et al. [4] highlights how remote sensing advancements provide detailed forest information efficiently, but individual tree stem diameter cannot be directly measured from remote data and must instead be estimated using tree crown, height, and stand structure attributes. Predictive models using a nonlinear mixed-effects approach were developed for black spruce, Douglas-fir, and lodgepole pine, achieving root mean squared error (RMSE) values between 0.9 and 1.8 cm, representing 10–11% of the mean diameter. The study emphasizes the importance of including crown area and spacing/thinning variables to enhance the predictive performance of diameter models. Hall RJ et al. [5] studied on the performance of 12 diameter prediction models, which included linear and logarithmic transformations of tree height and crown area. The models were tested for several species, including white spruce, lodgepole pine, trembling aspen, and balsam poplar. While all models were statistically significant, their performance varied across species and model forms. Two models were identified for further investigation. The study found that tree height was generally more correlated with DBH than crown area, except for lodgepole pine. Additionally, it highlighted that using both tree height and crown area improves prediction accuracy but increases the measurement cost significantly, from \$10.29 to \$17.50 per plot.

Koji Shimano [6] explored the relationship between DBH and crown projection area (CPA) for both deciduous and coniferous trees using several models, including a newly proposed one. Among the four models tested, the power-sigmoid function proved to be the most suitable due to its good fit and mechanistic meaning. This model demonstrates that CPA increases with the second power of DBH, but the rate of increase slows as DBH continues to grow. The power-sigmoid function can also be transformed to describe individual basal area (IBA)-CPA relations as a single-saturate function, showing high compatibility between the two approaches.

When analyzing the differences in DBH-CPA relationships between deciduous and coniferous trees, it was observed that both groups had similar initial CPA growth rates. However, CPA growth slowed earlier for coniferous trees compared to deciduous trees. The power-sigmoid function provides a meaningful way to examine the DBH-CPA relationship, allowing researchers to analyze both the initial CPA growth rate and the final form of the tree independently.

Samantha J. et al. [7] focused on developing models for tree crown radius for several conifer species in California, using typical forest inventory variables such as DBH, tree height, height-to-crown base, crown class, basal area per hectare, and trees per hectare. The models were fitted using both ordinary and weighted least squares methods. For most species, the ordinary least squares linear regression with DBH as the sole independent variable was found to be sufficient. While adding other independent variables provided some minor improvements, the impact was limited.

Turan Sönmez [8] tested seven models for predicting crown diameter using DBH for Picea orientalis L. (Link.) in Artvin, based on data from 4,208 trees measured across 117 temporary sample plots. The sample plots were categorized by 20-year mean age class intervals and further grouped by 2 cm DBH intervals. Statistical analysis was conducted on the mean DBH and crown diameter across age classes.

Among the seven models, one linear and six non-linear functions were tested to model the DBH-crown diameter relationship. The regression analysis indicated a statistically significant ($P > 0.05$) and strong relationship ($R^2 > 0.80$) between DBH and crown diameter. Several models—cubic, compound, growth, and exponential—were found to have similar $R^2$ values. However, the cubic model provided the best fit, demonstrating the strongest predictive capability for the relationship between DBH and crown diameter. Vezina PE [9] found a strong correlation between crown width and diameter at breast height (DBH) for open-grown balsam fir and white spruce. Using this relationship, Crown Competition Factor (CCF) values were applied to even-aged balsam fir stands across various ages and site indices to evaluate their relationship with other stand density measures. The preliminary results suggest that CCF is an effective metric for expressing stand density in even-aged balsam fir stands. However, the utility of CCF will become even more significant if it can be successfully linked with growth and yield, providing a more comprehensive understanding of forest stand productivity.

## 2.2 Existing Research on TCD detection

Several papers in 2024 have attempted to estimate the diameter of the tree crown by two measurement methods. The first one is called LIDAR technology, which is expensive equipment but produces high quality dataset and has the highest accuracy. Here are some papers that focus on this method:

**Table 1**: Research Papers using LIDAR Technology

| Author | method | Results |
|---|---|---|
| Solares et al. [10] | CAE (Convolutional Autoencoder) | (P%) = 98.9 |

| | | |
|---|---|---|
| Yang et al. [11] | Point Transformer deep learning network | Intersection over Union (IoU) of 96.0% |
| Zhu et al. [12] | segmentation based on hierarchical strategy (SHS) (which achieved precision = 1) | extracted crown diameters with $R^2$ values of 0.91 |
| Fu Y et al. [13] | multiscale adaptive local maximum filter for treetop detection, the Dalponte region-growing method for crown delineation, and mean-shift voxelization with supervoxel-weighted fuzzy c-means clustering to refine crown boundaries | Accuracy = 87.28% |
| Deng S et al. [14] | trunk point distribution indicator (TPDI) for identifying potential tree trunk positions, followed by RANSAC-based 3D line fitting to differentiate trunks from understory vegetation, and a region-growing segmentation method refined through crown shape and vertical profile analysis | F-scores ranging from 0.920 to 1.000 across 12 plots |
| Deng S et al. [15] | combining trunk point distribution indicators with treetop extraction from canopy height models (CHM) and marker-controlled watershed segmentation to refine trunk positions and differentiate true and false detections through 3D trunk and branch analysis | F-scores ranging from 0.723 to 0.829 across three plots |

| | | |
|---|---|---|
| Yu J et al. [16] | adaptive crown-shaped algorithm for individual tree segmentation and applying a region-growing method | individual tree segmentation accuracy = 87.7% |
| Dersch S et al. [17] | Mask R-CNN with the normalized cut clustering method to segment and classify individual trees | F1 scores of 79% |
| Zhang C et al. [18] | hierarchical filtering and clustering (HFC) algorithm using intensity filtering, SVD filtering, SOR, and clustering techniques | F1-score variations of 1–3 percentage points |
| Chen Q et al. [19] | bottom-up ITDS framework based on DBSCAN clustering for initial trunk detection, KNN reclassification of non-core points, RANSAC cylinder fitting for trunk correction, and centroid-based seed point calculation for individual tree segmentation (ITS) | overall recall of 95.2%, precision of 97.4%, and an F-score of 0.96 |
| Burmeister et al. [20] | coarse-to-fine algorithm for tree instance segmentation, combining the marker-controlled Watershed algorithm for coarse segmentation with 3D region growing for refinement in areas with overlapping crowns, along with Voronoi segmentation-based error removal | Precision = 83.9 % |
| Cheng D et al. [21] | TreeScope v1.0, a robotics dataset for precision agriculture and forestry, providing LiDAR data collected by UAVs, mobile robots, and human- | Overall accuracy = 95.5% |

| | | |
|---|---|---|
| | operated platforms along with 1,800 manually annotated semantic labels and field-measured tree diameters, aiming to improve tree counting, mapping, and diameter estimation | |
| Wang L et al. [22] | CHM segmentation,<br>point cloud clustering segmentation,<br>and layer stacking fitting segmentation | The point cloud clustering segmentation algorithm achieved the highest segmentation accuracy at 93%, followed by CHM segmentation at 88% and layer stacking fitting segmentation at 84% |
| Saeed T et al. [23] | 3 × 3m fixed window size method on unsmoothed CHM<br>Dalponte 2016 method<br>AMS3D algorithm | AMS3D algorithm achieved the highest F-score of 0.67 with crown radii errors within 0.1 m |
| Li Q et al. [24] | deep learning-based street tree segmentation method that maps 3D MLS point clouds to 2D RGB images for segmentation using algorithms like YOLOv8, YOLACT, and BlendMask, and then | The YOLOv8 model achieved the best performance with an IoU range of |

| | | |
|---|---|---|
| | maps the 2D segmentation mask back to the 3D point cloud to generate and optimize tree proposals | 0.85:0.05:0.95, a precision of 0.9988, recall of 0.9986, F1-score of 0.9987, and a segmentation time of 26 ms per image and 4.05 ms per scanline |
| Seidl J et al. [25] | DBSCAN clustering combined with graph theory, where clusters are connected into a 3D graph | The segmentation accuracy, compared with manual labels collected from orthophoto images, ranged between 82% and 93% |
| Terekhov V et al. [26] | specialized tree clustering algorithm | Accuracy not clear |
| Kurdi FT et al. [27] | rotating surface simulations of segmented tree point clouds, calculating X, Y, and Z matrices based on the tree's projection on the horizontal plane to enable modeling, visualization, and geometric parameter calculation | fit between 0.3 and 0.89 m |

| | | |
|---|---|---|
| Tarsha Kurdi et al. [28] | algorithm for 3D modeling of tree trunks using laser scanning point clouds from ALS and TLS data | achieved a modeling accuracy better than 4 cm |
| Zhu Y et al. [29] | feature extraction module of PointNet++ to detect understory branches and employing a graph-based segmentation algorithm that constructs a directed acyclic graph from grey image clustering components to segment individual tree crowns | PointNet++ achieved an average recall of 94.6% for tree trunk detection and precision of 96.2% |
| Shao J et al. [30] | Mobile Laser Scanning (MLS) and deep learning, with the ForestSPG model performing large-scale semantic segmentation on LiDAR data to map individual stems and measure Diameter at Breast Height (DBH) | Backpack LiDAR achieved an RMSE of 1.82 cm and UAV LiDAR achieved an RMSE of 3.13 cm for DBH measurements on trees with DBH greater than 38.1 cm (15 in) |
| Fallah M et al. [31] | multi-scale individual tree detection (MSITD) algorithm, which combines raster-based and point-based approaches for accurate tree detection from LiDAR data, and uses the SAFER semi-supervised regression algorithm to estimate Diameter at Breast Height (DBH) and Aboveground Biomass (AGB) | MSITD algorithm improved extraction and matching rates by 13%.<br>For DBH, SAFER achieved an RMSE of |

| | | |
|---|---|---|
| | | 3.38 cm, MAE of 2.84 cm, and $R^2$ of 0.59 |
| Wang Z et al. [32] | Multilevel Intuitive Attention Network (MIA-Net) for point cloud segmentation, utilizing local trigonometric encoding for fine-grained semantics, feature sampling with a point offset mechanism, and intuitive attention interaction to capture both local and global features efficiently | Overall Accuracy (OA) of 96.2% |
| Tang S et al. [33] | generating a structured 3D synthetic tree dataset of 13,000 tree models from ten common species, including detailed point clouds with hierarchical structures, branch-leaf separation, and tree skeleton information for enhanced testing and evaluation. | validated through airborne laser scanning simulations and the application of state-of-the-art algorithms for tasks such as 3D reconstruction, individual tree segmentation |
| Sun H et al. [34] | improved PointNeXt model that fuses ALS and TLS point cloud data, employs a Mean Shift algorithm with KdTree to remove gaps, and calculates canopy volume through convex hull area summation | mean IoU of 98.19%, a RMSE of 0.18 $m^3$, and an $R^2$ value of 0.92 |

| | | |
|---|---|---|
| Mukhandi H et al. [35] | SyS3DS (Systematic Sampling for 3D Semantic Segmentation) | Overall accuracy = 91.3% |
| Wielgosz M et al. [36] | 3D CNN architecture inspired by PointGroup | The model outperformed Point2Tree and TLS2trees by 20-30% in detection, omission |
| Yan Y et al. [37] | watershed algorithm and CHM-based methods | F-score of 0.761 |

Although LIDAR technology is known for its high costs, researchers have sought to reduce expenses by utilizing aerial drones to capture data from a top-down perspective, operating within a 2D plane. Modern drones are equipped with advanced sensors that provide highly accurate measurements of height, location coordinates, camera angle, and image quality. This precision enables researchers to gather reliable data. By using computer vision techniques and deep learning models, researchers have been able to segment tree with high accuracy, making drone-based methods a good alternative to traditional LIDAR for this application. Below is the table of recent studies on top view-based and side view-based dataset:

**Table 2**: Image Segmentation Research Papers Based on Top and Side Views

| Author | Method | Results |
|---|---|---|
| Condat R et al. [38] | ConvexMask, a convolutional neural network (CNN) for real-time tree instance segmentation, using a label | Accuracy = 68.4% |

| | | |
|---|---|---|
| | representation approach that combines a convex exterior polygon to define tree extremities and a binary mask to handle occlusions and details | |
| Guo Z et al. [39] | ATT-MRCNN model, which integrates channel and spatial attention mechanisms with Mask RCNN for target detection and identification of citrus images, using transfer learning to optimize training efficiency and parameter initialization | recognition rate exceeding 95% |
| Lavania P et al. [40] | U-Net-based deep learning model for semantic segmentation | accuracy of 84–89%, an F1 score of 0.91–0.94, and an Intersection over Union (IoU) of 0.79–0.89 |
| Chen J et al. [41] | YOLOv8 for apple tree organ segmentation during the bud stage by integrating ConvNeXt V2, Multi-scale Extended Attention Module (MSDA), and Dynamic Snake Convolution (DSConv) to improve the network's ability to extract contextual information | 82.58% mean Precision (mP), 74.58% mean Recall (mR), 77.94% mean Dice (mDice), 64.91% mean IoU (mIoU), and 79.75% mean Average Precision (mAP) |
| Bu X et al. [42] | DFSNet (Dynamic Fusion Segmentation Network) | accuracy rate of 89.43% and mIoU of 74.05% |

| | | |
|---|---|---|
| Liu H et al. [43] | cluster transformer for the encoder, which includes a cluster token mixer and spatial-channel feed-forward network (SC-FFN) to reduce redundant information and extract multiscale spatial and channel data. The D-cluster transformer in the decoder transfers robust features without traditional upsampling methods | accuracy rate of 63.15% |
| Steier J et al. [44] | manual tree crown annotations used for training deep learning models in forest stand mapping, comparing them against tree reference data from an official tree register and UAV laser scanning (ULS) segments at two study sites | manual annotations correctly detected only 37% of tree crowns in the forest-like plantation and 10% in the natural forest |
| Arakawa T et al. [45] | YOLOv4 object detection algorithm | R$^2$ value of 0.98 |
| Moysiadis V et al. [46] | machine learning-based tree detection method using Detectron2 and YOLOv8 to isolate individual trees in an orchard and generate tree masks, followed by OTSU thresholding to improve crown coverage | F1-Score of 94.85% for cherry tree detection and IoU of 85.30% for crown extraction. |

| | | |
|---|---|---|
| Wu W et al. [47] | DR-YOLO instance segmentation algorithm | DR-YOLO model achieved a 2.0% improvement in AP@0.5 and 1.1% improvement in Precision over YOLOv8-seg |
| Zhang Y et al. [48] | mask R-CNN | average overall accuracy of 0.953 |
| Xu J et al. [49] | BlendMask algorithm to accurately segment tree crowns and introduced a Bayesian neural network to model the relationship between tree crown size and diameter at breast height (DBH) | BlendMask algorithm achieved an accuracy of 0.893, compared to 0.721 for the traditional watershed algorithm. The model's DBH predictions for Ginkgo biloba, Pinus tabuliformis, and Populus nigra showed average discrepancies of 0.15 cm, 0.29 cm, and 0.49 cm |
| Mai Y et al. [50] | Taoism-Net, a minimalist deep learning model designed for real-time, pixel-level segmentation | 4.8% in mIoU |
| Vasavi S et al. [51] | semantic segmentation, U-Net with ResNet34 as its backbone | accuracy of 92% in detecting tree canopy and an accuracy of 84% in classifying the objects |
| Cloutier M et al. [52] | Convolutional Neural Network (CNN) | F1-score of 0.72 |

| | | |
|---|---|---|
| Zhang H et al. [53] | Double-Branch Multi-Scale Contextual Network (DB-MSC Net) | overall accuracy (OA) was improved by at least 0.16% compared to previous methods |
| Nashat AA et al. [54] | instance segmentation algorithm based on transfer learning using the Mask RCNN architecture with ResNet101 as the backbone and Feature Pyramid Network (FPN) for feature extraction | test accuracy of 97.76% and mAP@50:95 of 100% |
| Lin Y et al. [55] | YOLO and Mask R-CNN | Accuracy of YOLOv9 on branch detection = 98.8% |
| Zhang C et al. [56] | Mask R-CNN model with ResNeXt-50 as the backbone network | Accuracy of 92.2% |
| Wang J et al. [57] | YOLO-DCAM | precision of 96.1%, recall of 93.0% |
| Prousalidis K et al. [58] | YOLOv8n, RepViT-SAM, and EdgeSAM | YOLOv8n outperformed other models by 95% IoU |
| Khan Z et al. [59] | YOLOv8 | mean average precision (mAP) of 93.3%, a precision (P) of 93.6% |
| Zhu F et al. [60] | CEDAnet | bounding box AP of 0.498 and a segmentation AP of 0.493 on the iSCHID dataset, and a bounding box AP of 0.706 and a segmentation AP of 0.703 on the iSMMID dataset |

| | | |
|---|---|---|
| Yang T et al. [61] | YOLO and SegNet | mean intersection over union (mIoU) of 92.0%, mean pixel accuracy (mPA) of 95.9% |
| Xie Y et al. [62] | Mask R-CNN with ResNet50, ResNet101, and ResNeXt101 for instance segmentation | Bounding Box Average Precision (Box-AP) of 51.697% and Segmentation Average Precision (Segm-AP) of 54.946% |
| Deka B et al. [63] | modified Mask R-CNN instance segmentation model with a channel attention mechanism | mean average precision (mAP) of 81.47%, a recall of 92.81%, and an F1 score of 88.40% |
| Fu H et al. [64] | Mask R-CNN | precision of 0.896, recall of 0.916, F1 score of 0.906, and IoU of 0.822 |
| Wen F et al. [65] | Mask R-CNN | precision of 0.677 |
| Chadwick AJ et al. [66] | Mask R-CNN | (mAP) of 72%, with F1 scores of 69% for lodgepole pine and 78% for white spruce |
| Chen W et al. [67] | Att-Mask R-CNN | 65.29% mean average precision for detection, 80.44% mean intersection over union for segmentation, and a 90.67% |
| Sapkota R et al. [68] | YOLOv8 (one-stage) and Mask R-CNN (two-stage) | YOLOv8 outperformed Mask R-CNN across both datasets, achieving |

| | | |
|---|---|---|
| | | a precision of 0.90 and a recall of 0.95 |

Most of the method consists in using mask R-CNN for higher precision but slow calculation, or YOLO for higher speed and real time detection. The papers that achieved high accuracy are the ones with the individual trees, which makes easy for any model to detect and segment those trees. Some studies consist of calculating the biomass of a group of trees that can't be clearly identified. Depending on the difficulty of the dataset, the model was able to perform very well, and sometimes too well due to the nature of the dataset. But if it's in canopy area, in which the trees are dense. Then the model will find it difficult to detect and segment a tree crown.

### 2.3 Research Gap

Most of the existing research on tree detection and crown segmentation relies heavily on R-CNN-based models to mask out trees or utilize expensive LiDAR data to achieve high accuracy. While these approaches have proven effective, they come with limitations. R-CNN models often require complex layers, longer training times, and higher computational resources, which can limit their practicality for large-scale forest monitoring. Similarly, LiDAR data, though highly precise, is costly to collect and process, making it less accessible for routine applications or in regions with limited resources.

Our method aims to fill this gap by offering an alternative approach that balances accuracy, efficiency, and cost-effectiveness. By using YOLOv11 instance segmentation, we provide a lightweight, fast, and accessible solution for tree crown segmentation, it can analyze the image in 0.63 seconds using only CPU processor. This approach eliminates the need for costly LiDAR data while maintaining competitive accuracy. Additionally, it simplifies the detection process, making it more feasible for real-time monitoring and practical forestry applications. This study contributes to the existing body of knowledge by demonstrating that accurate tree crown and diameter predictions can be achieved using affordable tools and modern deep learning techniques.

## 2.4 Summary

Based on the reviewed literature, we can summarize the research gaps with the following key points:

- Few studies have utilized YOLO for tree segmentation tasks, with most research focusing on other deep learning methods or LIDAR type of dataset.
- Very few papers extend beyond tree segmentation to include calculations of critical tree parameters such as Diameter at Breast Height (DBH) and biomass estimation, limiting the practical applications in forestry and ecosystem management.
- Only one paper briefly mentioned the Canadian Spruce tree, without going deeply into the unique aspects or ecological significance of the species, indicating a lack of focused research on this important tree type.

The allometric study will provide us with a way to calculate and compare the relationship of the spruce features and measures. The literature review confirms that DBH correlates with the TCD value, and that we can use DBH to calculate the dry biomass of the spruce.

# Chapter Three: Methodology

This section outlines the systematic approach employed to predict the tree crown areas using two methods: FRIST METHOD is YOLOv5 + WST compared to SECOND METHOD YOLOv11 instance segmentation. The process involves capturing high-resolution aerial images using drones, annotating these images for tree crown detection, and applying YOLO models to automate the identification and segmentation of tree crows. Then, empirical correction method is used to refine the predictions of the first method, ensuring more accurate estimation of tree crown areas.

Below are two visual representations of the workflow utilized in this study, illustrating each step from entry data (which is an image) to the predicted TCD after YOLO training.

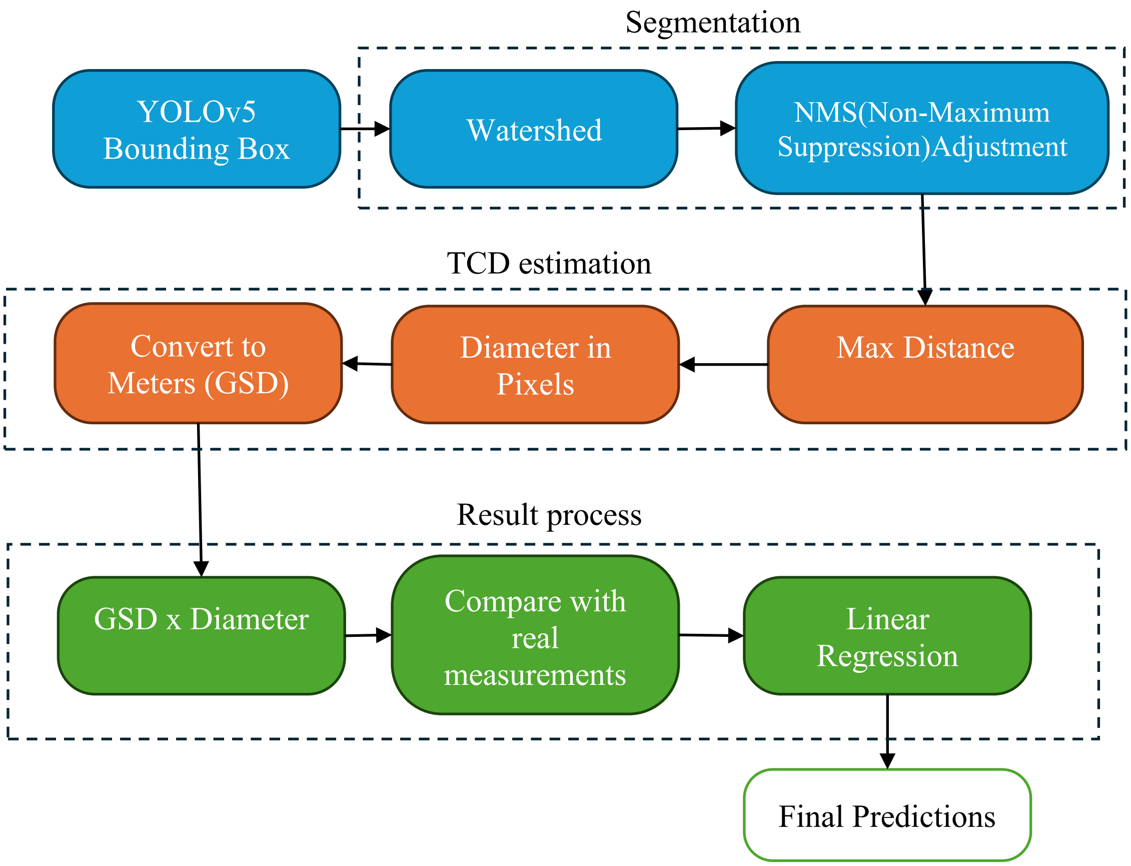


**Figure 2**: First method YOLOv5 + WST

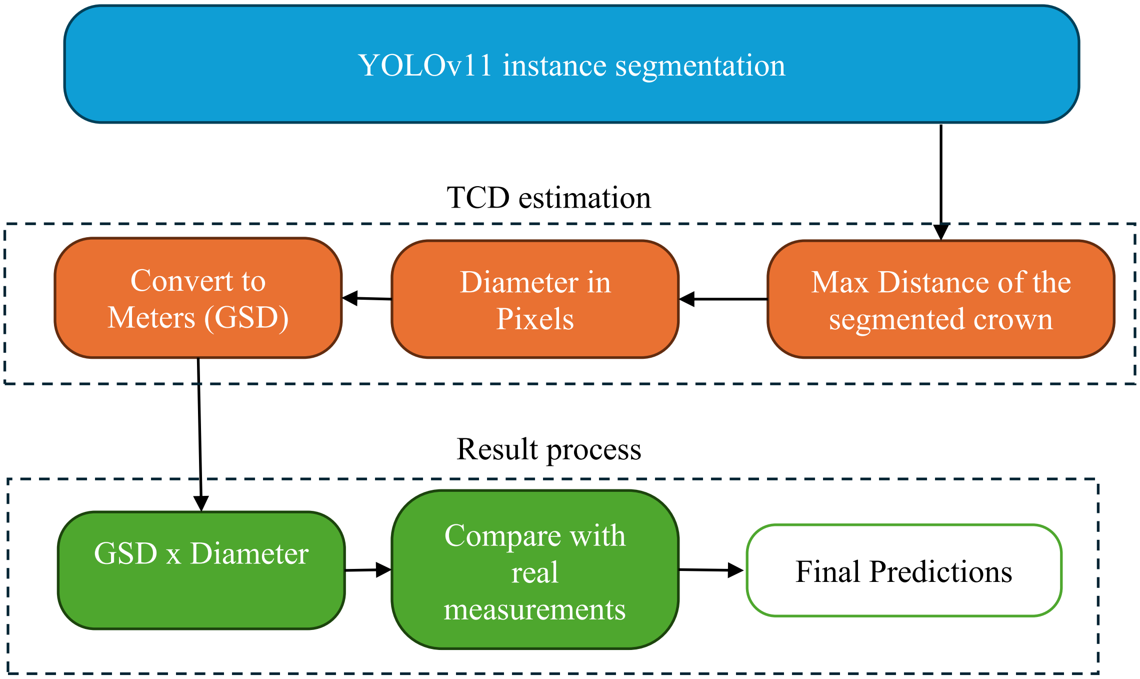


**Figure 3:** Second Method YOLOv11 Instance Segmentation Technique

### 3.1 Materials

Drone: DJI MAVIC 3 CLASSIC with a 4/3 CMOS Hasselblad camera was used for capturing high-resolution images of the study area:

