Automated Aerial Detection of Spruce Tree Crowns Through YOLOv5 and Watershed Segmentation

A Thesis

Submitted to the Faculty of Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science

in

Industrial Systems Engineering

University of Regina

by

Mahdi Mohemi Moshkenani

Regina, Saskatchewan

January 2025

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

Mahdi Mohemi Moshkenani, candidate for the degree of **Master of Applied Science in Industrial Systems Engineering,** has presented a thesis titled, ***Automated aerial detection of spruce tree crowns through YOLOv5 and watershed segmentation,*** in an oral examination held on **December 19, 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. Farshid Torabi, Energy & Process Systems Engineering
Supervisor(s):	Dr. Wei Peng, Engineering General
Committee Member:	Dr. Amr Henni, Industrial & Process Systems Engineering
Committee Member:	Dr. Golam Kabir, Industrial Systems Engineering
Chair of Defense:	Dr. Shakil M. Khan, Department of Computer Science

Abstract

The spruce tree, a key species in Canada, is crucial to industries like timber and pulp. Effective management of these resources relies on monitoring forest parameters to ensure long-term forest health and productivity. Tree crown dimensions contribute valuable insights into these parameters. This study investigates an automated approach for detecting and measuring spruce tree crowns using the YOLOv5 machine learning model combined with watershed segmentation. The method enhances the accuracy of crown measurements from aerial drone images. Over 2,000 spruce trees in a limited forested area of Saskatchewan, Canada, are analysed using top-view images captured by a DJI Mavic 3 Classic drone, which is sufficient for this project as the model trained well. The YOLOv5 model is initially employed to detect trees, following by watershed segmentation to refine the tree crown boundaries. In the forest measuring area, some regions contain closely spaced trees with overlapping crowns that cause challenges for accurate recognition of tree boundaries. Although feeding watershed segmentation with YOLO-detected individual trees addresses this issue, tuning the IoU threshold in the NMS stage, applying data augmentation, and utilizing high-resolution images further enhance detection accuracy.

One issue in predicting tree crowns is the underestimation of diameters, often resulting from systematic errors in image capture, measurement methods, environmental conditions, and limitations in image resolution. To address this issue, a linear regression model is applied to adjust the predicted crown diameters, aligning them more closely with the actual field measurements. This approach demonstrates an acceptable accuracy of 89.1% compared to prior research and existing methodologies, which reported accuracies ranging from 67.72% to 95.4%, particularly in complex forest environments.

Although the primary focus of this research is on measuring tree crowns, the findings have broader implications for forestry management activities, such as biomass estimation and forest health monitoring. Future research could explore further improvements to the model for real-time forest management applications, as well as expand its use to detect and measure other tree species in mixed forests and agriculture plants.

Acknowledgements

I want to take this opportunity to sincerely thank everyone who helped me complete this thesis.

First, I am deeply grateful to my supervisor, Dr. Wei Peng, for his constant support, guidance, and encouragement during this project. His knowledge, helpful feedback, and dedication are important in shaping this work.

I also want to thank the members of my thesis committee for their valuable suggestions, advice, and mentorship. Their help was key in improving the method and quality of this thesis.

I would like to thank Mr. H. Zouaghi for his significant help in creating the dataset. His assistance was crucial for the dataset preparation.

Additionally, I would like to express my gratitude to the University of Regina for providing me with the opportunity to undertake this research and supporting me throughout this journey.

Finally, I would like to end by thanking the late Dr. Rene Mayorga, whose guidance and contributions still inspire me.

Dedication

For my loving wife, who was always by my side, **Masoumeh**

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Acc	Accuracy
AD	Actual Dimension
Att-Mask	Attention based Mask
CHM	Canopy Height Model
Conv	Convolution
CSF	Cloth Simulation Filtering
DBH	Diameter at Breast Height
FN	False Negative
FOV	Field of View
FP	False Positive
GFLOP	Giga Floating Point Operations per Second
GPS	Global Positioning System
GSD	Ground Sample Distance
HSV	Hue, Saturation, Value
HFOV	Horizontal Field of View
IoU	Intersection over Union
ITC	Individual Tree Crown
LAI	Leaf Area Index
LiDAR	Light Detection and Ranging
LM	Local Maximum
MAE	Mean Absolute Error
MCWST	Marker-Controlled Watershed Transformation
MD	Measured Dimension
ML	Machine Learning
MPE	Mean Percentage Error
MSE	Mean Squared Error
NMS	Non-Maximum Suppression
PANet	Path Aggregation Network
R-CNN	Region-based Convolutional Neural Network

RGB	Red, Green, Blue
RMSE	Root Mean Square Error
SGD	Stochastic Gradient Descent
SLAM	Simultaneous Localization and Mapping
TP	True Positive
UAV	Unmanned Aerial Vehicle
WST	Watershed Segmentation Technique
YOLOv5	You Only Look Once version 5

List of Symbols

α	Species-specific coefficient
B	Estimated biomass
n	Number of observations
$x1, y1$	Coordinates for calculating X distances (Pythagoras' theorem)
$x2, y2$	Coordinates for calculating Y distances (Pythagoras' theorem)
y_i	Actual values
$