- Specs: 20 MP effective pixels, 24 mm equivalent format, and an 84-degree field of view

Measurement Tools:

- Haglof Vertex V Hypsometer: Used for measuring the crown diameter of the trees with an accuracy of ± 1cm
- Richter 5m Tree Diameter Tape: Used to measure the Diameter at Breast Height (DBH) of trees with an accuracy of ± 2 to 5 mm

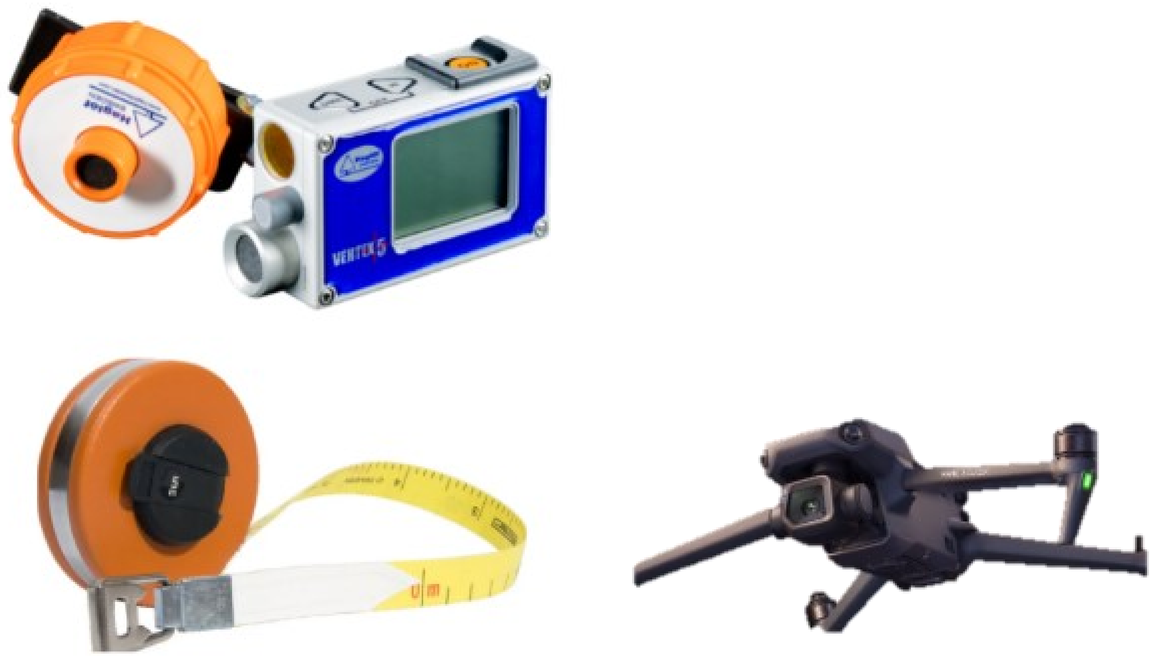


**Figure 4**: Tools Used for the Project

### 3.2 Data Acquisition

Images were taken from two different heights (50 meters and 70 meters) in batches for each area. The images were taken at 11 am on three consecutive sunny days. A total of 72 drone images were captured and 2,155 of two different spruce tree species (White spruce and blue spruce) in the area marked by the map figure. These two types of spruce share the same allometric property, the only difference is in their color. The division of these spruces will increase the precision of our model in terms of identification and not confuse the color of the spruce. We had before conducted the detection training without the separation of two types only for the model to mistakenly miss some Spruces and detect some other species such as Pine trees. We also believe that the color of each spruce tree gives different textures and different edges in the image, which explains why we decided to separate two types. The datasets are obtained from the pure spruce region, mixed region, condensed tree area, spreading tree area, lined up tree area, and half-infected abnormal tree areas.

The goal of covering various regions is to ensure diversity in tree crown shapes and sizes and challenge our model to perform in all types of areas. The reason why the study area is limited is due to the prohibition of the flight area. This area is the only area with a good number of public spruce trees that is not located near the housing area.

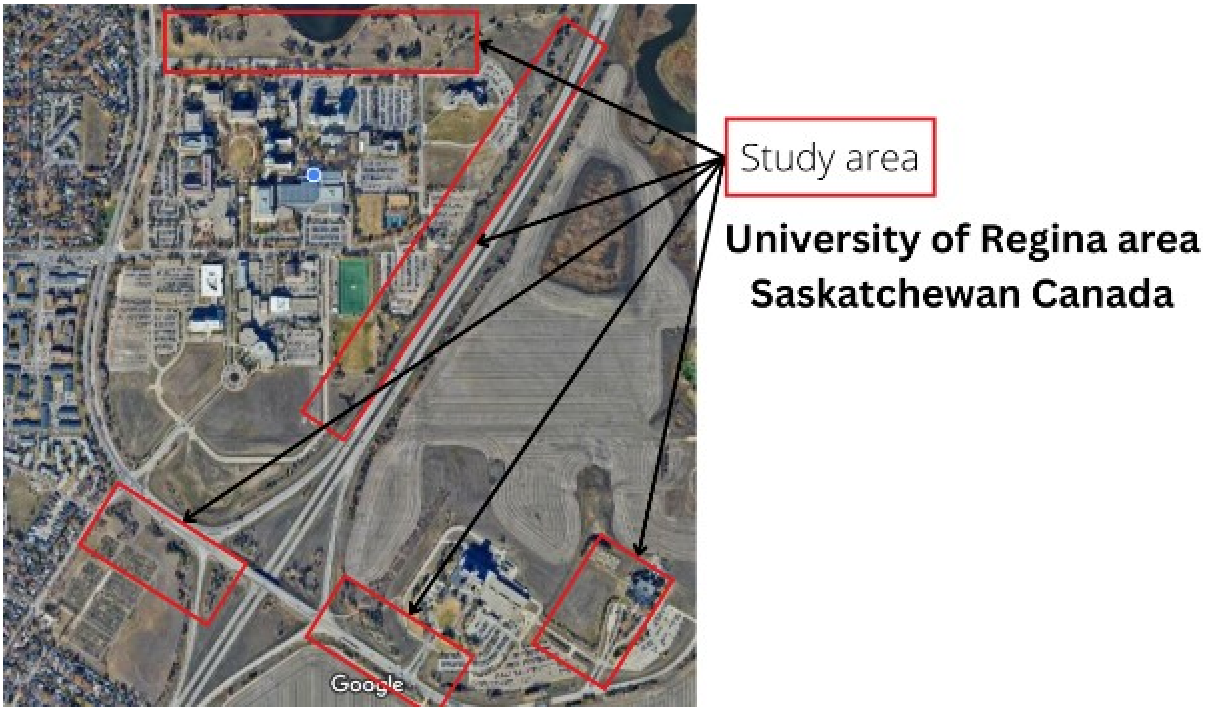
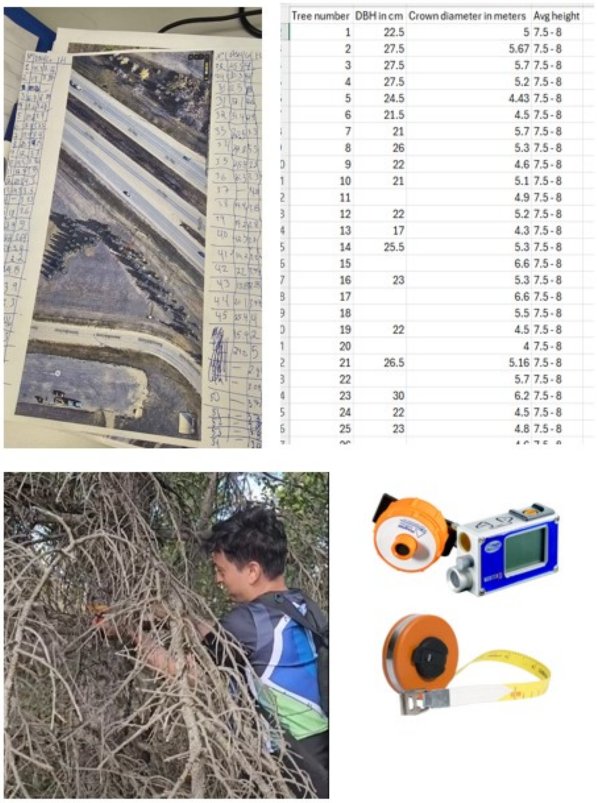


**Figure 5**: Overview of the study area: Geographical location of University of Regina Saskatchewan taken by google map 3737 Wascana Pkwy, Regina, SK S4S 0A2

The same number of trees on-site measurements are conducted with help of Haglof Vertex V Hypsometer tool to measure TCD. In addition, 303 samples of DBH are taken using diameter reeling tape. The measuring took a total of 21 hours carefully labeling each tree by its corresponding number.

These on-site measurements are to ensure the validation of our model with real data; these measurements were taken and recorded in multiple .csv files that will allow us to compare the prediction that the model made within the same tree from that area. The statistical distribution of the crown diameter is shown in figure. We can see from the figure that the distribution is following a normal distribution.



| Tree number | DBH in cm | Crown diameter in meters | Avg height |
|---|---|---|---|
| 1 | 22.5 | 5 | 7.5 - 8 |
| 2 | 27.5 | 5.67 | 7.5 - 8 |
| 3 | 27.5 | 5.7 | 7.5 - 8 |
| 4 | 27.5 | 5.2 | 7.5 - 8 |
| 5 | 24.5 | 4.43 | 7.5 - 8 |
| 6 | 21.5 | 4.5 | 7.5 - 8 |
| 7 | 21 | 5.7 | 7.5 - 8 |
| 8 | 26 | 5.3 | 7.5 - 8 |
| 9 | 22 | 4.6 | 7.5 - 8 |
| 10 | 21 | 5.1 | 7.5 - 8 |
| 11 | | 4.9 | 7.5 - 8 |
| 12 | 22 | 5.2 | 7.5 - 8 |
| 13 | 17 | 4.3 | 7.5 - 8 |
| 14 | 25.5 | 5.3 | 7.5 - 8 |
| 15 | | 6.6 | 7.5 - 8 |
| 16 | 23 | 5.3 | 7.5 - 8 |
| 17 | | 6.6 | 7.5 - 8 |
| 18 | | 5.5 | 7.5 - 8 |
| 19 | 22 | 4.5 | 7.5 - 8 |
| 20 | | 4 | 7.5 - 8 |
| 21 | 26.5 | 5.16 | 7.5 - 8 |
| 22 | | 5.7 | 7.5 - 8 |
| 23 | 30 | 6.2 | 7.5 - 8 |
| 24 | 22 | 4.5 | 7.5 - 8 |
| 25 | 23 | 4.8 | 7.5 - 8 |

**Figure 6**: Onsite measurement for crown diameter and DBH (excel file and on paper)

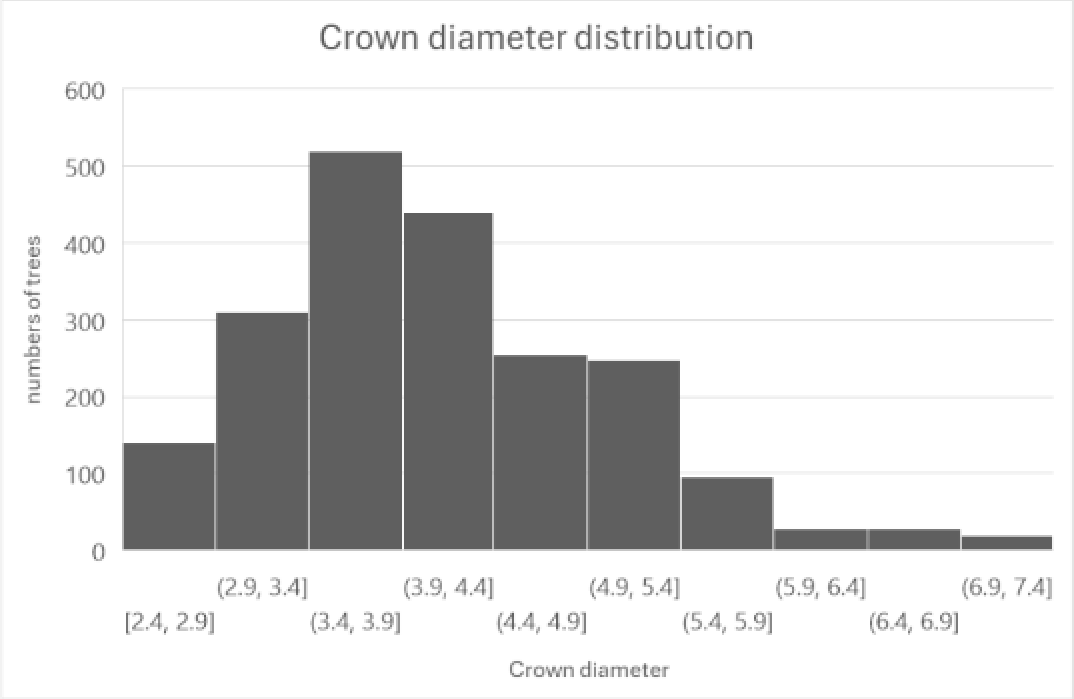


**Figure 7**: An overview on the measured crown diameter distribution (using Haglof Vertex V Hypsometer)

This histogram suggests that most trees in the sample have medium-sized crowns, with a central tendency around 3.4 to 3.9 meters in diameter. There are fewer trees with very small or very large crown diameters, and the distribution slightly skews toward larger diameters. Most trees (more than 80%) have crown diameters within the range of (2.9, 5.4] meters. This general distribution pattern is consistent with natural growth distributions in tree populations, where the majority of trees are of moderate size, with fewer extremes. Therefore, we can ensure that our measurement is credible.

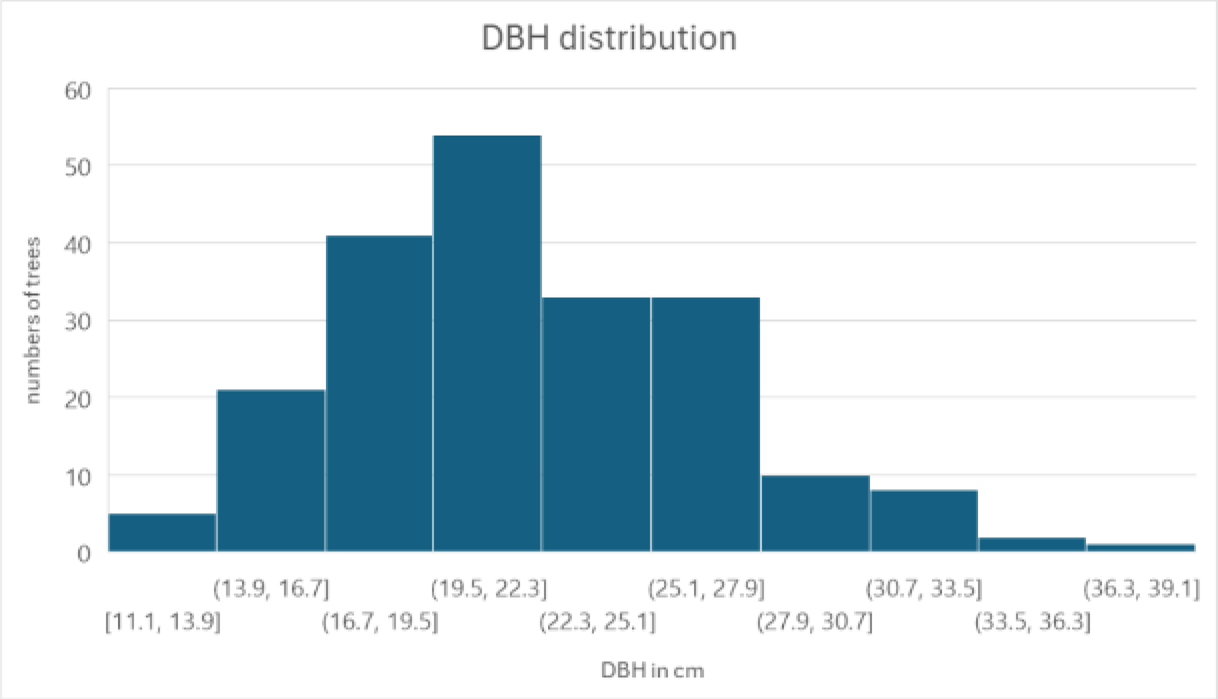


**Figure 8:** An overview on the measured DBH distribution of 303 samples

This DBH histogram of 303 samples, has the same distribution as the Crown Diameter histogram. Note that this does not show that there is a relationship between them. But it shows our model domain, which say that our accuracy is correct in that distribution and interval. If the majority of the trees has smaller or bigger to either extremity of the distribution, our model might behave differently.

### 3.3 Data Preparation

**Labeling with bounding box for YOLOv5:**

In the first method, manual labeling of tree crowns was conducted using the LabelImg tool (example shown in figure 9), a graphical annotation software for drawing bounding boxes around objects of interest in images.

The bounding box manual labeling took 3 hours. Each tree crown was enclosed in a bounding box, and the labels were saved in a YOLO-compatible format as `.txt` files. These files included essential information such as the class of the object (tree crown), the x and y coordinates of the bounding box center, as well as its width and height. All coordinates and dimensions are transformed to 640 pixels height to ensure compatibility with the YOLO model (because YOLO model can accept only this type of image dimensions). A dataset was subsequently created, comprising images and their corresponding labeled bounding boxes. This dataset was split into 80% training and 10% validation sets and 10% testing, where the training data was used to teach the YOLO model how to identify tree crowns. The validation set was reserved to evaluate the model's performance on unseen data, ensuring its ability to generalize. This dataset is well planned to ensure accurate predictions of crown dimensions, which were later used to analyze crown diameter distribution in forested areas.

We manually label 2,155 trees across 72 different images for training: White spruce: 1,248 samples, these trees are very common in Canadian land. Their leaves are dark green even though they are called White type.

The other type is blue spruce: 907 samples, they have a light blue close to white when it is struck by sunlight. They are a special gene that is common in country areas. They do share the same characteristic in terms of allometric measurements. These trees are usually found in gardens and playgrounds. And due to its domestic nature and beauty, gardeners choose them as decoration. We took those trees into consideration because it is sometime grows next to the white common spruce trees.

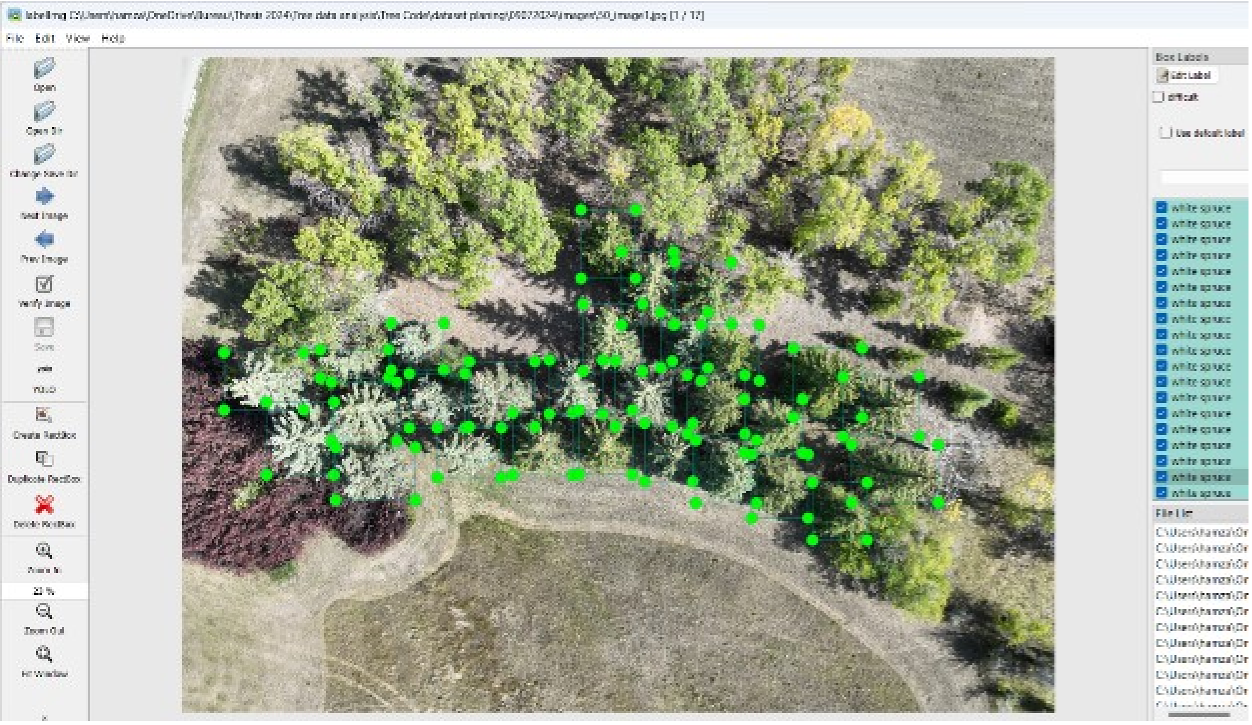


**Figure 9**: Bounding Box Labeling

## Labeling with Segmentation for YOLOv11(second method):

The second method utilized for instance segmentation, where each tree crown is not only detected but also segmented and labelled individually. This process was also carried out using the Labelme tool, but instead of bounding box we are using it for instance segmentation tasks. The segmentation labelling took around 16 hours. Please note that this is a different labelling than the bounding box. In this method, instead of simple bounding boxes, precise polygonal outlines were drawn around each tree crown, allowing for a more accurate representation of the crown's shape and boundaries. The file segmentation coordinates then saved in JSON Source File. After labeling the tree crowns, the segmented dataset was further processed using the Roboflow platform. Roboflow provides a suite of tools for data management, including data augmentation to create 121 images for the training set indicated in figure 11. This data augmentation happened in both datasets ( the bounding box dataset and the segmentation dataset), which helps increase the variability of the dataset by applying transformations such as rotation, flipping, and scaling. This augmentation process ensures that the model trained on this dataset becomes more robust and can generalize better to unseen data. Finally, the augmented dataset was automatically split into training, testing, and validation sets within Roboflow. This split enables the model to learn from the training set, while the validation set helps tune the model during training, and the test set is used to assess the model's performance on completely unseen data. This data can be extracted in google Colab with the combability with YOLOv11 model.

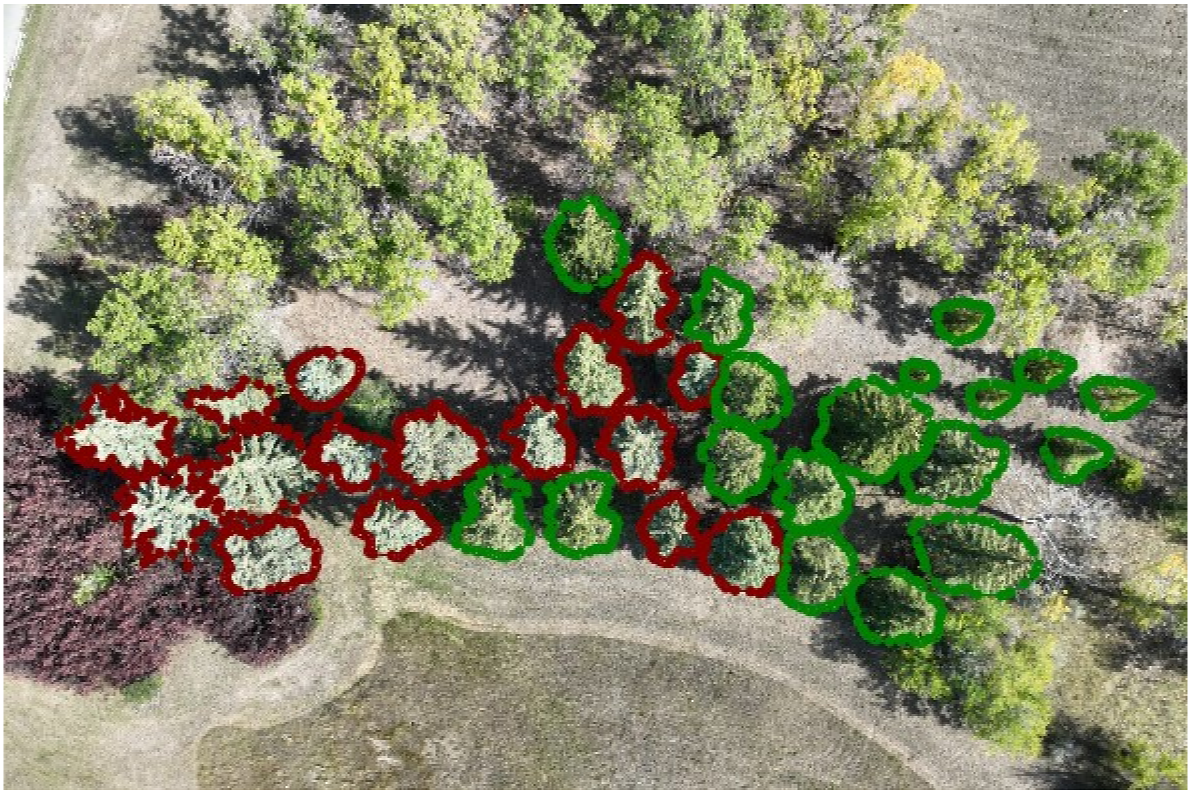
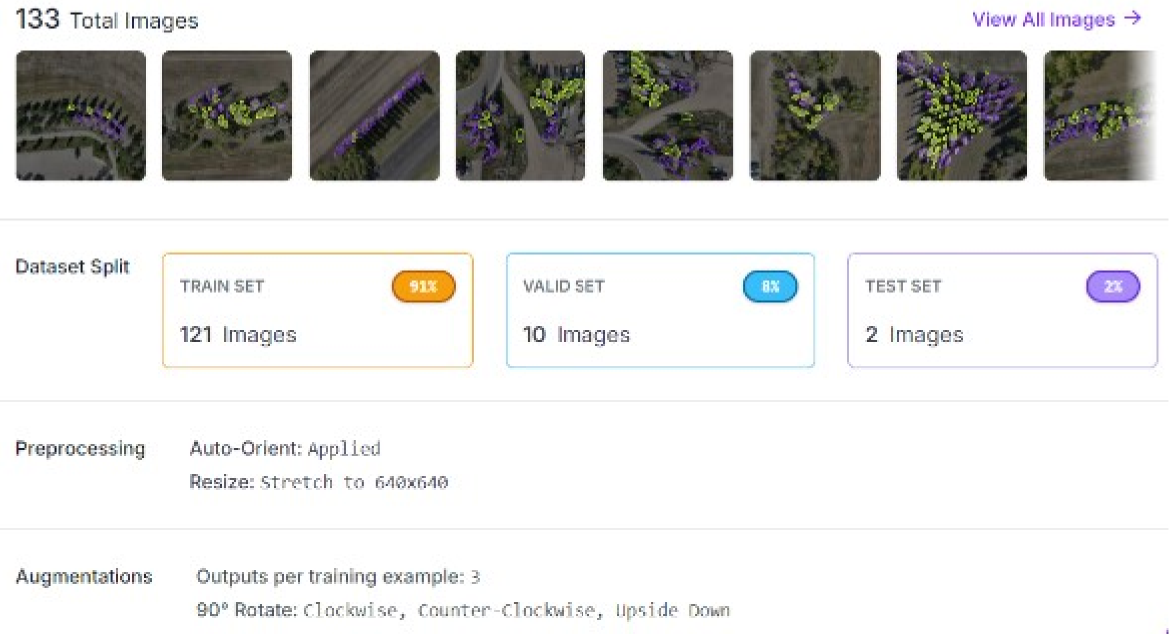


**Figure 10:** Segmentation using Labelme app (red is blue spruce, green is white spruce)

Figure 10 shows the intensity of the labeling task, which is took 16 hours to label all 2,155 trees of the spruce and other type won't be labeled and will be treated as a background. The labeling task is done by clicking on the contour of each tree. Each tree consists of an average of 30 connected dots to make a detailed contour. We also respect the classification of the white spruce and the blue spruce, and not treat those two types as one.



133 Total Images

Dataset Split

TRAIN SET 91%
121 Images

VALID SET 8%
10 Images

TEST SET 2%
2 Images

Preprocessing
Auto-Orient: Applied
Resize: Stretch to 640x640

Augmentations
Outputs per training example: 3
90° Rotate: Clockwise, Counter-Clockwise, Upside Down

**Figure 11**: Data processing in Roboflow website after segmenting in labelme tool

### 3.4 YOLOv5 Model Training (FOR FRIST METHOD)

The objective of training a YOLOv5 model is to automate the detection of tree crowns, thereby reducing the need for manual labeling. In the study area, there are two types of spruce trees: white spruce and blue spruce. Although differentiating between these two species is not a primary focus of the research, the dataset was created with separate labels for white and blue spruce. This approach allows the model to better distinguish spruce trees from other species that are not of interest, enhancing its ability to identify specific characteristics of spruce trees and minimizing false detection of other tree species. In the following table the key elements of the YOLOv5 architecture are shown, where 'Layer' is the index of each layer, 'Params' indicates the number of trainable params, 'Module' specifies the type of operation (e.g., Conv for convolution), and 'Arguments' define the configuration for each module, such as input/output channels and kernel size; for example, in the layer 0, 3 is number of input channels, 32 is number of output channels, 6 is kernel size, the first 2 is stride, and another 2 is padding. The rest Arguments are not padding value.

**Table 3**: YOLOv5 Model Architecture

| Layer | Params | Module | Arguments [input channels, output channels, kernel size, stride, padding] |
|---|---|---|---|
| 0 | 3,520 | Conv | [3, 32, 6, 2, 2] |
| 1 | 18,560 | Conv | [32, 64, 3, 2] |
| 3 | 73,984 | Conv | [64, 128, 3, 2] |
| 5 | 295,424 | Conv | [128, 256, 3, 2] |
| 7 | 1,180,672 | Conv | [256, 512, 3, 2] |
| 9 | 656,896 | SPPF | [512, 512, 5] |
| ... | ... | ... | ... |
| 18 | 147,712 | Conv | [128, 128, 3, 2] |
| 20 | 296,448 | C3 | [256, 256, 1, False] |
| 23 | 1,182,720 | C3 | [512, 512, 1, False] |
| 24 | 18,879 | Detect | [2, [[10, 13, 16, 30, 33, 23], ... [128, 256, 512]] |

The process of training and evaluation is the following:

- Manually label 2,155 trees across 72 different images for training
  - White spruce: 1,248 samples

– Blue spruce: 907 samples

• Train for 50 and 100 epochs to evaluate performance

Use different evaluation metrics such as:

$$\mathrm{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \qquad \text{(Equation 3-1)}$$

$$\text{Precision} = \frac{\text{Correct Detections}}{\text{Detected Trees}} \qquad \text{(Equation 3-2)}$$

$$\text{Recall} = \frac{\text{Correct Detections}}{\text{Total Trees}} \qquad \text{(Equation 3-3)}$$

$$\text{F1} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \qquad \text{(Equation 3-4)}$$

where:

**IoU (Intersection over Union)**: This metric is used to assess how well the predicted bounding box overlaps with the ground truth bounding box. If the result is more than 0.5 it is counted as a correct detection.

- **Area of Intersection**: The area where the predicted and actual bounding boxes overlap.
- **Area of Union**: The combined area covered by both the predicted and actual bounding boxes.

**Precision**: it measures how accurate the model is at detecting objects.

- **Correct Detections**: The number of correctly predicted trees (based on IoU threshold).
- **Detected Trees**: The total number of trees detected by the model.

**Recall:** measures how well the model captures all the actual objects (in this case, trees).

- **Correct Detections**: The number of correct tree detections.
- **Total Trees**: The total number of actual trees present in the image.

**F1 Score** provides a balance between precision and recall, helping to evaluate the overall effectiveness of the model.

### 3.5 Watershed Segmentation and Contour Adjustment with Non Maximum Suppression (NMS)

WST is used to refine tree crown boundaries from each tree identified by the bounding box of the YOLOv5 model.

Watershed Segmentation works on separating overlapping objects in an image. The inspiration for this approach comes from geography. Therefore, the name "watershed lines" refers to those that define drainage basins within a landscape. This technique is particularly applicable when segmenting touching or overlapping objects, in this case, trees.

The Watershed algorithm works by viewing the grayscale image as a 3D topographic surface as shown in figure 12:

- Bright areas are interpreted as high points (e.g., ridges).
- Dark areas are interpreted as low points (e.g., valleys).

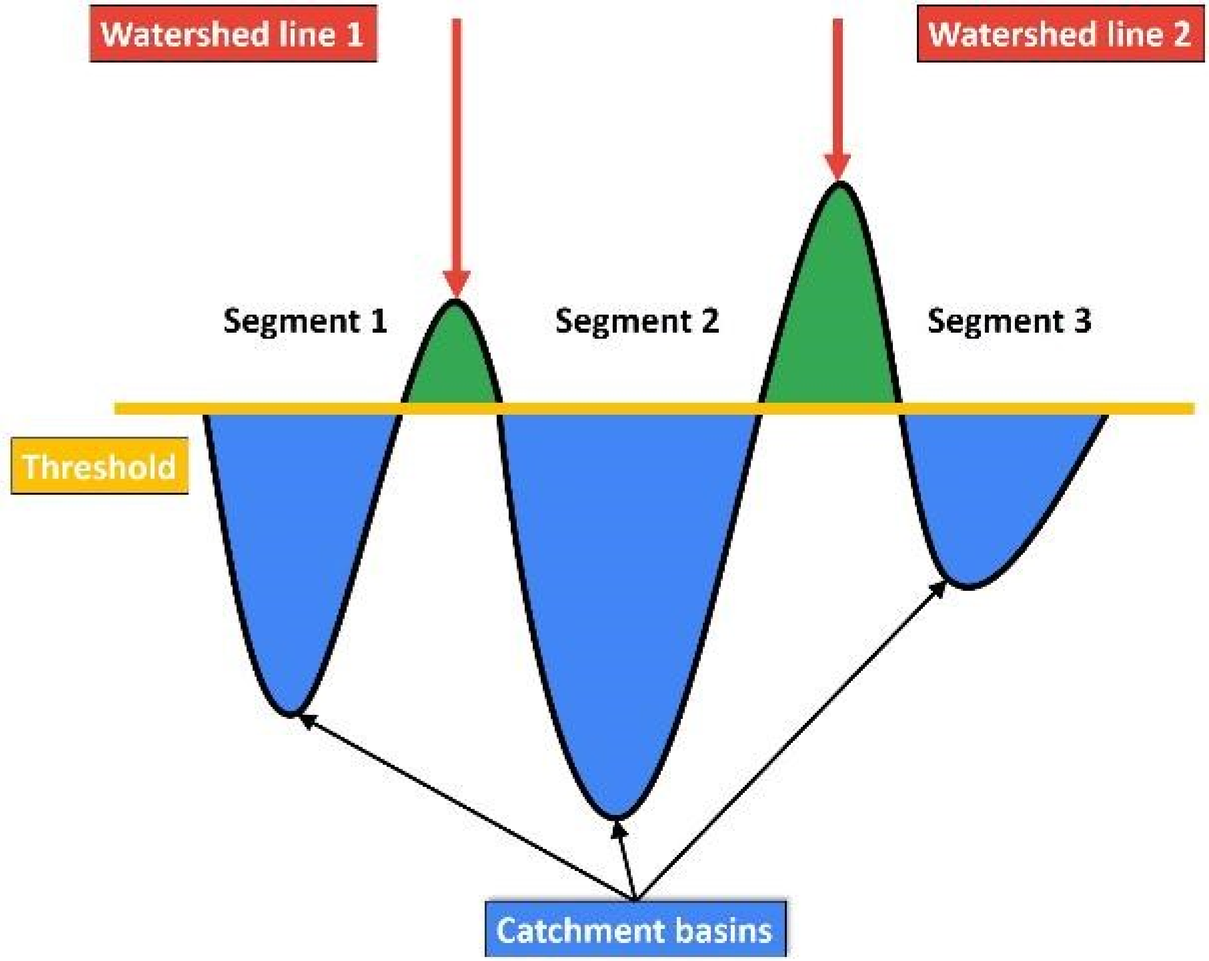


**Figure 12**: Watershed technique side view topology from [71]

The process of WST generally includes the following steps.

Preprocessing:

The preprocessing methods are: blurring (to reduce noise), thresholding (to create a binary image).

Marker-based Segmentation:

Markers are pixels that are pre-defined as belonging to specific objects or the background.

There are two types of markers:

- Foreground markers which are Parts that are certainly part of the object(s).
- Background markers, Parts that are certainly part of the background.

Flooding Process:

The flooding is carried out in such a way that the regions grow uniformly, and boundaries are formed where two different regions meet.

Constructing Watershed Lines:

The points where different regions meet during the flooding process are marked as watershed lines, effectively forming the boundaries between objects.

While (NMS) Non-maximum suppression ensures no overlapping contours by supressing redundant detection, also helps to adjust the contours for better accuracy.

### 3.6 Crown Diameter Estimation for pixel values

To estimate the area of the tree crown (with watershed or instance segmentation), a critical step is calculating the largest distance between two points on the convex hull(it's the extremities of the segmentation which forms a mountain like shape in the pixel values) of the detected contour in an image. This distance is initially measured in pixels. However, in order to convert this measurement into real-world units (meters), a **Ground Sampling Distance (GSD)** calculation is required. The GSD represents the distance between the centers of two consecutive pixels on the ground, as captured by an aerial imaging system, and is fundamental for converting pixel measurements into metric units.

The formula for GSD is given by:

$$\mathrm{GSD} = \frac{\text{Sensor Width} \times \text{Flight Altitude}}{\text{Focal Length} \times \text{Image Width}} \qquad \text{(Equation 3-5)}$$

In this formula:

- **Sensor Width** represents the width of the camera sensor.
- **Flight Altitude** refers to the altitude at which the aerial images were captured.
- **Focal Length** is the distance between the camera lens and the image sensor.
- **Image Width** represents the total number of pixels in the width of the image.

With these parameters, it becomes possible to calculate the GSD and convert the detected crown diameters from pixels to meters. In this project, we will be using the following values:

- **Sensor Width** = 13.2 mm
- **Flight Altitude** = 50 or 70 meters
- **Focal Length** = 12 mm
- **Image Width** = 5280 pixels (for first method), 640 pixels (for second method)

The GSD can be computed, providing a scale factor to translate pixel-based measurements into real-world dimensions. This GSD value is crucial when estimating tree crown dimensions or other spatial features in the aerial imagery.

To convert the diameter of the tree crown from pixels to meters, one simply multiplies the measured crown diameter (in pixels) by the calculated GSD value. This conversion enables accurate estimation of tree crown sizes in meters, given that the camera is calibrated.

Finally, to compare the original and the predictions, the following evaluation metrics will be used:

Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{Equation 3-6}$$

Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2 \tag{Equation 3-7}$$

Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{Equation 3-8}$$

Mean Percentage Error (MPE):

$$\mathrm{MPE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right) \times 100 \tag{Equation 3-9}$$

where:

- $y_i$ represents the actual value or the true observation at the i-th instance in your dataset.
- $\widehat{y_i}$ represents the predicted value from your model for the i-th instance.
- n is the total number of observations in your dataset.

### 3.7 YOLOv11 Model Training

The reason of the usage of v11 instance segmentation is because v5 doesn't have that option. The second reason is that v5 is just a showcase where the performance in detection doesn't change. The third reason is that if we use v11 in detection only, then watershed technique becomes obsolete.

The difference between WST and instance segmentation is that WST use algorithms and pixel hills to estimate the contour of the tree crown. But Instance segmentation uses Deep Learning network to estimate the same contour but using features instead of pixel values denting.

In this method, the primary objective of training a YOLOv11 instance segmentation model is to detect and segment Spruce tree crowns, mimicking the combined function of bounding box and WST. The YOLOv11 model is a cutting-edge deep learning architecture designed for both object detection and instance segmentation, making it highly suitable for accurately identifying and segmenting tree crowns in aerial images. By leveraging this model, the segmentation process can be automated, leading to faster and more scalable tree crown analysis.

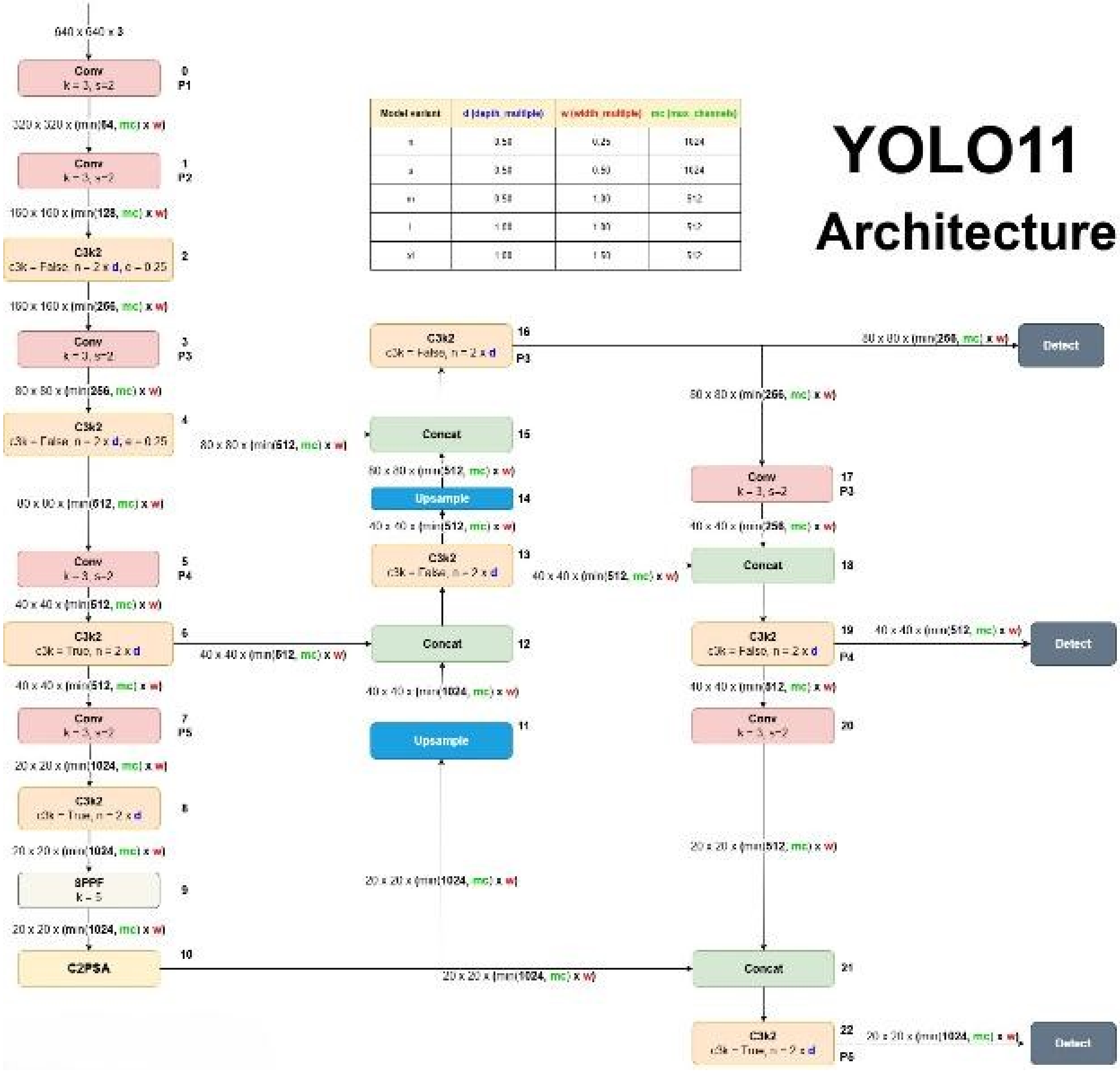
The dataset for training the YOLOv11 model was annotated using the Roboflow platform. Roboflow provides a comprehensive set of tools for dataset management, including labeling and data augmentation. Through this process, 121 labeled images were generated for training, 2 for testing, and 10 for validation. The data augmentation techniques applied, such as rotation, flipping, and scaling, increased the variability within the dataset, enhancing the model's ability to generalize to unseen data. This augmentation is crucial for improving the robustness of the YOLOv11 model, allowing it to perform well on new images with varying lighting conditions, angles, and tree structures.

**Table 4**: YOLOv11 Model Summary

| Layer (backbone, neck, head) explained in figure 13) | Params | Module | Arguments [input channels, output channels, stride, kernel size] |
|---|---|---|---|
| 0 | 928 | ultralytics.nn.modules.conv.Conv | [3, 32, 3, 2] |
| 1 | 18560 | ultralytics.nn.modules.conv.Conv | [32, 64, 3, 2] |
| 2 | 26080 | ultralytics.nn.modules.block.C3k2 | [64, 128, 1, False, 0.25] |
| 3 | 147712 | ultralytics.nn.modules.conv.Conv | [128, 128, 3, 2] |
| 4 | 103360 | ultralytics.nn.modules.block.C3k2 | [128, 256, 1, False, 0.25] |
| 5 | 590336 | ultralytics.nn.modules.conv.Conv | [256, 256, 3, 2] |
| 6 | 346112 | ultralytics.nn.modules.block.C3k2 | [256, 256, 1, True] |
| 7 | 1180672 | ultralytics.nn.modules.conv.Conv | [256, 512, 3, 2] |
| 8 | 1380352 | ultralytics.nn.modules.block.C3k2 | [512, 512, 1, True] |
| 9 | 656896 | ultralytics.nn.modules.block.SPPF | [512, 512, 5] |
| 10 | 990976 | ultralytics.nn.modules.block.C2PSA | [512, 512, 1] |
| 11 | 0 | torch.nn.modules.upsampling.Upsample | [None, 2, 'nearest'] |
| 12 | 0 | ultralytics.nn.modules.conv.Concat | [1] |

| | | | |
|---|---|---|---|
| 13 | 443776 | ultralytics.nn.modules.block.C3k2 | [768, 256, 1, False] |
| 14 | 0 | torch.nn.modules.upsampling.Upsample | [None, 2, 'nearest'] |
| 15 | 0 | ultralytics.nn.modules.conv.Concat | [1] |
| 16 | 127680 | ultralytics.nn.modules.block.C3k2 | [512, 128, 1, False] |
| 17 | 147712 | ultralytics.nn.modules.conv.Conv | [128, 128, 3, 2] |
| 18 | 0 | ultralytics.nn.modules.conv.Concat | [1] |
| 19 | 345472 | ultralytics.nn.modules.block.C3k2 | [384, 256, 1, False] |
| 20 | 590336 | ultralytics.nn.modules.conv.Conv | [256, 256, 3, 2] |
| 21 | 0 | ultralytics.nn.modules.conv.Concat | [1] |
| 22 | 1511424 | ultralytics.nn.modules.block.C3k2 | [768, 512, 1, True] |
| 23 | 1474678 | ultralytics.nn.modules.head.Segment | [2, 32, 128, [128, 256, 512]] |

YOLO11s-seg summary: 355 layers, 10,083,062 parameters, 10,083,046 gradients, 35.6 GFLOPs (Floating point operations per second)



| Model variant | d (depth_multiple) | w (width_multiple) | mc (max_channels) |
|---|---|---|---|
| n | 0.50 | 0.25 | 1024 |
| s | 0.50 | 0.50 | 1024 |
| m | 0.50 | 1.00 | 512 |
| l | 1.00 | 1.00 | 512 |
| x | 1.00 | 1.50 | 512 |

YOLO11
Architecture
640 x 640 x 3
Conv k = 3, s=2
0 P1
320 x 320 x (min(64, mc) x w)
Conv k = 3, s=2
1 P2
160 x 160 x (min(128, mc) x w)
C3k2 c3k = False, n = 2 x d, e = 0.25
2
160 x 160 x (min(256, mc) x w)
Conv k = 3, s=2
3 P3
80 x 80 x (min(256, mc) x w)
C3k2 c3k = False, n = 2 x d, e = 0.25
4
80 x 80 x (min(512, mc) x w)
Conv k = 3, s=2
5 P4
40 x 40 x (min(512, mc) x w)
C3k2 c3k = True, n = 2 x d
6
40 x 40 x (min(512, mc) x w)
Conv k = 3, s=2
7 P5
20 x 20 x (min(1024, mc) x w)
C3k2 c3k = True, n = 2 x d
8
20 x 20 x (min(1024, mc) x w)
SPPF k = 5
9
20 x 20 x (min(1024, mc) x w)
C2PSA
10
80 x 80 x (min(512, mc) x w)
40 x 40 x (min(512, mc) x w)
Upsample
11
40 x 40 x (min(1024, mc) x w)
Concat
12
C3k2 c3k = False, n = 2 x d
13
40 x 40 x (min(512, mc) x w)
Upsample
14
80 x 80 x (min(512, mc) x w)
Concat
15
C3k2 c3k = False, n = 2 x d
16 P3
80 x 80 x (min(256, mc) x w)
Detect
80 x 80 x (min(256, mc) x w)
Conv k = 3, s=2
17 P3
40 x 40 x (min(256, mc) x w)
40 x 40 x (min(512, mc) x w)
Concat
18
C3k2 c3k = False, n = 2 x d
19 P4
40 x 40 x (min(512, mc) x w)
Detect
40 x 40 x (min(512, mc) x w)
Conv k = 3, s=2
20
20 x 20 x (min(512, mc) x w)
20 x 20 x (min(1024, mc) x w)
20 x 20 x (min(1024, mc) x w)
Concat
21
C3k2 c3k = True, n = 2 x d
22 P5
20 x 20 x (min(1024, mc) x w)
Detect

**Figure 13**: YOLOv11 architecture according to early analysis from Dr.Priyanto Hidayatullah [69] (low resolution fix)

Figure 11 shows the detailed structure of the new YOLOv11, this is not taken from the YOLO author himself, but it was an early analysis of the structure. This is due to Ultralytics company not releasing the paper of this model. However, in there model code, we can understand how this model works.

The model is divided into three parts: Backbone, Neck, and Head.

**The Backbone:** is the feature extraction component which downsample the initial inputs into flat one. It has certain layers which I will explain here:

Convolutional Layers (Conv):

The input image of size (640 x 640 x 3) passes through a sequence of convolutional layers, gradually reducing the spatial dimensions while increasing the depth.

Each Conv layer is followed by a stride (s=2), which reduces the spatial resolution, effectively downsampling the image to extract higher-level features while losing spatial resolution.

C3k2 Blocks:

Are repeats block that are used to define n parameter to contact with the bottom neck when the layers are upscaling. It is a new block which is introduced to YOLOv11. It is also controlled by the expansion value that will influence the concatenation process.

**Neck**: is responsible for feature fusion, helping to concat the feature maps from convolution layer of the backbone into the neck.

SPPF (Spatial Pyramid Pooling Fast):

The SPPF layer combines information from multiple scales using different kernel sizes, capturing global information from the feature maps.

C2PSA Block:

The C2PSA block follows the SPPF block. It is used to define the end parameters. This repeat block is also a new block introduced in this new model.

Upsample Layers:

The Upsample operations are used to increase the spatial resolution of feature maps.

Concat Layers:

The Concat operations are used to combine feature maps coming from different levels of the backbone and neck.

**The Head:** is responsible for making final predictions for object detection, including predicting

**Summery Flow of Information in the Network:**

Input to Backbone:

The input image goes through a series of convolutions and C3k2 blocks that progressively extract features at multiple scales (P1 to P5).

Feature Extraction in the Backbone:

The backbone ends with SPPF and C2PSA, which aggregate multi-scale information and apply an attention mechanism.

Feature Fusion in the Neck:

In the Neck, features from different levels (P3, P4, P5) are combined using concatenation and upsampling, to ensure that both spatial details and abstract features are included.

Prediction in the Head:

The Head makes predictions at 3 scales using the Detect layers, which ensures that objects of different sizes are properly detected.

One potential challenge in this approach lies in identifying a suitable evaluation metric to compare the performance of the YOLOv11 instance segmentation model against the previous technique, which relied on the Watershed method. The evaluation metric is essential for determining how well the new model performs in terms of both detection accuracy and segmentation quality.

There will be two stages of evaluation in order to compare this technique with the first one:

- Spruce tree detection: Inersection over Union (IoU), Precision, Recall

- Crown diameter estimation (comparing the predicted TCD to the real measured TCD): MAE, MSE, RMSE, and MPE

### 3.8 Mask RCNN

This method is old and have been a lot of use for tree segmentation. It is 7 years old code that uses dependencies that aren't compatible with the updates that are available today. MaskRCNN uses TensorFlow as the library. The most famous one which can be found in github account called matterport [70]. However, many coders faced the same dependency problem and try to update it from TensorFlow 1.0 to 2.0. However, these attempts also faced the same problems afterwards. This technique didn't have many updates due to the lack of interest of the researchers in recent 3 years. The technique suffers from slow training and slow detection. This is because the numbers of parameters were very high and wasn't optimised.

However, we are going to test if this method can stand up in terms of detecting spruce trees. We expect that this method will show poor results in terms of tree detection and speed. If the model could not perform well in detecting, then we can disregard it in terms of segmentation. The results will show in the next chapter.

### 3.9 Summery

In this chapter we have explained two workflows of each method: YOLOv5 + WST and YOLOv11 instance segmentation. Then we proceed on the materials description to gather dataset and combine it with our measuring. The images are labeled with segmentations for instance segmentation data and bounding box for YOLOv5 tree detection data, then processed into data augmentation. Lastly we describe each method and the YOLOs layers structure of each version.

# Chapter Four: Results and Findings

In this chapter, we will present our results on the detection of spruce trees, followed by the estimation of Tree Crown Diameter (TCD). The performance of the proposed methods will be evaluated and compared to determine their effectiveness in accurately identifying tree crowns and estimating their size. Specifically, we will compare the results obtained from the YOLOv5 + Watershed Transform (WST) approach against the outcomes from the YOLOv11 instance segmentation model. Note that the difference in these two methods can be summarize by that the watershed is just an algorithm and instance segmentation is a deep learning that require training. This comparison will highlight the strengths and weaknesses of each method in terms of detection accuracy, segmentation precision, and computational efficiency.

In addition to evaluating the detection and segmentation methods, we will also explore the most effective Machine Learning (ML) models for predicting the DBH. The accurate prediction of DBH is important, as it plays a central role in calculating biomass estimation. This is because dry biomass is calculated using specific allometric equations that rely on the DBH value as a key input variable.

Ultimately, the results from this chapter will help identify which combination of methods and models yields the highest accuracy in both tree crown detection and DBH estimation.

## 4.1 Spruce tree detection model Results

The first objective of this project is to determine the spruce tree (both white and blue) which will be segmented for further analysis. In this section, we will present a detailed comparison of each model used for tree crown segmentation, examining their performance and analyzing the results obtained from training and validation processes.

A key part of our evaluation involves comparing the training duration of each model, assessing how well they perform at different epoch counts, and examining their Loss_Box values for both training and validation datasets. The Loss_Box value(which is compared by the validation dataset, it calculates the offset of the predicted box and the true box) measures how well the predicted bounding boxes align with the ground truth during training. For a model to be effective, the Loss_Box values in the training and validation sets should closely align and follow similar trends. A significant divergence between the two sets would indicate a potential issue with overfitting, where the model becomes overly specialized in the training data and fails to generalize well with new, unseen data.

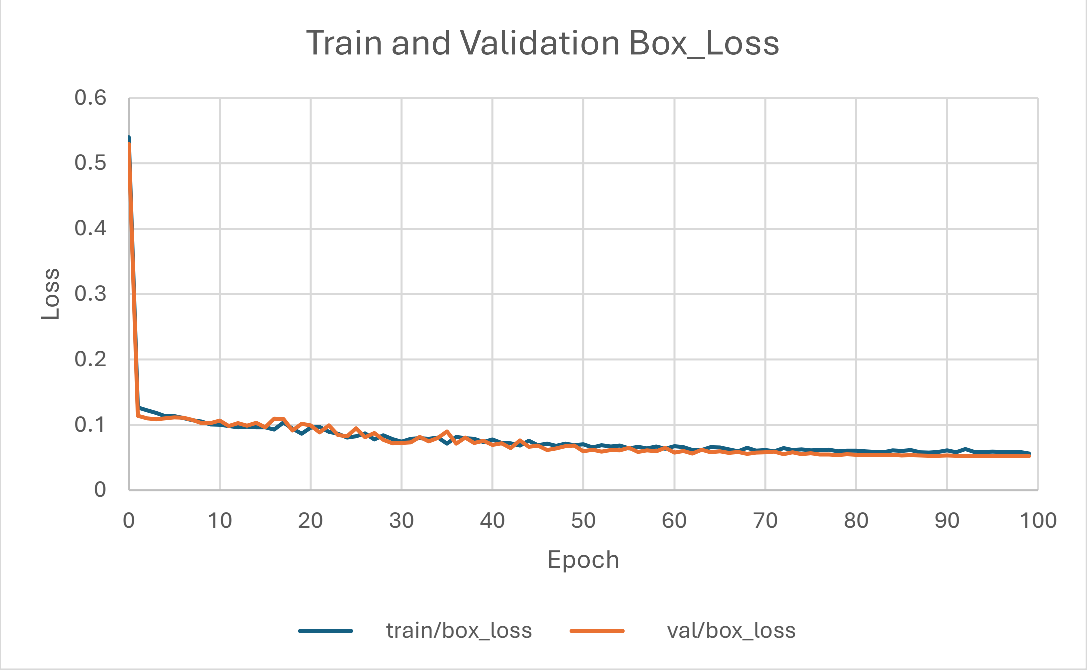


**Figure 14:** Train and Validation Box_Loss for YOLOv5 bounding box

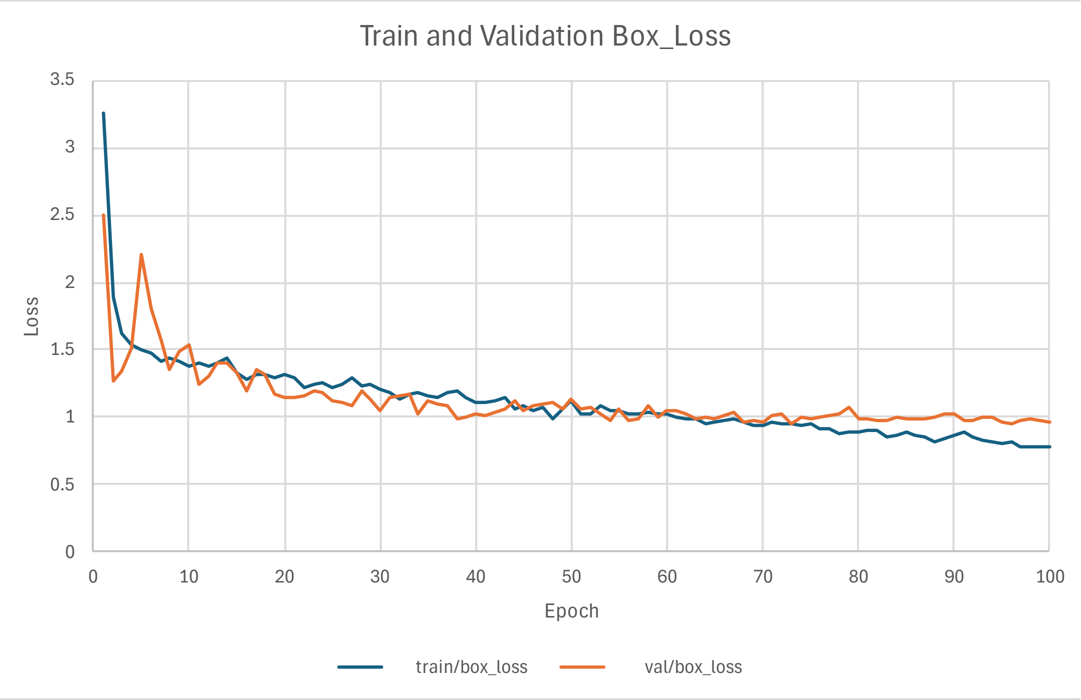


**Figure 15**: Train and Validation Box_Loss for YOLOv11 instance segmentation

Figure 14 shows that YOLOv5 has better Box_Loss than YOLOv11 instance segmentation. This is because the pretrained initial model of YOLOv5 was more recognizing in aerial detection and does not take account on segmenting a tree. While YOLOv11 instance segmentation was pretrained on trees that are photographed from ground. We can also notice a slight or the beginning of overtraining by YOLOv11 starting from 75 epoch. Which means it was a good call for it to end the training on 100 epoch. This can show us later on a credible comparison between both methods.

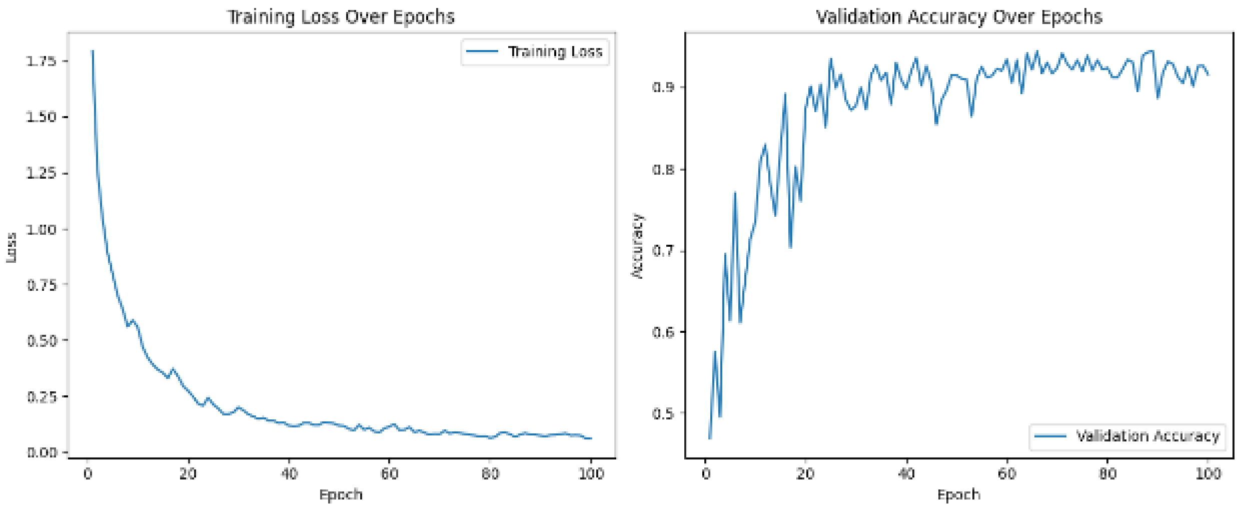


**Figure 16**: Training of MaskRCNN

### MaskRCNN Results

We have also wanted to compare the MaskRCNN model to our two methods of YOLO and check if it is worth to consider the use of MaskRCNN.

The training took about 6 hours using the same amount of GPU and batches and other parameters as YOLO. While YOLO model interestingly takes less than 5 minutes to finish 100 epoch.

This example of tree detection shows the poor performance of this model in a simple term of bounding box and detection in Figure 16.

The figure shows the repeated boxes that overlap each other and misses. And that image explain why it couldn’t reach the same accuracy that the YOLOs can do. It notes that the RCNN can not preform segmentation on these tree crowns, therefore, we can’t count the accuracy of segmentation.

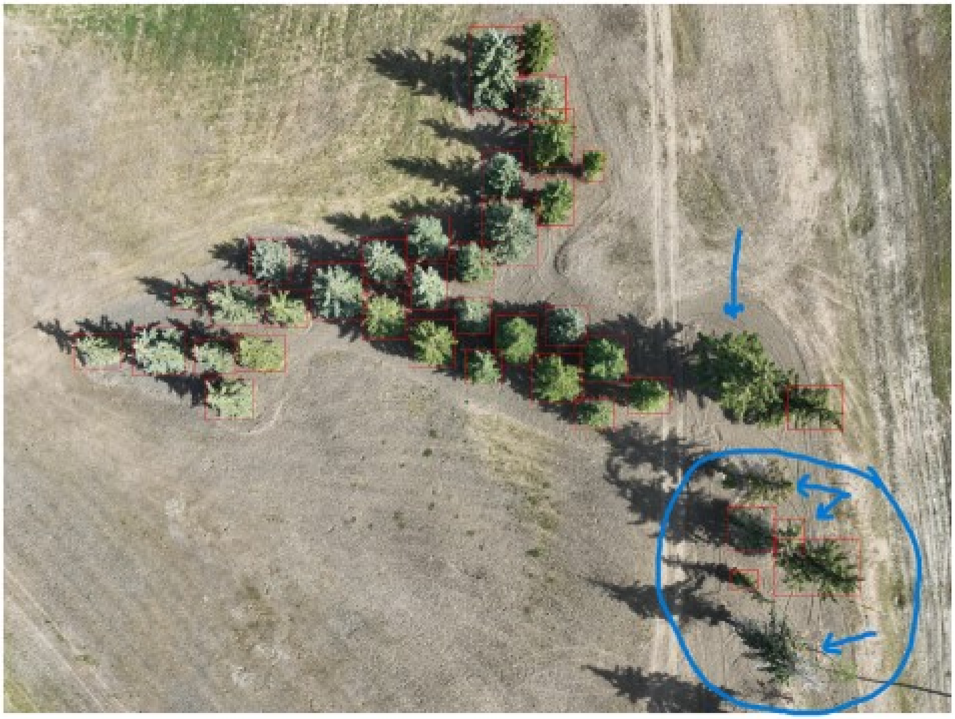


**Figure 17**: Detection by MaskRCNN

Next table we show the comparison of identification among these three methods:

**Table 5:** Identification Comparison

| Metric | YOLOv5 | YOLOv11 | MaskRCNN |
|---|---|---|---|
| Precision (P) | 0.95 | 0.95 | 0.91 |
| Recall (R) | 0.96 | 0.94 | 0.9 |
| mAP@50 (Mean Average Precision at IoU 50%) | 0.94 | 0.98 | 0.89 |
| Mask Precision | - | 0.95 | - |
| Mask Recall | - | 0.94 | - |
| Mask mAP@50 | - | 0.98 | - |

In the table 5, we can see that both YOLOv5 and YOLOv11 achieved 0.95 precision, meaning that 95% of all detected bounding boxes by both models are correctly identified as spruce tree boxes (does not take account of segmentation yet). This indicates both models are equally effective at avoiding false positives (wrongly detected objects). YOLOv5 has a slightly higher recall (0.96) compared to YOLOv11's recall (0.94). This suggests that YOLOv5 detected a marginally higher percentage of the actual trees, meaning it is slightly better at avoiding missed detections (false negatives). YOLOv11 achieves a higher mAP@50 (0.98) compared to YOLOv5's mAP@50 (0.94). This indicates that YOLOv11 provides more consistent and accurate bounding box predictions, particularly in scenarios where precise overlap with the ground truth is required. This suggests that YOLOv11 performs better in terms of overall detection quality, even though its recall is slightly lower than YOLOv5.

Mask R-CNN shows lower performance in object detection compared to YOLOv5 and YOLOv11, with a precision of 0.91, recall of 0.90, and mAP@50 of 0.89. This indicates that it has a higher rate of false positives and misses more objects.

### 4.2 Tree Crown Diameter (TCD) Estimation

After spruce tree detection, a segmentation is performed for the two models. And then, each segmented tree is processed to calculate the diameter of the masked part. We are going to show just one example of many comparisons in this section. Other comparisons and results are shown in the appendices.

Note that each tree is numerated by a number so we can compare it by the corresponding real measured dataset. In addition, we are not comparing YOLO, we are comparing the segmentation part of each method: watershed against instance segmentation. The results shown doesn’t contain boxes, it contains segmentation.

The WST with the help of Non-Maximum Suppression NMS segment the tree crown by separating the tree from the background.  The segmentation process successfully detects tree crowns at both altitudes, providing a strong foundation for further diameter estimation. However, upon close inspection, some irregularities can be observed, particularly for trees positioned near the edges of the images. These trees occasionally exhibit inaccurate or incomplete crown detection, likely due to reduced visibility or distortions at the image boundaries.

Despite these peripheral detection issues, the key aspect of interest; detecting the farthest points of each tree crown has been accurately handled. The segmentation correctly identifies the outermost edges of the tree crowns, which is critical for the next step in our analysis: measuring the crown diameter. This precision ensures that the detected bounding boxes will allow for accurate calculations of tree crown size and distance, forming the basis for further linear regression analysis and comparison with ground-truth data.

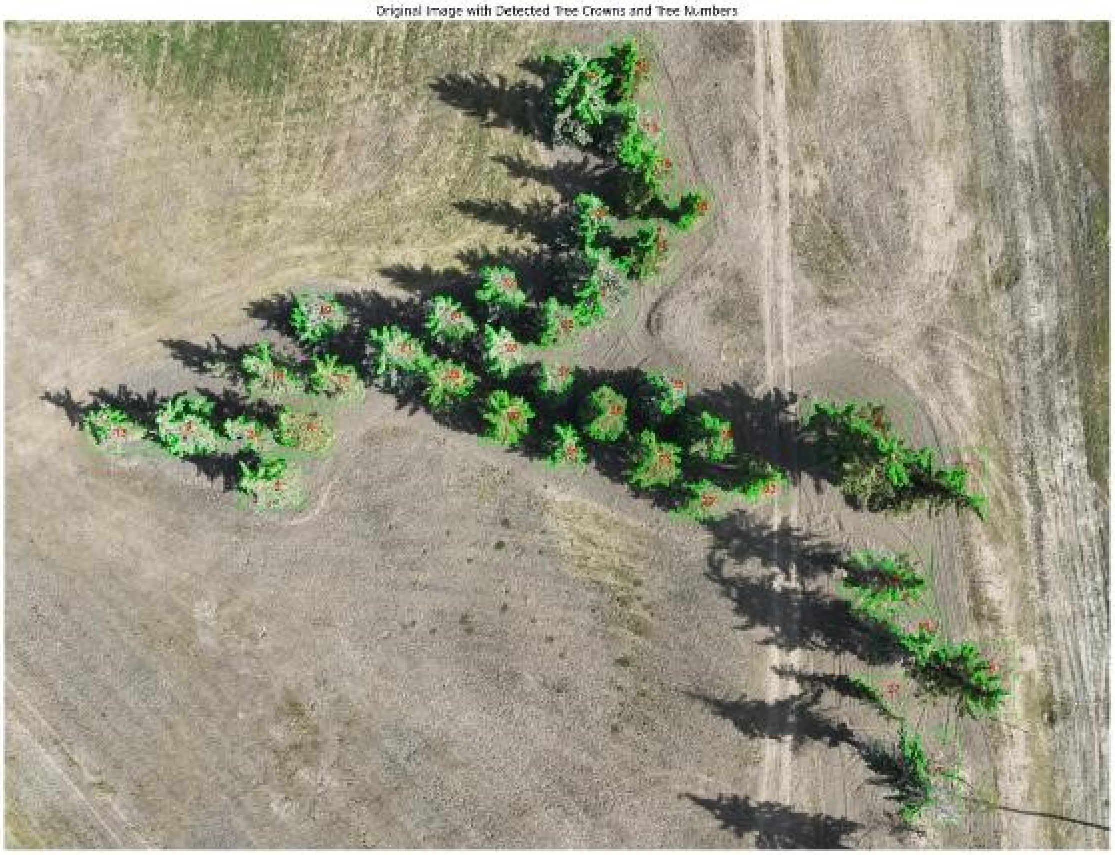


**Figure 18**: Watershed (WST) segmentation result of the example picture

Problem with this segmentation for Watershed technique, is that some segmentations are in the background instead of the tree itself (figure 18, right down corner of the image). This might be an inaccurate segmentation, but it still provides us with the information needed for calculating the TCD.

However, the second method with instance segmentation did a better job than the first one. In the figure, you can see that its segmentation was near perfect. With only one tree non detected.

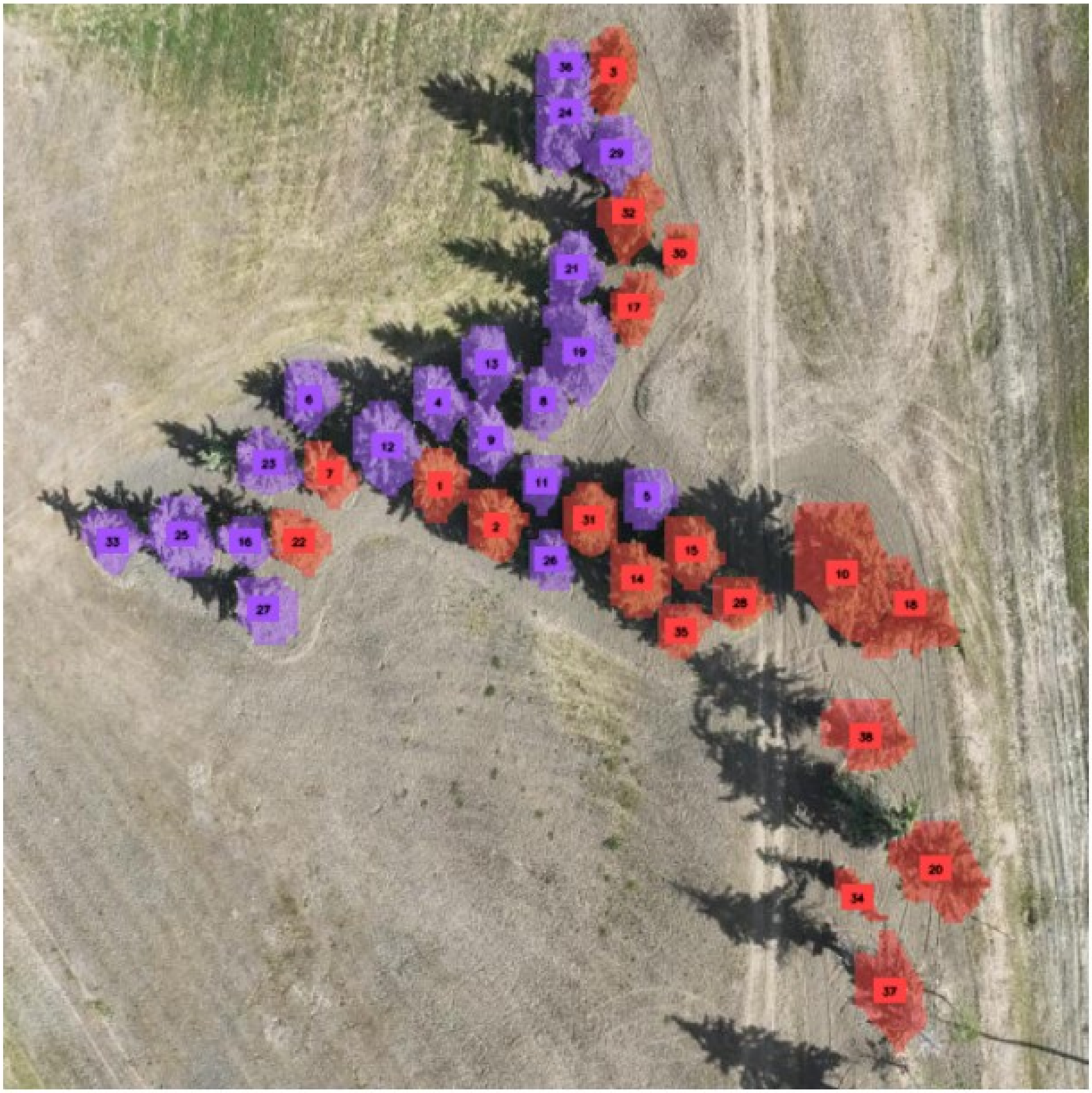


**Figure 19**: Instance segmentation result of the example picture

Purple and Red Segments are the spruce species which are white spruce (red) species or blue spruce (purple). The segmentation contours visible in the image closely follow the shape of individual tree crowns, showing that YOLOv11's instance segmentation can capture crown boundaries more accurately than WST, which might struggle with irregular shapes. WST also struggles due to its reliance on simple intensity gradients algorithm that separates the object (tree) from the background using only algorithm. YOLOv11 excels in handling complex structures, as seen with the different sizes and shapes of the tree crowns.

This segmentation result shows the superior capability of YOLOv11 instance segmentation over traditional Watershed techniques. The segmentation is more precise, handling irregular crown shapes and complex tree arrangements effectively. Additionally, the use of advanced bounding polygons ensures accurate identification without over-segmentation or overlapping regions. The plot below will show the visual differences between both methods.

Comparison between actual diameter and predicted in figure 20 and 21, is about comparing the predicted TCD to the real measurements that is already taken onsite. Note that the numbers of predicted trees aren't the same, this is because instance segmentation my took 2 very close trees into one tree. Which is valid in terms of forestry biology measurements. If the 2 trees are predicted to be one, we then can compare it to the real measurements and after combining those trees in the real measurements.

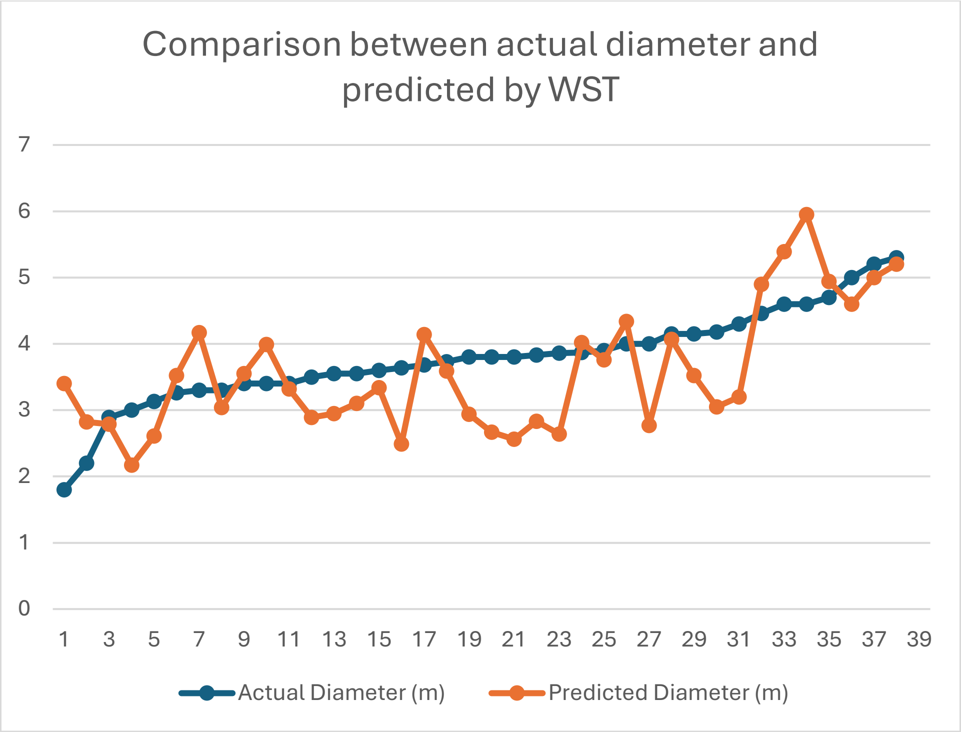


**Figure 20**: Comparison between actual diameter and predicted by WST of the example picture

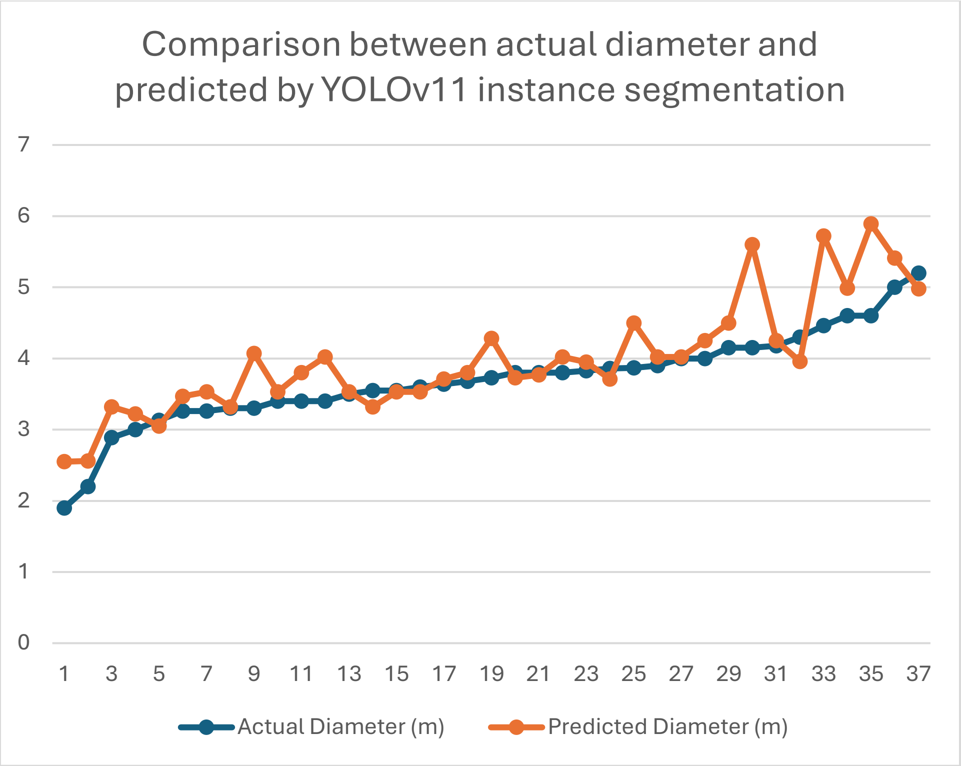


**Figure 21**: Comparison between actual diameter and predicted by YOLOv11 instance segmentation of the example picture

The plot of Comparison between actual diameter and predicted is organized by sorting the smallest actual diameter to largest. This will able us to see the nature of the graph lines in each method. We can identify the region that is struggles to maintain an accurate TCD predictions. We can see that the predicted diameter from instance segmentation is more stable than the WST segmentation. The values are much closer the actual ones. We can also notice that the values of WST are generally lower than the actual and underestimate the value. However, we can also comment on the last values of the prediction from YOLOv11 instance segmentation. We notice that the prediction starts to get instable when it reaches more than 4 meters in the actual diameter.

**Table 6:** Segmentation Comparison between Watershed and YOLOv11 via Diameter

| Technique | MAE | MSE | RMSE | MPE (%) | Accuracy (%) |
|---|---|---|---|---|---|
| Watershed Technique | 0.46 | 0.34 | 0.58 | 11.13 | 88.87 |
| YOLOv11 | 0.37 | 0.23 | 0.48 | 8.36 | 91.64 |

The results demonstrate that instance segmentation consistently outperforms the watershed technique. It achieves lower MAE (0.37 vs. 0.46), MSE (0.23 vs. 0.34), RMSE (0.48 vs. 0.58), and MPE (8.36% vs. 11.13%), indicating more accurate crown diameter predictions with fewer large errors.

Most notably, instance segmentation (in terms of  predicting TCD compared to the real measurements) achieves higher accuracy (91.64% vs. 88.87%), reflecting better boundary precision and reliable detection. In contrast, the watershed technique struggles with complex tree shapes and overlaps, leading to more segmentation errors. Overall, the results confirm that instance segmentation is a more precise and reliable approach for crown diameter estimation

The overall accuracy of TCD is good from both methods, although note that the segmentation of YOLOv11 targets only trees while watershed can target a part of a tree or the background. The accuracy reflects the bounding box of YOLOv5 which is fairly can estimate the diameter of tree crown. The reason of using instance segmentation is because it provides more accurate diameter and more information than the box. Which is the shape of the segmentation. A bounding box is a rectangle that surrounds the entire object, which means it cannot conform to the natural, irregular shape of a tree crown. This results in the inclusion of unnecessary background area, leading to overestimation of the crown diameter.

### 4.3 DBH Estimation from TCD using Machine learning

The relationship between tree crown diameter and DBH has long been recognized in forestry and it is concluded by researchers. However, the result may change depending on the location, weather, and soil of the spruce trees. Therefore, a new updated allometric relationship model for these trees is required. This is why we have measured DBH and crown diameter manually. This step is required to make our results more credible and accurate for these trees that are in our location.

By using a simple regression with excel, we noticed that the trend of the scatter chart matches the shape and curve of the studies that are made in literature review. However, the figure below didn't match the slope value of that study. Therefore, we will depend on our data and our regression model instead.

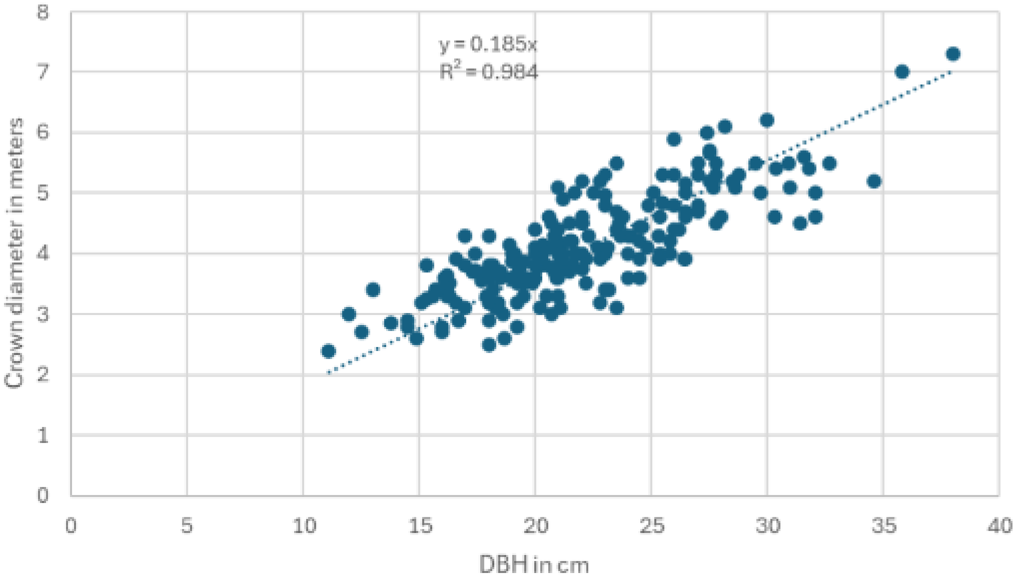


**Figure 22**: An overview on the measured crown diameter distribution in function of DBH in Regina

The scatter plot shown above illustrates the relationship between Diameter at Breast Height (DBH) in centimeters and crown diameter in meters. The trend line fitted to the data points follows a linear relationship, expressed by the equation:

$$y = 0.185x \qquad \text{(Equation 4-1)}$$

where y is the crown diameter, and x is the DBH. The coefficient of determination (R²) value is 0.984, indicating a very strong positive correlation between DBH and crown diameter. This suggests that as the DBH of a tree increases, its crown diameter also increases proportionally. The high R² value **of the slope** implies that 98.4% of the variability in crown diameter can be explained by the variation in DBH. The linear model is an effective predictor of crown diameter based on DBH, showing that these two parameters are closely related. This relationship is useful in forestry applications, as it allows for the estimation of crown size from easily measurable DBH, supporting tasks like biomass estimation, canopy cover analysis, and tree growth modeling. The scattered points around the line demonstrate a generally good fit with minimal deviation, further validating the accuracy of this linear model.

We tried different types of Machine Learning models. The table below table shows the results and evaluation of these models. We have implemented early stopping algorithm when training Neural Networks, to avoid overfitting. The stopping accrued around 65 epoch with batch of 16.

**Table 7:** Machine Learning models evaluation of predicting DBH

| ML Model | Parameters | MSE | RMSE | R^2 |
|---|---|---|---|---|
| KNN | n_neighbors = 5 | 10.5 | 3.24 | 0.27 |
| KNN | n_neighbors = 10 | 8.21 | 2.87 | 0.43 |
| KNN | n_neighbors = 15 | 7.24 | 2.69 | 0.5 |
| KNN | n_neighbors = 25 | 6.7 | 2.59 | 0.53 |
| NN | layers [64, 64] | 5.97 | 2.44 | 0.58 |
| NN | layers [128, 128] | 6.84 | 2.62 | 0.52 |

| | | | | |
|---|---|---|---|---|
| NN | layers [64, 64, 64] | 8.12 | 2.85 | 0.43 |
| NN | layers [128] | 6.55 | 2.56 | 0.54 |
| RF | random_state=42, n_estimators=50, max_depth=10 | 9.5 | 3.08 | 0.34 |
| RF | random_state=42, n_estimators=100, max_depth=15 | 9.37 | 3.06 | 0.35 |
| RF | random_state=42, n_estimators=200, max_depth=none | 9.42 | 3.07 | 0.34 |
| RF | random_state=42, n_estimators=100, max_depth=20 | 9.37 | 3.06 | 0.35 |

The best-performing NN neural network model is the one with [64, 64] layers, achieving the lowest MSE (5.97) and RMSE (2.44), along with the highest R² score (0.58). However, increasing the complexity of the network, such as using three layers (e.g., [64, 64, 64] configuration), did not improve performance and resulted in higher errors.

The KNN model with 25 neighbors showed the best performance among the KNN variations, with the lowest MSE (6.70), RMSE (2.59), and highest R² score (0.53). Increasing the number of neighbors generally reduced prediction errors and improved the R² score, indicating a better fit to the data.

Among the random forest (RF) models, the best performance was observed with the configuration of 200 estimators and max depth=None, achieving an MSE of 6.55, RMSE of 2.56, and an R² score of 0.54. While increasing the number of trees improved the model's performance, the improvement became marginal beyond a certain point.

In conclusion, for predicting DBH from crown diameter, the NN model with two hidden layers ([64, 64]) provides the best balance between accuracy and model complexity based on this evaluation.

### 4.4 Summary of Key Findings

This chapter presented the results and key findings from the research, focusing on tree detection, crown diameter estimation, and the prediction of Diameter at Breast Height (DBH) using machine learning models. So, we can use it to calculate the biomass of spruce trees.

The YOLOv11 instance segmentation model demonstrated high accuracy in detecting and segmenting individual tree crowns. The model outperformed YOLOv5 + WST technique by achieving better precision and minimizing segmentation errors, even in complex environments with overlapping crowns.

Therefore, an accurate estimation of Tree Crown Diameter (TCD) was achieved through the YOLOv11 model's precise segmentation. The results showed a strong correlation between the model's predicted TCD and the actual measurements, validating the model's reliability for tree crown size estimation.

Using the estimated TCD, various machine learning models were evaluated to predict DBH. Among the models tested, the neural network with two hidden layers ([64, 64]) provided the best balance between accuracy and model complexity. This approach allows for efficient DBH estimation, which is essential for biomass calculations.

# Chapter Five: Discussion and Conclusion

In this chapter we will give a summery conclusion of this research, limitation of this project and what are future work.

## 5.1 General Conclusion

We conclude that YOLOv11 preforms better than YOLOv5 + WST in terms of predicting TCD by 91.64% in accuracy. We have chosen YOLO instead of MaskRCNN because we wanted real time fast detection and the detection of MaskRCNN is worse than YOLO. Average segmentation is 3 seconds depending on the number of trees. We also conclude that Neural Networks with two hidden layers ([64, 64]) gives the best DBH prediction with lowest MSE (5.97) and RMSE (2.44).

## 5.2 Limitations of the Study

The limitation are as follows:

- Limited Scope to Canadian Spruce Trees in Regina, Saskatchewan: The focus of this study is restricted to a specific geographical location and tree species. While the results provide valuable insights into tree segmentation and detection in urban environments, the findings may not be fully generalizable to other regions or tree species without further testing and validation.
- Data Collection Restrictions Due to Flight Area Limitations: The use of Unmanned Aerial Vehicles (UAVs) for data collection was constrained by flight area restrictions in urban environments. This limited the scope of the dataset, reducing the possibility of collecting more extensive and diverse data, which could have improved the model's robustness across various urban forestry conditions.
- Challenges in Gathering DBH and TCD for Model Accuracy: The manual collection for validating the model's accuracy is highly labour-intensive and time-consuming. It also

involved physical effort, planning, and exposure to natural risks such as bug stings, making it difficult to gather enough samples to fully assess the model's performance.

- Inability to fully detect the trees that are in edges of the image in dataset. We could perform cropping of the extremities, but it will ruin the area of study.
- Inability to Evaluate Biomass Estimation: This is due to the inaccessibility of the necessary data on the wood weight of the Canadian Spruce species, which is essential for accurate biomass calculations. Without the actual wood weight, it was not possible to provide a reliable estimate of spruce dry biomass.
- No studies to compare with: In case there will be another researcher who uses our open-source segmented dataset in Roboflow: https://app.roboflow.com/instance-segmentation-spruce-day-1

### 5.3 Possible Future Research

Orthogonal dataset: this can produce easier training and segmentation. But it requires a lot of data processing which could be another whole research area. But this method will eliminate the trees that are inclined in the extremities of the image. Also, this method could give inaccuracies in term of pixel distances. Therefor, this method is a solution for easier training but provokes other problems such as TCD inaccuracies.

We can use this technique for more lucrative projects such as agriculture. Which can monitor the yield of fruits and vegetables and provides with early net worth gain before harvesting.

# References

[1] Widlowski J, Verstraete M, Pinty B, Gobron N. Allometry of Selected European Tree Speces: Betula pensula, Fagus selvatica, Larix decidua, Picea abies, Pinus sylvestris. EUR 20855 EN. 2003. JRC26286

[2] Sharma RP, Bílek L, Vacek Z, Vacek S. Modelling crown width–diameter relationship for Scots pine in the central Europe. Trees. 2017 Dec;31:1875-89.

[3] Colin F, Houllier F. Branchiness of Norway spruce in northeastern France: predicting the main crown characteristics from usual tree measurements. InAnnales des sciences forestières 1992 (Vol. 49, No. 5, pp. 511-538). EDP Sciences.

[4] Filipescu CN, Groot A, MacIsaac DA, Cruickshank MG, Stewart JD. Prediction of diameter using height and crown attributes: a case study. Western Journal of Applied Forestry. 2012 Jan 1;27(1):30-5.

[5] Hall RJ, Morton RT, Nesby RN. A comparison of existing models for DBH estimation from large-scale photos. The Forestry Chronicle. 1989 Apr 1;65(2):114-20.

[6] Shimano K. Analysis of the relationship between DBH and crown projection area using a new model. Journal of Forest Research. 1997 Nov 1;2(4):237-42.

[7] Gill SJ, Biging GS, Murphy EC. Modeling conifer tree crown radius and estimating canopy cover. Forest ecology and management. 2000 Feb 25;126(3):405-16.

[8] Sönmez T. Diameter at breast height-crown diameter prediction models for Picea orientalis.

[9] Vezina PE. Crown width-DBH relationships for open-grown balsam fir and white spruce in Quebec. The Forestry Chronicle. 1962 Dec 1;38(4):463-73.

[10] Solares-Canal A, Alonso L, Picos J, Armesto J. Individual Tree Identification and Segmentation in Pinus spp. Stands through Portable LiDAR. Forests. 2024 Jun 28;15(7):1133.

[11] Yang J, Gan R, Luo B, Wang A, Shi S, Du L. An Improved Method for Individual Tree Segmentation in Complex Urban Scene Based on Using Multispectral LiDAR by Deep Learning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2024 Mar 7.

[12] Zhu D, Liu X, Zheng Y, Xu L, Huang Q. Improved Tree Segmentation Algorithm Based on Backpack-LiDAR Point Cloud. Forests. 2024 Jan 9;15(1):136.

[13] Fu Y, Niu Y, Wang L, Li W. Individual-Tree Segmentation from UAV–LiDAR Data Using a Region-Growing Segmentation and Supervoxel-Weighted Fuzzy Clustering Approach. Remote Sensing. 2024 Feb 6;16(4):608.

[14] Deng S, Xu Q, Yue Y, Jing S, Wang Y. Individual tree detection and segmentation from unmanned aerial vehicle-LiDAR data based on a trunk point distribution indicator. Computers and Electronics in Agriculture. 2024 Mar 1;218:108717.

[15] Deng S, Jing S, Zhao H. A Hybrid Method for Individual Tree Detection in Broadleaf Forests Based on UAV-LiDAR Data and Multistage 3D Structure Analysis. Forests. 2024 Jun 17;15(6):1043.

[16] Yu J, Lei L, Li Z. Individual Tree Segmentation Based on Seed Points Detected by an Adaptive Crown Shaped Algorithm Using UAV-LiDAR Data. Remote Sensing. 2024 Feb 27;16(5):825.

[17] Dersch S, Schöttl A, Krzystek P, Heurich M. Semi-supervised multi-class tree crown delineation using aerial multispectral imagery and lidar data. ISPRS Journal of Photogrammetry and Remote Sensing. 2024 Oct 1;216:154-67.

[18] Zhang C, Song C, Zaforemska A, Zhang J, Gaulton R, Dai W, Xiao W. Individual tree segmentation from UAS Lidar data based on hierarchical filtering and clustering. International Journal of Digital Earth. 2024 Dec 31;17(1):2356124.

[19] Chen Q, Luo H, Cheng Y, Xie M, Nan D. An Individual Tree Detection and Segmentation Method from TLS and MLS Point Clouds Based on Improved Seed Points. Forests. 2024 Jun 22;15(7):1083.

[20] Burmeister JM, Richter R, Reder S, Mund JP, Döllner J. Tree Instance Segmentation in Urban 3D Point Clouds Using a Coarse-to-Fine Algorithm Based on Semantic Segmentation. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2024 Jun 27;10:79-86.

[21] Cheng D, Cladera F, Prabhu A, Liu X, Zhu A, Green PC, Ehsani R, Chaudhari P, Kumar V. TreeScope: An agricultural robotics dataset for LiDAR-based mapping of trees in forests and orchards. In2024 IEEE International Conference on Robotics and Automation (ICRA) 2024 May 13 (pp. 14860-14866). IEEE.

[22] Wang L, Zhang R, Zhang L, Yi T, Zhang D, Zhu A. Research on Individual Tree Canopy Segmentation of Camellia oleifera Based on a UAV-LiDAR System. Agriculture. 2024 Feb 24;14(3):364.

[23] Saeed T, Hussain E, Ullah S, Iqbal J, Atif S, Yousaf M. Performance evaluation of individual tree detection and segmentation algorithms using ALS data in Chir Pine (Pinus roxburghii) forest. Remote Sensing Applications: Society and Environment. 2024 Apr 1;34:101178.

[24] Li Q, Yan Y. Street tree segmentation from mobile laser scanning data using deep learning-based image instance segmentation. Urban Forestry & Urban Greening. 2024 Feb 1;92:128200.

[25] Seidl J, Kačmařík M, Klimánek M. A Tree Segmentation Algorithm for Airborne Light Detection and Ranging Data Based on Graph Theory and Clustering. Forests. 2024 Jun 27;15(7):1111.

[26] Terekhov V, Bondarenko D, Ryzhkova I, Zelinskii D. Tree segmentation of LiDAR point clouds using a graph-based algorithm. In2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE) 2024 Feb 29 (pp. 1-6). IEEE.

[27] Kurdi FT, Lewandowicz E, Shan J, Gharineiat Z. 3D modeling and visualization of single tree Lidar point cloud using matrixial form. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2024 Jan 4.

[28] Tarsha Kurdi F, Gharineiat Z, Lewandowicz E, Shan J. Modeling the Geometry of Tree Trunks Using LiDAR Data. Forests. 2024 Feb 16;15(2):368.

[29] Zhu Y, Lin Y, Chen B, Yun T, Wang X. Synergizing a Deep Learning and Enhanced Graph-Partitioning Algorithm for Accurate Individual Rubber Tree-Crown Segmentation from Unmanned Aerial Vehicle Light-Detection and Ranging Data. Remote Sensing. 2024 Jul 31;16(15):2807.

[30] Shao J, Lin YC, Wingren C, Shin SY, Fei W, Carpenter J, Habib A, Fei S. Large-scale Inventory in Natural Forests with Mobile LiDAR Point Clouds. Science of Remote Sensing. 2024 Oct 9:100168.

[31] Fallah M, Aghighi H, Matkan A. Advancements in individual tree detection and forest structural attributes estimation from LiDAR Data: MSITD and SAFER approaches. Earth and Space Science. 2024 Mar;11(3):e2023EA003306.

[32] Wang Z, Chen H, Liu J, Qin J, Sheng Y, Yang L. Multilevel intuitive attention neural network for airborne LiDAR point cloud semantic segmentation. International Journal of Applied Earth Observation and Geoinformation. 2024 Aug 1;132:104020.

[33] Tang S, Ao Z, Li Y, Huang H, Xie L, Wang R, Wang W, Guo R. TreeNet3D: A large scale tree benchmark for 3D tree modeling, carbon storage estimation and tree segmentation. International Journal of Applied Earth Observation and Geoinformation. 2024 Jun 1;130:103903.

[34] Sun H, Ye Q, Chen Q, Fu L, Xu Z, Hu C. Tree Canopy Volume Extraction Fusing ALS and TLS Based on Improved PointNeXt. Remote Sensing. 2024 Jul 19;16(14):2641.

[35] Mukhandi H, Ferreira JF, Peixoto P. SyS3DS: Systematic Sampling of Large-Scale LiDAR Point Clouds for Semantic Segmentation in Forestry Robotics. Sensors. 2024 Jan 26;24(3):823.

[36] Wielgosz M, Puliti S, Xiang B, Schindler K, Astrup R. SegmentAnyTree: A sensor and platform agnostic deep learning model for tree segmentation using laser scanning data. arXiv preprint arXiv:2401.15739. 2024 Jan 28.

[37] Yan Y, Lei J, Jin J, Shi S, Huang Y. Unmanned Aerial Vehicle–Light Detection and Ranging-Based Individual Tree Segmentation in Eucalyptus spp. Forests: Performance and Sensitivity. Forests. 2024 Jan 20;15(1):209.

[38] Condat R, Vasseur P, Allibert G. Focusing on Object Extremities for Tree Instance Segmentation in Forest Environments. IEEE Robotics and Automation Letters. 2024 Apr 24.

[39] Guo Z, Shi Y, Ahmad I. Design of smart citrus picking model based on Mask RCNN and adaptive threshold segmentation. PeerJ Computer Science. 2024 Mar 4;10:e1865.

[40] Lavania P, Singh RK, Kumar P, Savad K, Gupta G, Dobriyal M, Pandey AK, Kumar M, Singh S. Unveiling the green guardians: Mapping and identification of Azadirachta indica trees with semantic segmentation deep learning neural network technique. The Egyptian Journal of Remote Sensing and Space Sciences. 2024 Sep 1;27(3):491-500.

[41] Chen J, Ji C, Zhang J, Feng Q, Li Y, Ma B. A method for multi-target segmentation of bud-stage apple trees based on improved YOLOv8. Computers and Electronics in Agriculture. 2024 May 1;220:108876.

[42] Bu X, Liu C, Liu H, Yang G, Shen Y, Xu J. DFSNet: A 3D Point Cloud Segmentation Network toward Trees Detection in an Orchard Scene. Sensors. 2024 Mar 31;24(7):2244.

[43] Liu H, Li W, Jia W, Sun H, Zhang M, Song L, Gui Y. Clusterformer for pine tree disease identification based on UAV remote sensing image segmentation. IEEE Transactions on Geoscience and Remote Sensing. 2024 Feb 6.

[44] Steier J, Goebel M, Iwaszczuk D. Is Your Training Data Really Ground Truth? A Quality Assessment of Manual Annotation for Individual Tree Crown Delineation. Remote Sensing. 2024 Jul 30;16(15):2786.

[45] Arakawa T, Tanaka TS, Kamio S. Detection of on-tree chestnut fruits using deep learning and RGB unmanned aerial vehicle imagery for estimation of yield and fruit load. Agronomy Journal. 2024 May;116(3):973-81.

[46] Moysiadis V, Siniosoglou I, Kokkonis G, Argyriou V, Lagkas T, Goudos SK, Sarigiannidis P. Cherry Tree Crown Extraction Using Machine Learning Based on Images from UAVs. Agriculture. 2024 Feb 18;14(2):322.

[47] Wu W, He Z, Li J, Chen T, Luo Q, Luo Y, Wu W, Zhang Z. Instance Segmentation of Tea Garden Roads Based on an Improved YOLOv8n-seg Model. Agriculture. 2024;14(7):1163.

[48] Zhang Y, Wang M, Mango J, Xin L, Meng C, Li X. Individual tree detection and counting based on high-resolution imagery and the canopy height model data. Geo-spatial Information Science. 2024 Jan 21:1-7.

[49] Xu J, Su M, Sun Y, Pan W, Cui H, Jin S, Zhang L, Wang P. Tree Crown Segmentation and Diameter at Breast Height Prediction Based on BlendMask in Unmanned Aerial Vehicle Imagery. Remote Sensing. 2024 Jan 16;16(2):368.

[50] Mai Y, Zheng J, Luo Z, Yu C, Lu J, Yu C, Lin Z, Liao Z. Taoism-Net: A Fruit Tree Segmentation Model Based on Minimalism Design for UAV Camera. Agronomy. 2024 May 28;14(6):1155.

[51] Vasavi S, Likhitha AL, Premchand VS, Yasaswini J. Object Classification by Effective Segmentation of Tree Canopy Using U-Net Model. Journal of Advances in Information Technology. 2024;15(3).

[52] Cloutier M, Germain M, Laliberté E. Influence of temperate forest autumn leaf phenology on segmentation of tree species from UAV imagery using deep learning. Remote Sensing of Environment. 2024 Sep 1;311:114283.

[53] Zhang H, Liu S. Double-Branch Multi-Scale Contextual Network: A Model for Multi-Scale Street Tree Segmentation in High-Resolution Remote Sensing Images. Sensors. 2024 Feb 8;24(4):1110.

[54] Nashat AA, Mazen FM. Instance Segmentation and Classification of Coffee Leaf Plant using Mask RCNN and Transfer Learning. Fayoum University Journal of Engineering. 2024 Jan 1;7(1):130-41.

[55] Lin Y, Xue B, Zhang M, Schofield S, Green R. Drone Stereo Vision for Radiata Pine Branch Detection and Distance Measurement: Integrating SGBM and Segmentation Models. arXiv preprint arXiv:2409.17526. 2024 Sep 26.

[56] Zhang C, Zhang Y, Liang S, Liu P. Research on Key Algorithm for Sichuan Pepper Pruning Based on Improved Mask R-CNN. Sustainability. 2024 Apr 19;16(8):3416.

[57] Wang J, Zhang H, Liu Y, Zhang H, Zheng D. Tree-Level Chinese Fir Detection Using UAV RGB Imagery and YOLO-DCAM. Remote Sensing. 2024 Jan 14;16(2):335.

[58] Prousalidis K, Bourou S, Velivassaki TH, Voulkidis A, Zachariadi A, Zachariadis V. Olive Tree Segmentation from UAV Imagery. Drones. 2024 Aug 21;8(8):408.

[59] Khan Z, Liu H, Shen Y, Zeng X. Deep learning improved YOLOv8 algorithm: Real-time precise instance segmentation of crown region orchard canopies in natural environment. Computers and Electronics in Agriculture. 2024 Jun 19:109168.

[60] Zhu F, Chen Z, Li H, Shi Q, Liu X. CEDAnet: Individual tree segmentation in dense orchard via context enhancement and density prior. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2024 Mar 19.

[61] Yang T, Zhou S, Xu A, Ye J, Yin J. YOLO-SegNet: A Method for Individual Street Tree Segmentation Based on the Improved YOLOv8 and the SegFormer Network. Agriculture. 2024 Sep 15;14(9):1620.

[62] Xie Y, Wang Y, Sun Z, Liang R, Ding Z, Wang B, Huang S, Sun Y. Instance segmentation and stand-scale forest mapping based on UAV images derived RGB and CHM. Computers and Electronics in Agriculture. 2024 May 1;220:108878.

[63] Deka B, Chakraborty D. UAV Sensing-Based Litchi Segmentation Using Modified Mask-RCNN for Precision Agriculture. IEEE Transactions on AgriFood Electronics. 2024 Jul 12.

[64] Fu H, Zhao H, Jiang J, Zhang Y, Liu G, Xiao W, Du S, Guo W, Liu X. Automatic detection tree crown and height using Mask R-CNN based on unmanned aerial vehicles images for biomass mapping. Forest Ecology and Management. 2024 Mar 1;555:121712.

[65] Wen F, Tian J, Troles J, Döllerer M, Kindu M, Knoke T. Comparing Deep Learning and MCWST Approaches for Individual Tree Crown Segmentation. InInternational Society for Photogrammetry and Remote Sensing 2024 (pp. 1-7).

[66] Chadwick AJ, Coops NC, Bater CW, Martens LA, White B. Transferability of a mask R–CNN model for the delineation and classification of two species of regenerating tree crowns to untrained sites. Science of Remote Sensing. 2024 Jun 1;9:100109.

[67] Chen W, Guan Z, Gao D. Att-Mask R-CNN: an individual tree crown instance segmentation method based on fused attention mechanism. Canadian Journal of Forest Research. 2024 Jun 4.

[68] Sapkota R, Ahmed D, Karkee M. Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. Artificial Intelligence in Agriculture. 2024 Sep 1;13:84-99.

[69] Priyanto H. YOLO11 Architecture - Detailed Explanation. https://www.youtube.com/watch?v=L9Va7Y9UT8E&t=78s

[70] Waleed A. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. {\url{https://github.com/matterport/Mask_RCNN}

[71] Strahinja Z. #007 OpenCV projects – Image segmentation with Watershed algorithm. \url{https://datahacker.rs/007-opencv-projects-image-segmentation-with-watershed-algorithm/}

# Appendices

## Appendix A: Other Results Samples of this project

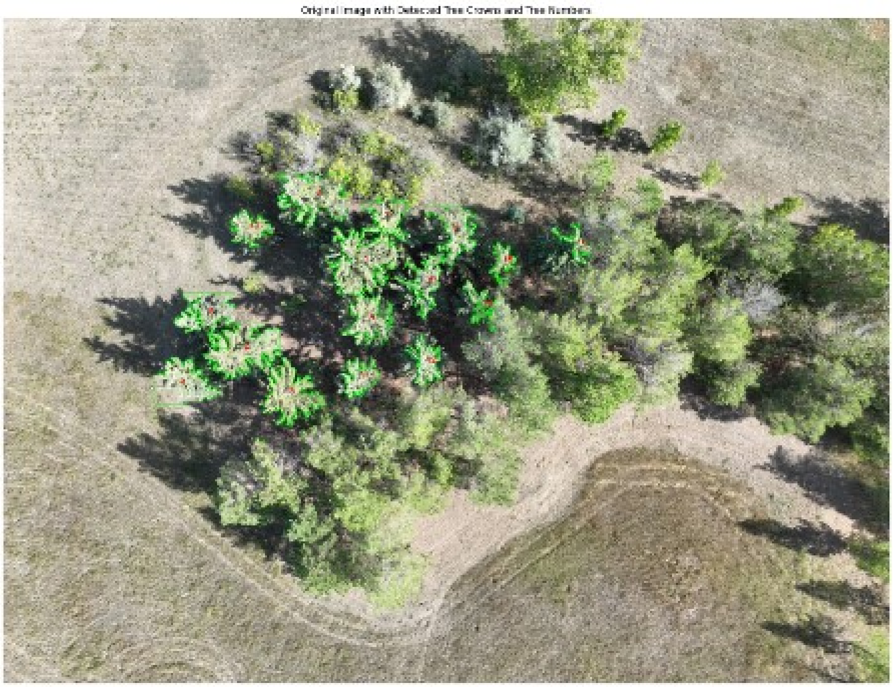


**Figure 23**: YOLOv5 + WST segment results 1

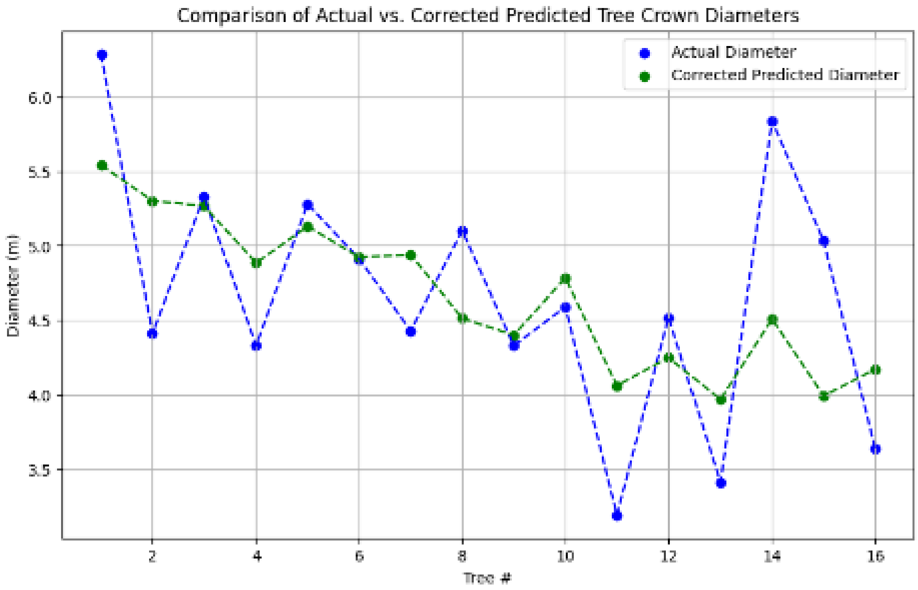


**Figure 24**: YOLOv5 + WST segment results 1 comparison

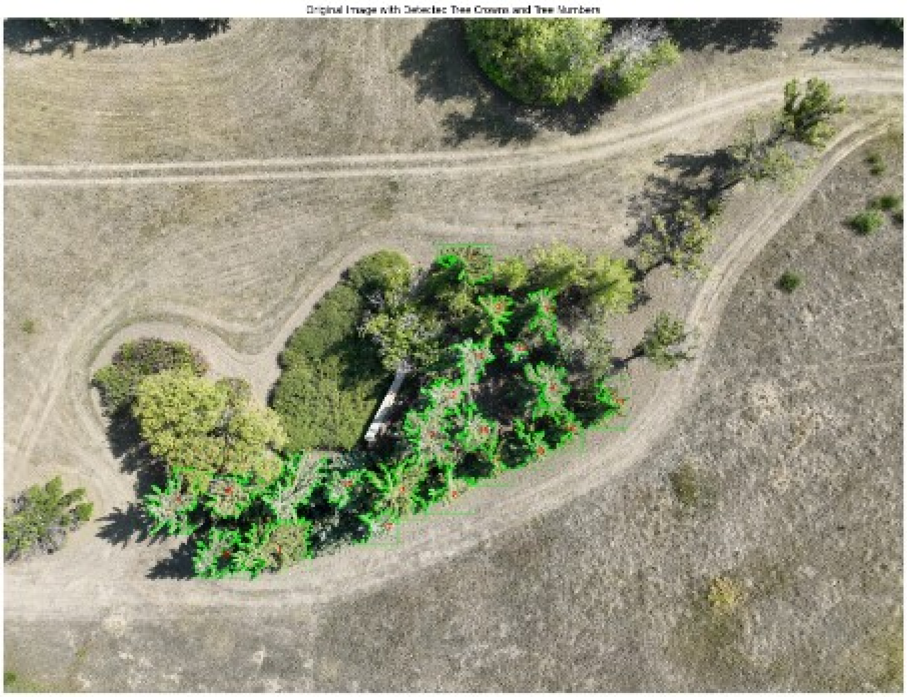


**Figure 25:** YOLOv5 + WST segment results 2

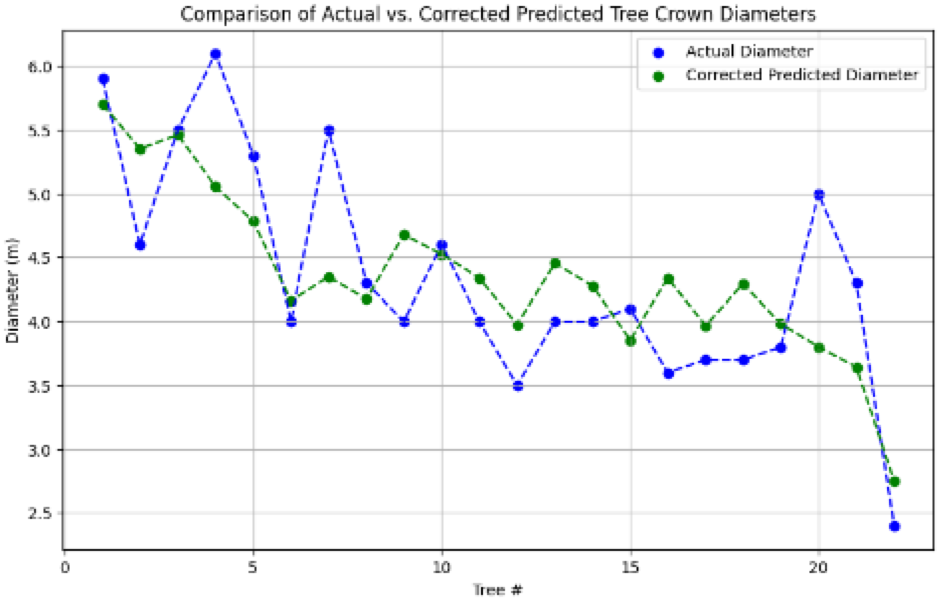


**Figure 26**: YOLOv5 + WST segment results 2 comparison

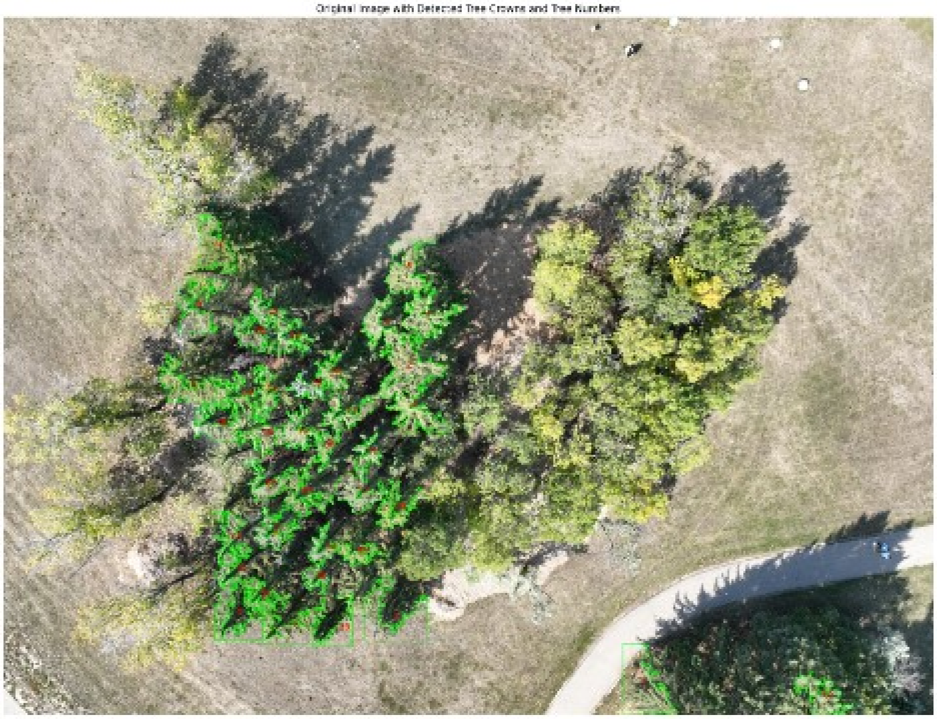


**Figure 27**: YOLOv5 + WST segment results 3

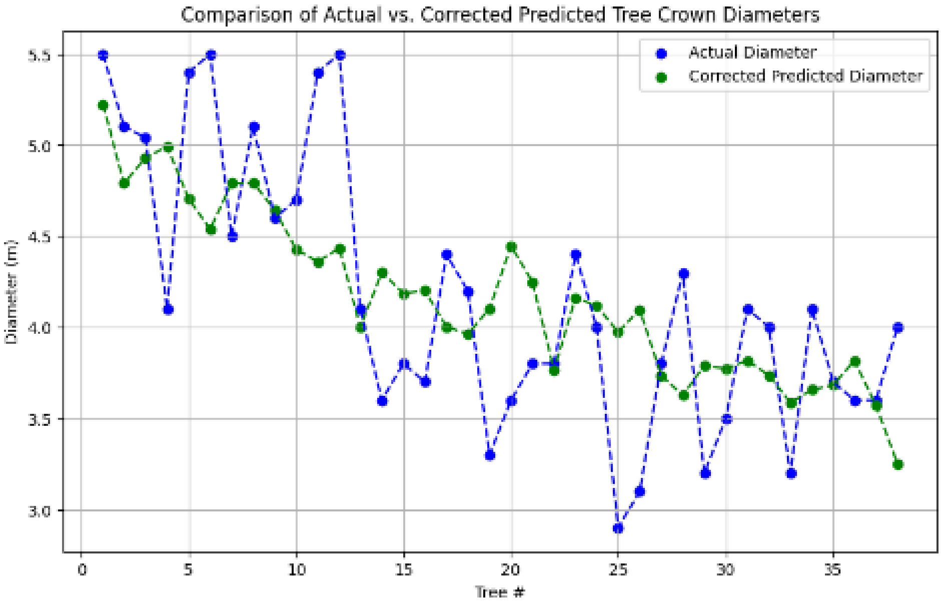


**Figure 28**: YOLOv5 + WST segment results 3 comparison

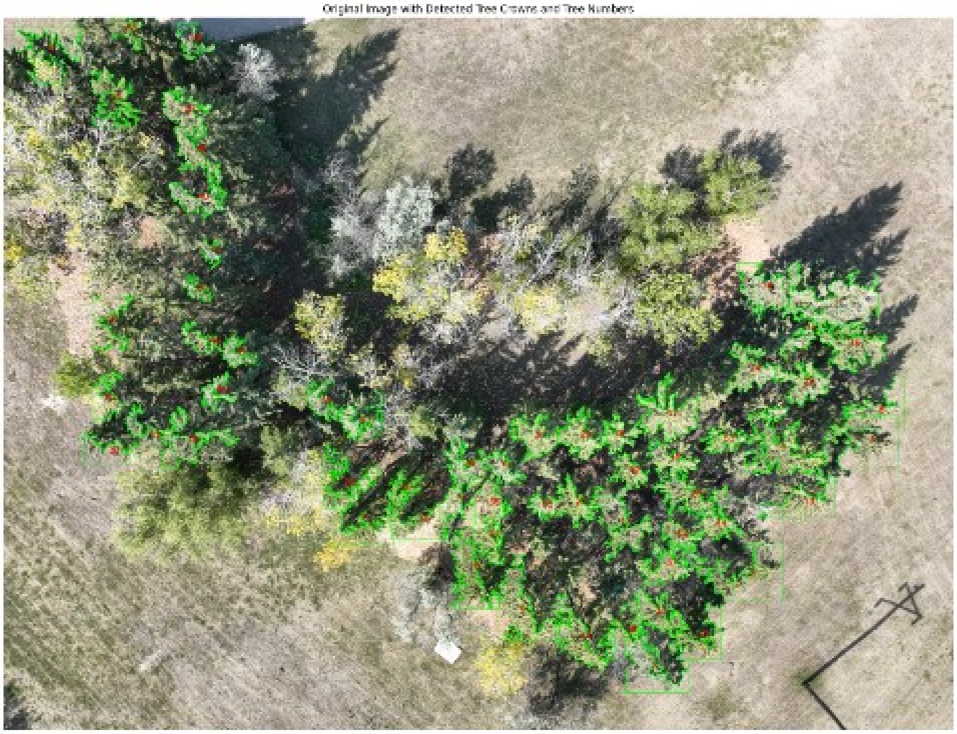


**Figure 29**: YOLOv5 + WST segment results 4

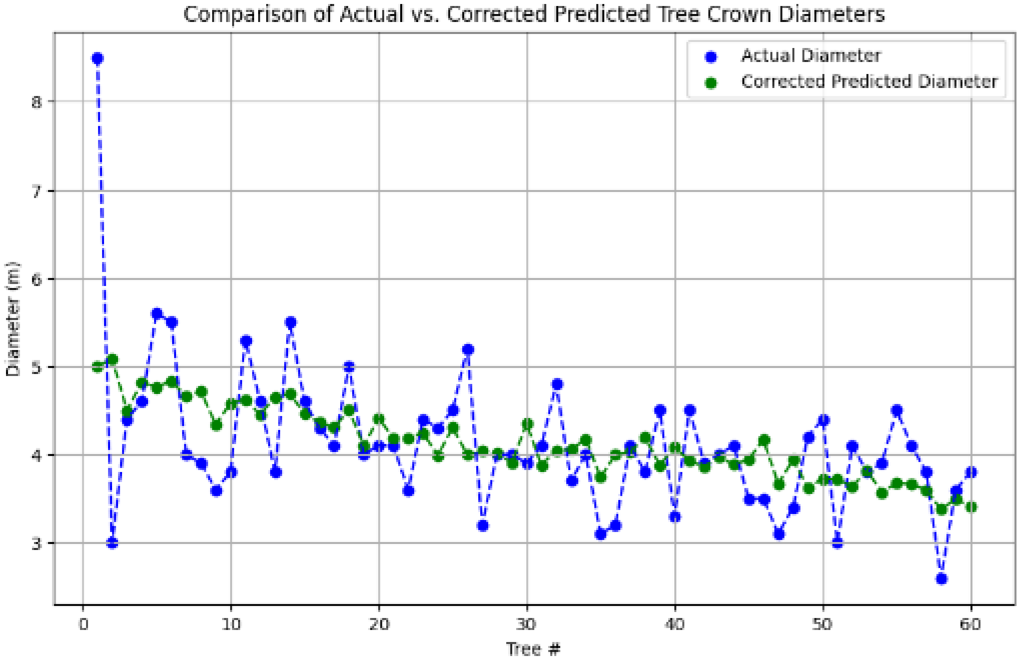


**Figure 30**: YOLOv5 + WST segment results 4 comparison

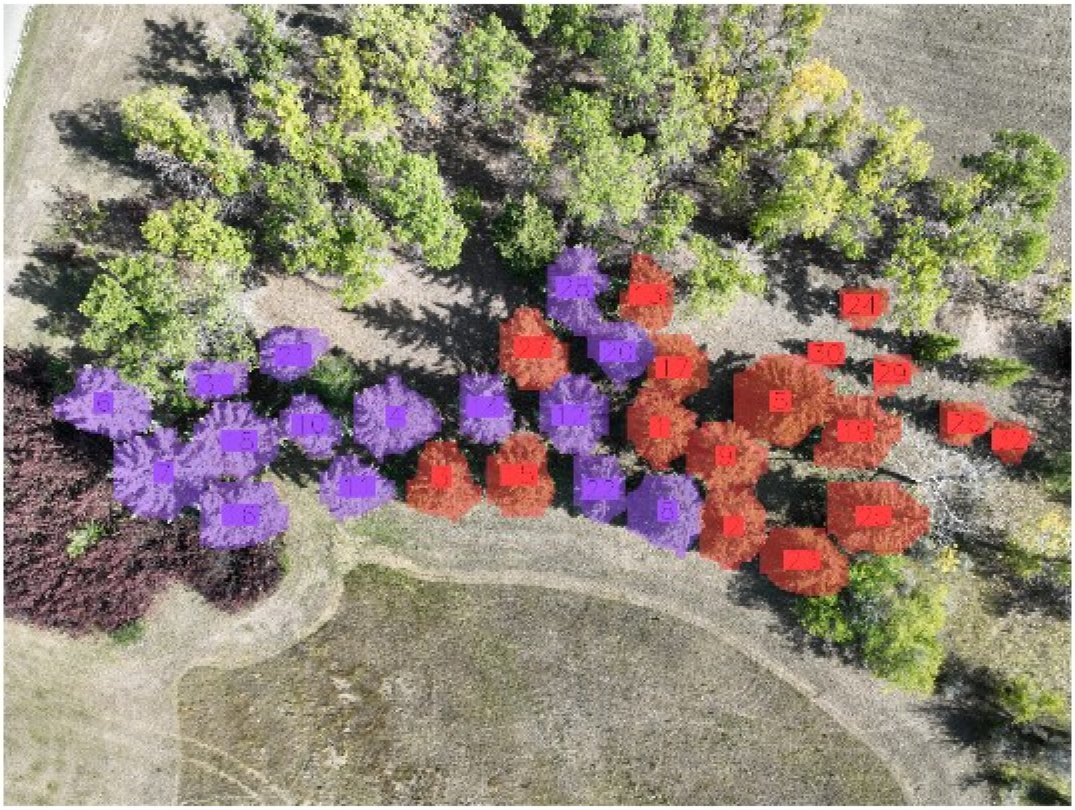


**Figure 31**: YOLOv11 instance segmentation example 1 segmented image

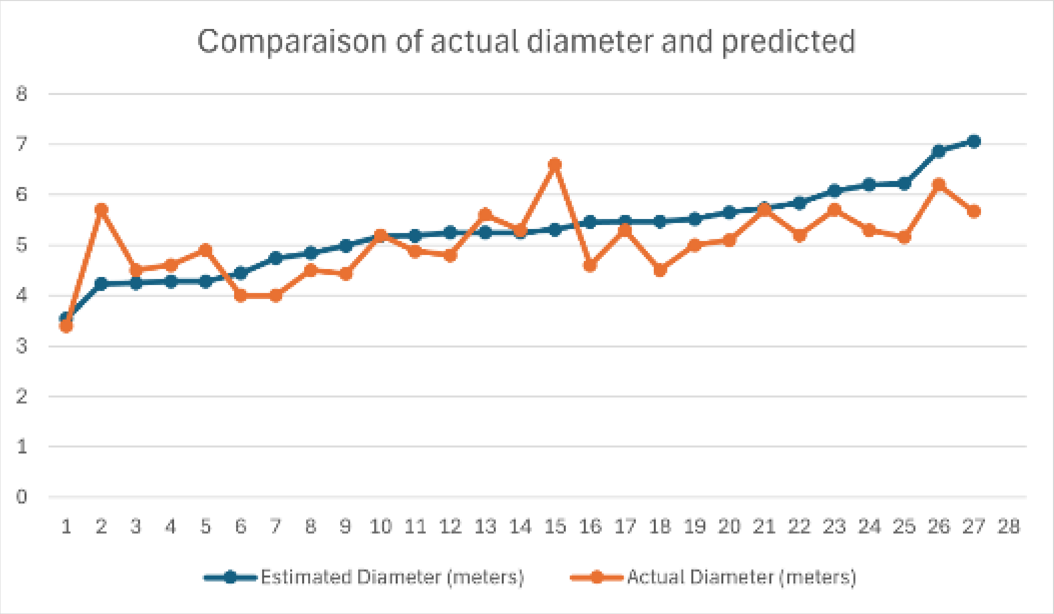


**Figure 32:** YOLOv11 instance segmentation example 1 comparison

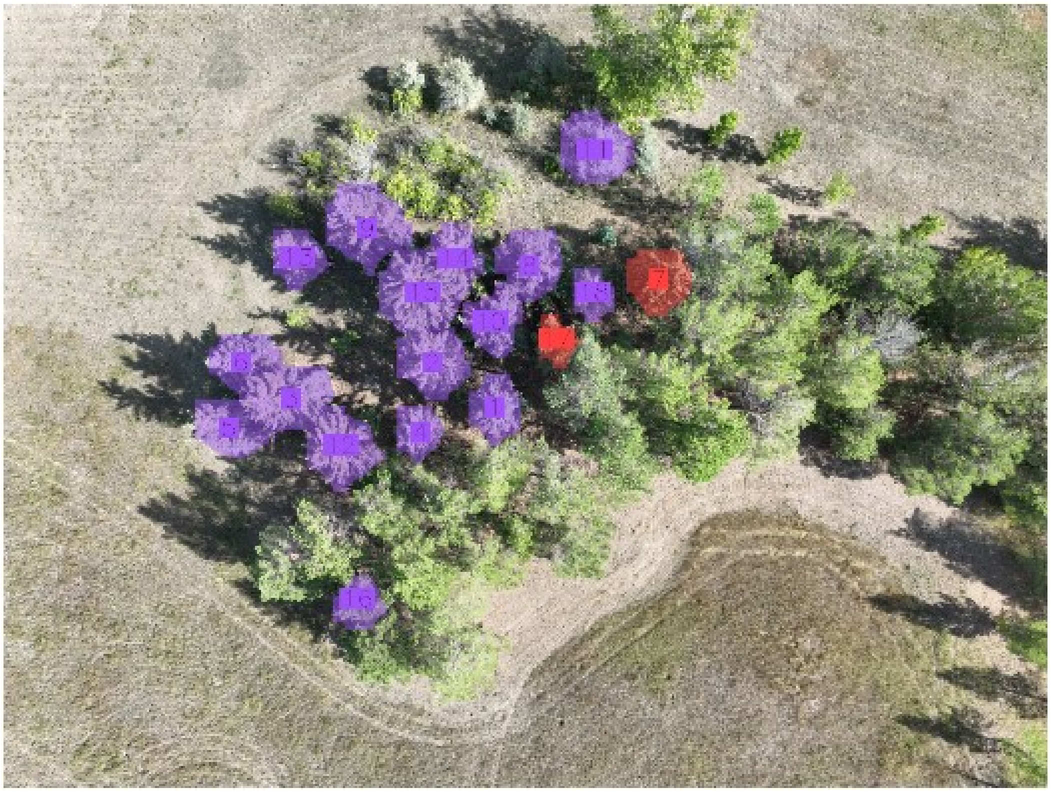


**Figure 33**: YOLOv11 instance segmentation example 2 segmented image

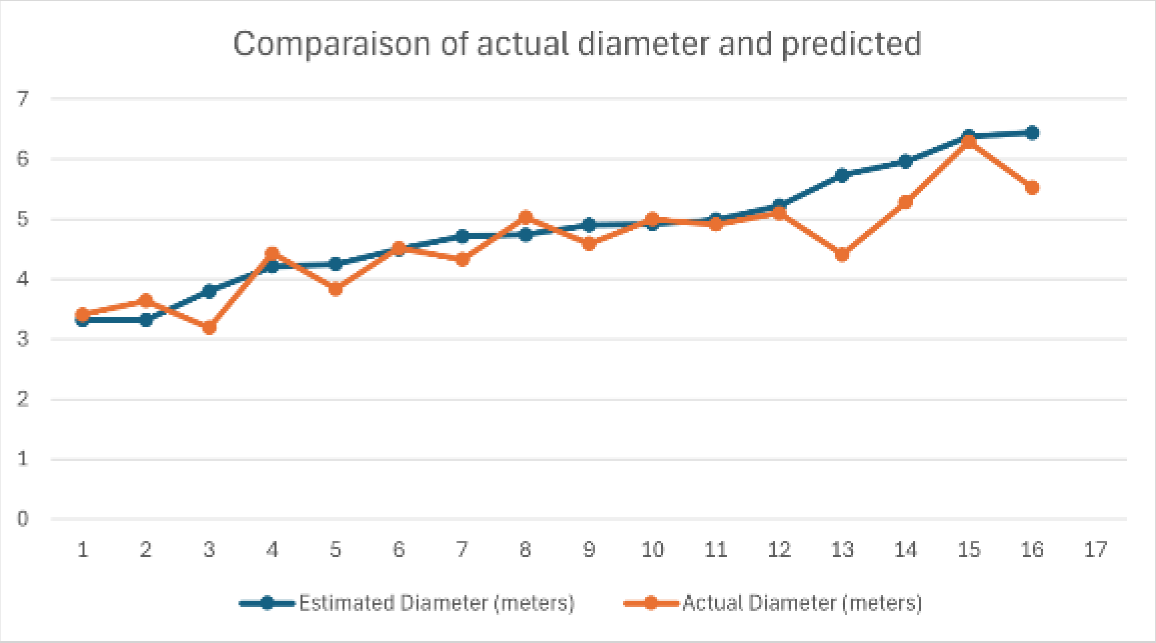


**Figure 34**: YOLOv11 instance segmentation example 2 comparison

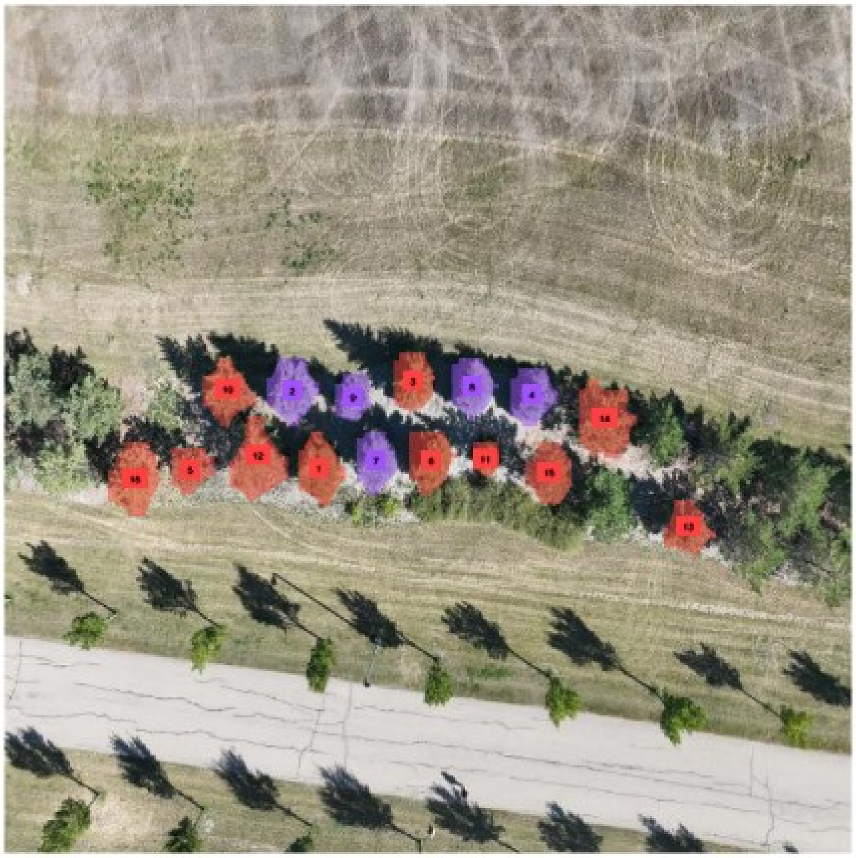


**Figure 35**: YOLOv11 instance segmentation example 3 segmented image

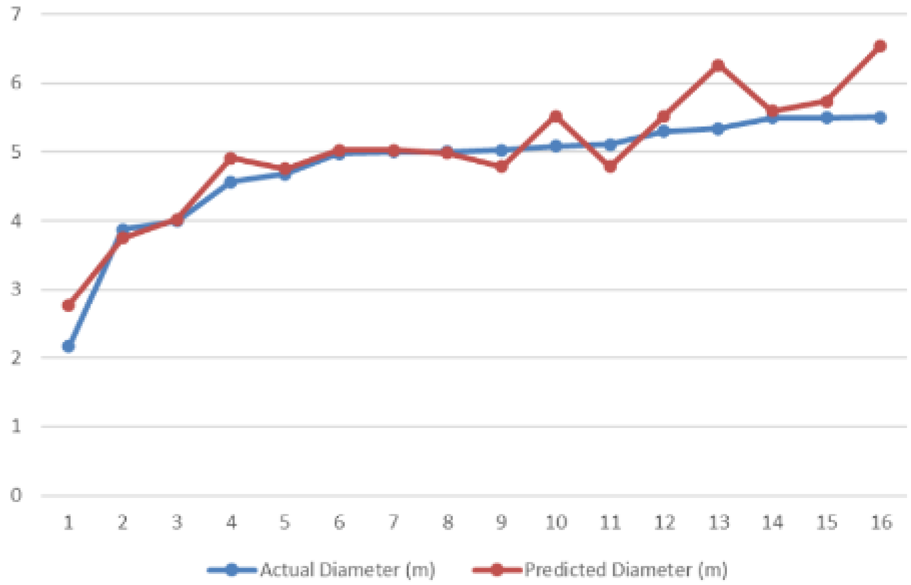


**Figure 36**: YOLOv11 instance segmentation example 3 comparison

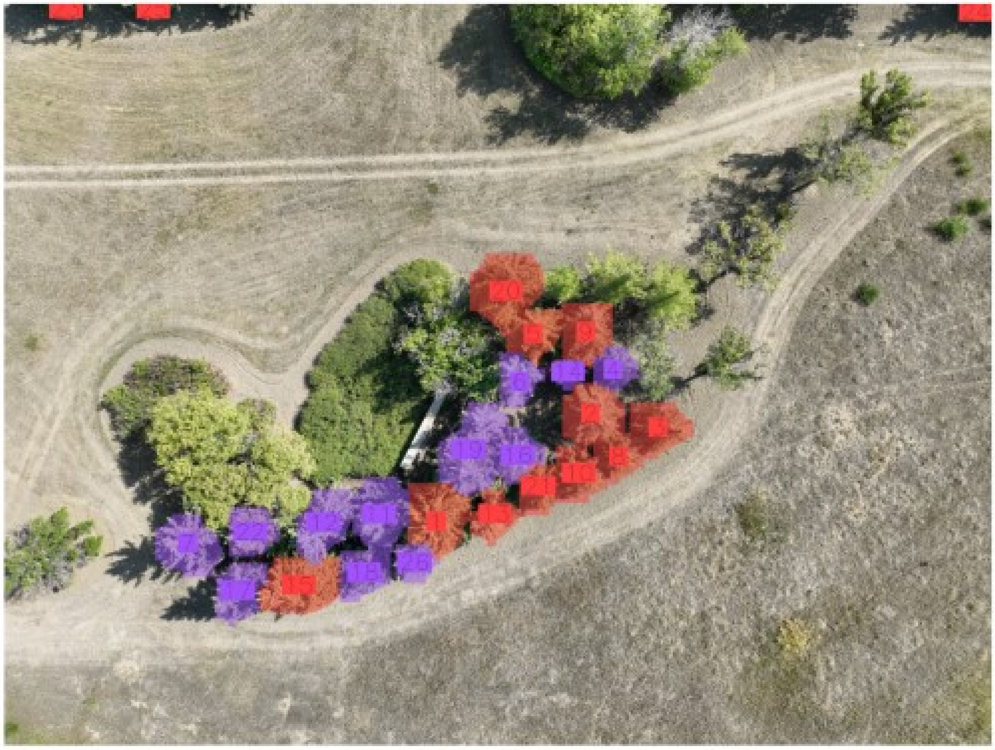


**Figure 37**: YOLOv11 instance segmentation example 4 segmented image

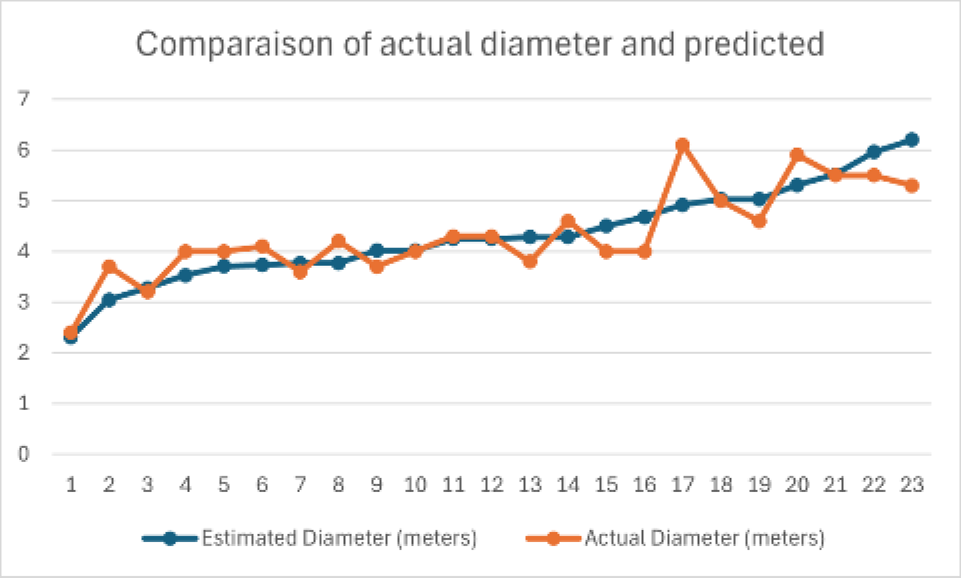
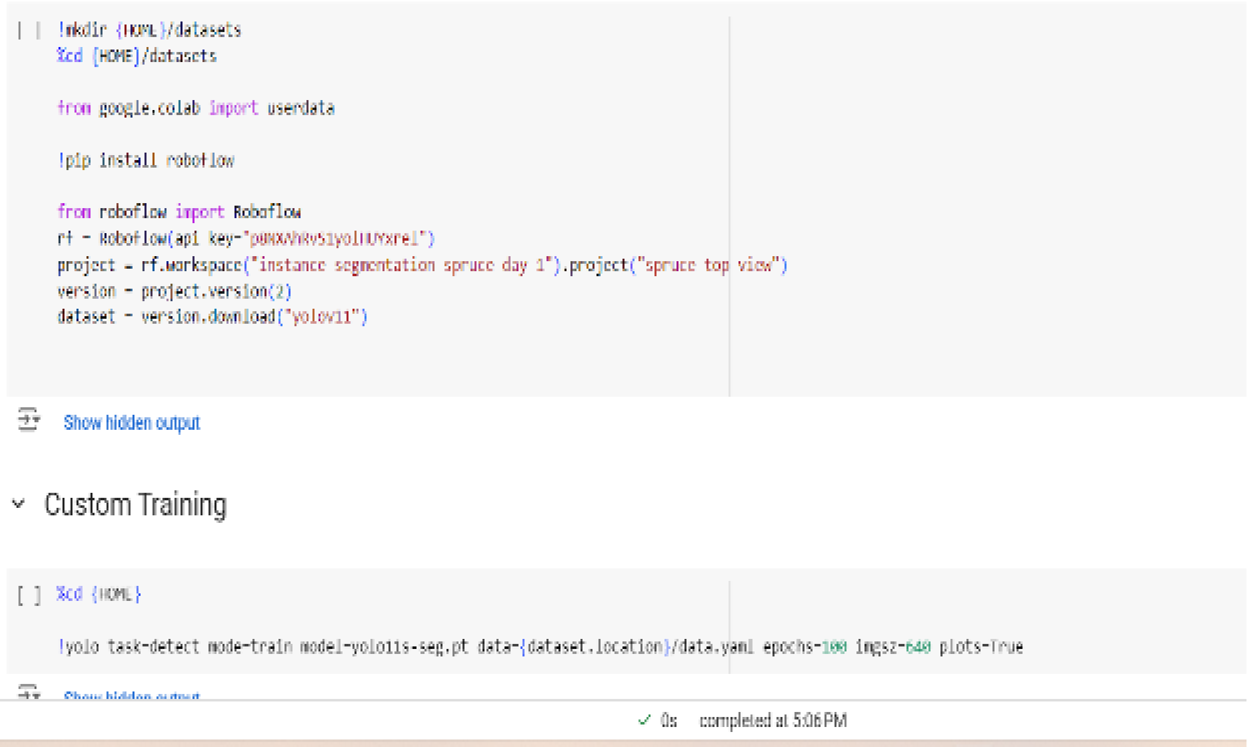


**Figure 38**: YOLOv11 instance segmentation example 4 comparison

## Appendix B: Snipped of code and Excel tables

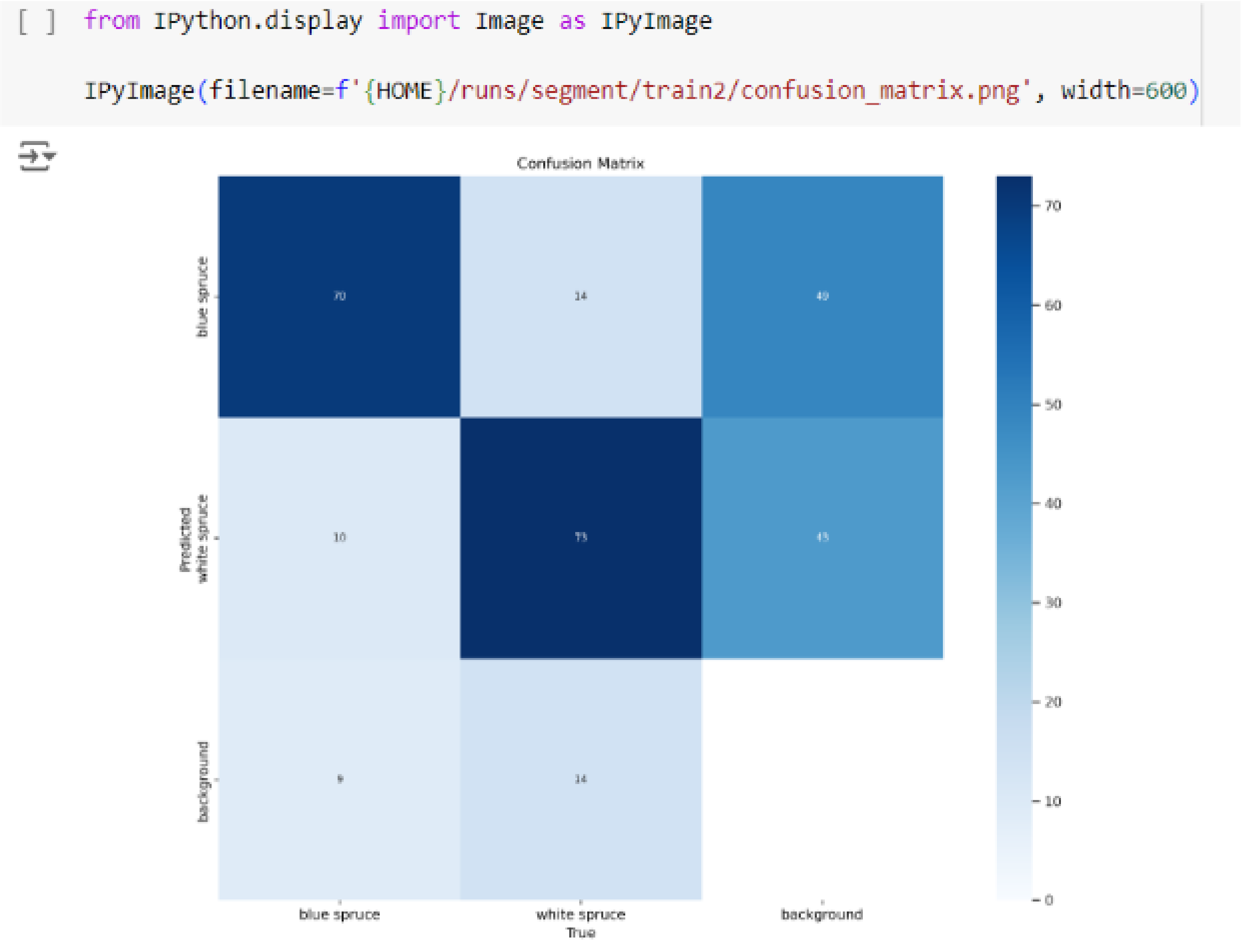


```
!mkdir {HOME}/datasets
%cd {HOME}/datasets

from google.colab import userdata

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="pUNXVhRv51yolHUYxrel")
project = rf.workspace("instance segmentation spruce day 1").project("spruce top view")
version = project.version(2)
dataset = version.download("yolov11")
```

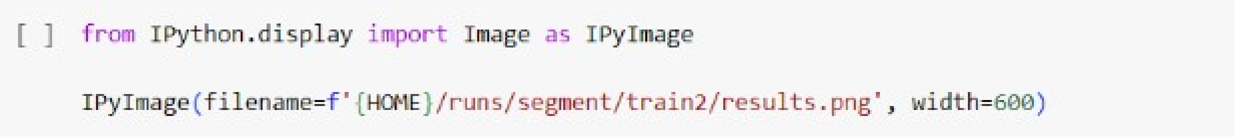
Custom Training

```
%cd {HOME}

!yolo task=detect mode=train model=yolo11s-seg.pt data={dataset.location}/data.yaml epochs=100 imgsz=640 plots=True
```

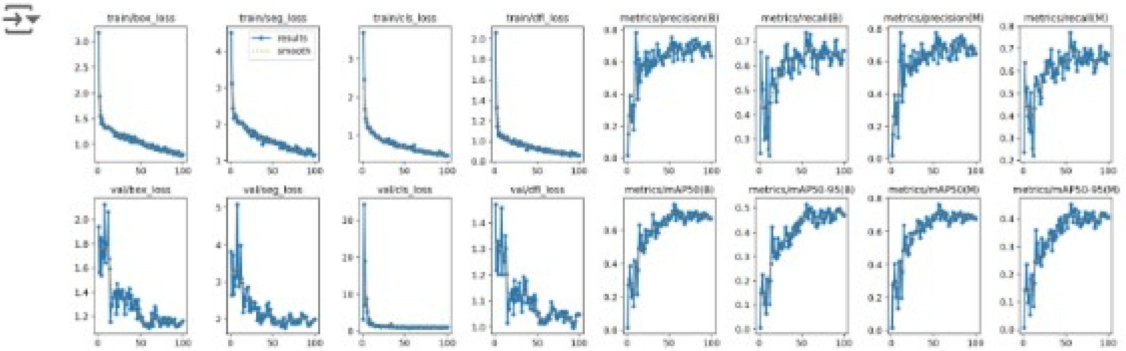
**Figure 39:** YOLOv11 Training

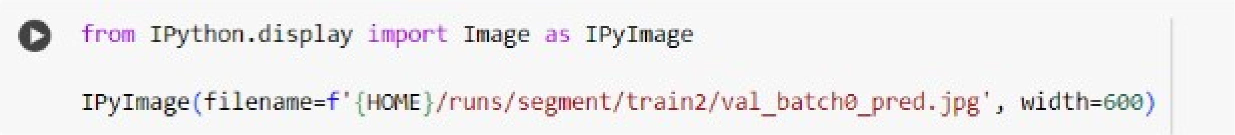


```
[ ]  from IPython.display import Image as IPyImage

     IPyImage(filename=f'{HOME}/runs/segment/train2/confusion_matrix.png', width=600)
```

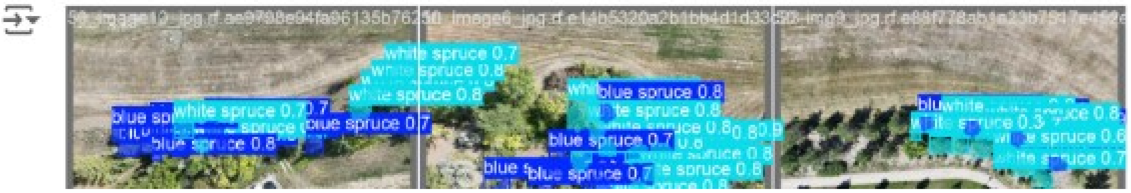


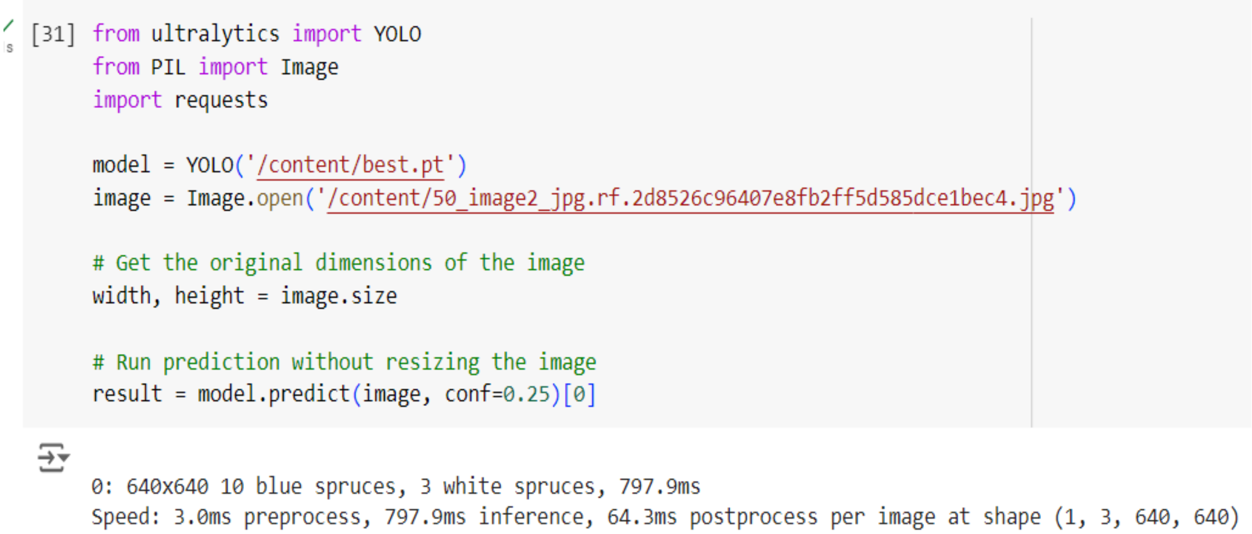
```
[ ]  from IPython.display import Image as IPyImage

     IPyImage(filename=f'{HOME}/runs/segment/train2/results.png', width=600)
```





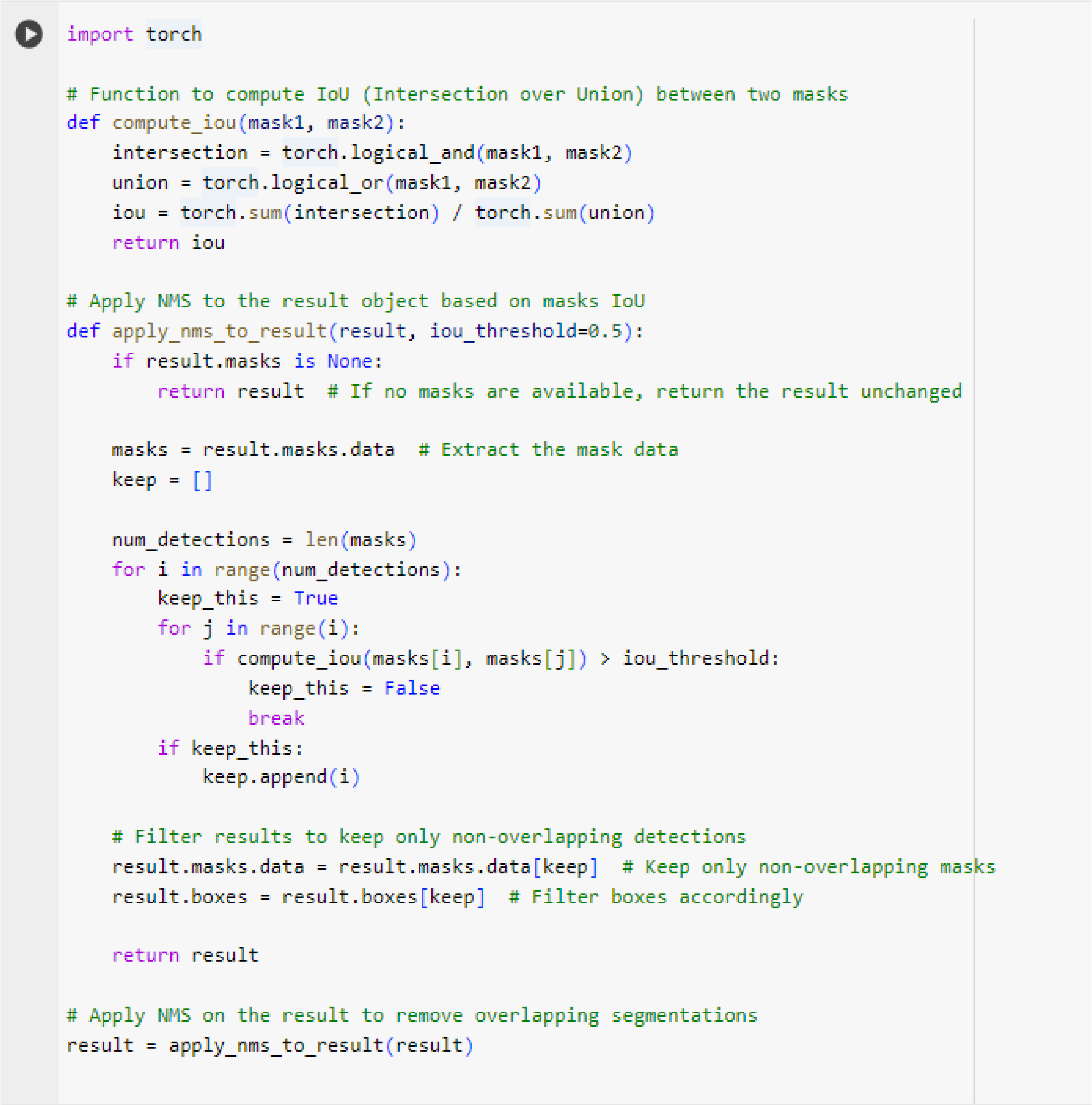
```
from IPython.display import Image as IPyImage

IPyImage(filename=f'{HOME}/runs/segment/train2/val_batch0_pred.jpg', width=600)
```



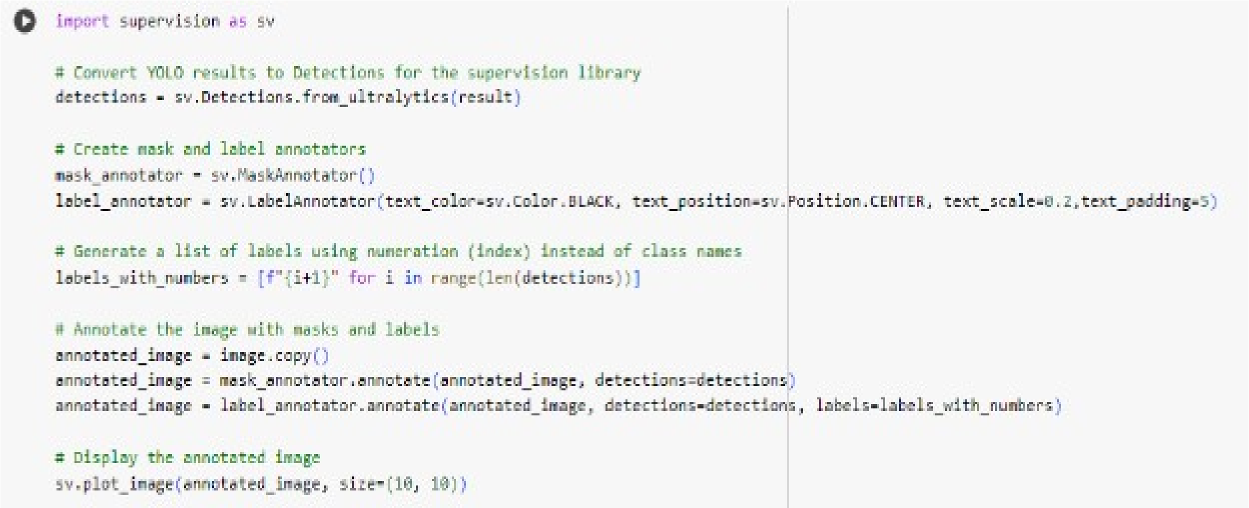


```
[31] from ultralytics import YOLO
     from PIL import Image
     import requests

     model = YOLO('/content/best.pt')
     image = Image.open('/content/50_image2_jpg.rf.2d8526c96407e8fb2ff5d585dce1bec4.jpg')

     # Get the original dimensions of the image
     width, height = image.size

     # Run prediction without resizing the image
     result = model.predict(image, conf=0.25)[0]
```

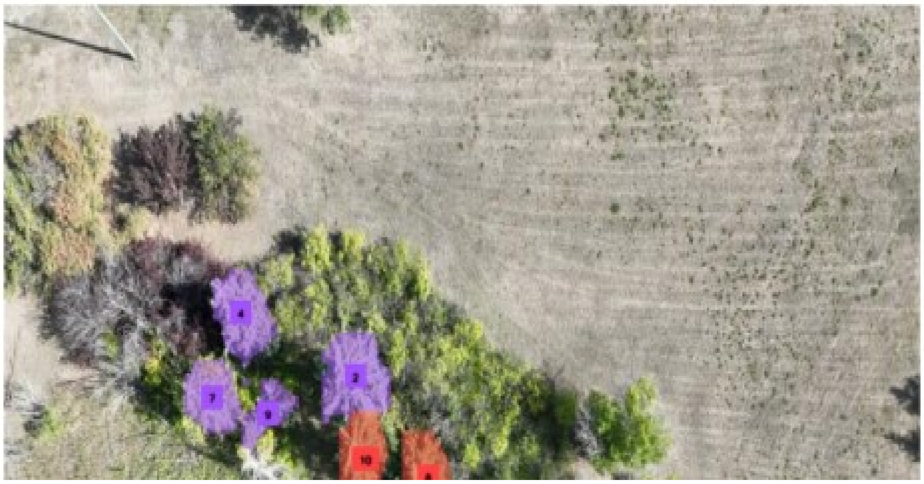
```
0: 640x640 10 blue spruces, 3 white spruces, 797.9ms
Speed: 3.0ms preprocess, 797.9ms inference, 64.3ms postprocess per image at shape (1, 3, 640, 640)
```

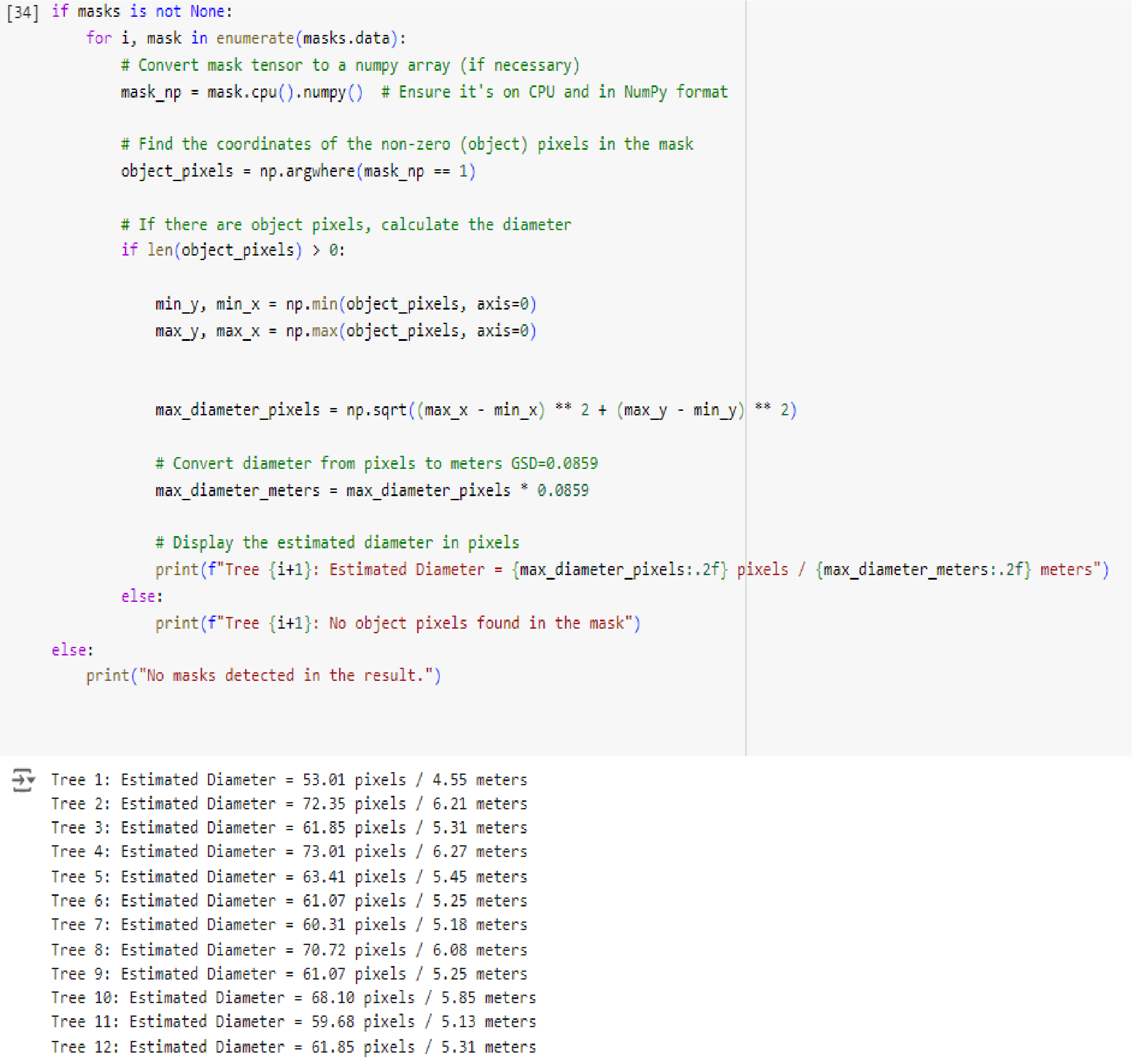


```
import torch

# Function to compute IoU (Intersection over Union) between two masks
def compute_iou(mask1, mask2):
    intersection = torch.logical_and(mask1, mask2)
    union = torch.logical_or(mask1, mask2)
    iou = torch.sum(intersection) / torch.sum(union)
    return iou

# Apply NMS to the result object based on masks IoU
def apply_nms_to_result(result, iou_threshold=0.5):
    if result.masks is None:
        return result  # If no masks are available, return the result unchanged

    masks = result.masks.data  # Extract the mask data
    keep = []

    num_detections = len(masks)
    for i in range(num_detections):
        keep_this = True
        for j in range(i):
            if compute_iou(masks[i], masks[j]) > iou_threshold:
                keep_this = False
                break
        if keep_this:
            keep.append(i)

    # Filter results to keep only non-overlapping detections
    result.masks.data = result.masks.data[keep]  # Keep only non-overlapping masks
    result.boxes = result.boxes[keep]  # Filter boxes accordingly

    return result

# Apply NMS on the result to remove overlapping segmentations
result = apply_nms_to_result(result)
```



```
import supervision as sv

# Convert YOLO results to Detections for the supervision library
detections = sv.Detections.from_ultralytics(result)

# Create mask and label annotators
mask_annotator = sv.MaskAnnotator()
label_annotator = sv.LabelAnnotator(text_color=sv.Color.BLACK, text_position=sv.Position.CENTER, text_scale=0.2,text_padding=5)

# Generate a list of labels using numeration (index) instead of class names
labels_with_numbers = [f"{i+1}" for i in range(len(detections))]

# Annotate the image with masks and labels
annotated_image = image.copy()
annotated_image = mask_annotator.annotate(annotated_image, detections=detections)
annotated_image = label_annotator.annotate(annotated_image, detections=detections, labels=labels_with_numbers)

# Display the annotated image
sv.plot_image(annotated_image, size=(10, 10))
```

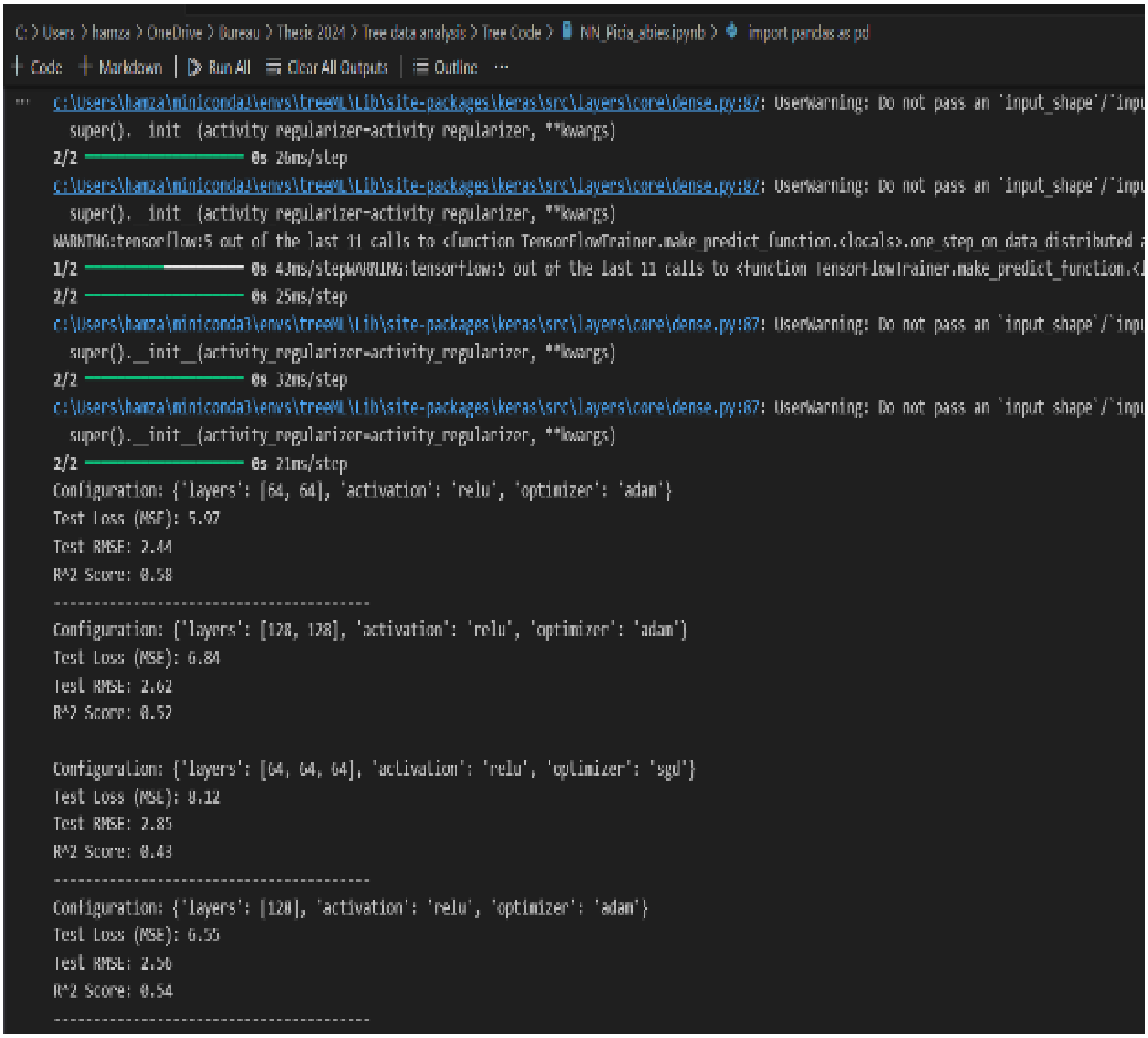




```
[34] if masks is not None:
         for i, mask in enumerate(masks.data):
             # Convert mask tensor to a numpy array (if necessary)
             mask_np = mask.cpu().numpy()  # Ensure it's on CPU and in NumPy format

             # Find the coordinates of the non-zero (object) pixels in the mask
             object_pixels = np.argwhere(mask_np == 1)

             # If there are object pixels, calculate the diameter
             if len(object_pixels) > 0:

                 min_y, min_x = np.min(object_pixels, axis=0)
                 max_y, max_x = np.max(object_pixels, axis=0)


                 max_diameter_pixels = np.sqrt((max_x - min_x) ** 2 + (max_y - min_y) ** 2)

                 # Convert diameter from pixels to meters GSD=0.0859
                 max_diameter_meters = max_diameter_pixels * 0.0859

                 # Display the estimated diameter in pixels
                 print(f"Tree {i+1}: Estimated Diameter = {max_diameter_pixels:.2f} pixels / {max_diameter_meters:.2f} meters")
             else:
                 print(f"Tree {i+1}: No object pixels found in the mask")
     else:
         print("No masks detected in the result.")
```

```
Tree 1: Estimated Diameter = 53.01 pixels / 4.55 meters
Tree 2: Estimated Diameter = 72.35 pixels / 6.21 meters
Tree 3: Estimated Diameter = 61.85 pixels / 5.31 meters
Tree 4: Estimated Diameter = 73.01 pixels / 6.27 meters
Tree 5: Estimated Diameter = 63.41 pixels / 5.45 meters
Tree 6: Estimated Diameter = 61.07 pixels / 5.25 meters
Tree 7: Estimated Diameter = 60.31 pixels / 5.18 meters
Tree 8: Estimated Diameter = 70.72 pixels / 6.08 meters
Tree 9: Estimated Diameter = 61.07 pixels / 5.25 meters
Tree 10: Estimated Diameter = 68.10 pixels / 5.85 meters
Tree 11: Estimated Diameter = 59.68 pixels / 5.13 meters
Tree 12: Estimated Diameter = 61.85 pixels / 5.31 meters
```

**Figure 40**: YOLOv11 TCD calculation


C: > Users > hamza > OneDrive > Bureau > Thesis 2024 > Tree data analysis > Tree Code > NN_Picia_abies.ipynb > import pandas as pd
+ Code
+ Markdown
Run All
Clear All Outputs
Outline
···

```
c:\Users\hamza\miniconda3\envs\treeML\Lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
c:\Users\hamza\miniconda3\envs\treeML\Lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
WARNING:tensorflow:5 out of the last 11 calls to <function TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed a
1/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/stepWARNING:tensorflow:5 out of the last 11 calls to <function TensorFlowTrainer.make_predict_function.<l
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
c:\Users\hamza\miniconda3\envs\treeML\Lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
c:\Users\hamza\miniconda3\envs\treeML\Lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
Configuration: {'layers': [64, 64], 'activation': 'relu', 'optimizer': 'adam'}
Test Loss (MSE): 5.97
Test RMSE: 2.44
R^2 Score: 0.58
----------------------------------------
Configuration: {'layers': [128, 128], 'activation': 'relu', 'optimizer': 'adam'}
Test Loss (MSE): 6.84
Test RMSE: 2.62
R^2 Score: 0.52

Configuration: {'layers': [64, 64, 64], 'activation': 'relu', 'optimizer': 'sgd'}
Test Loss (MSE): 8.12
Test RMSE: 2.85
R^2 Score: 0.43
----------------------------------------
Configuration: {'layers': [128], 'activation': 'relu', 'optimizer': 'adam'}
Test Loss (MSE): 6.55
Test RMSE: 2.56
R^2 Score: 0.54
----------------------------------------
```

**Figure 41:** Result of NN model to detect D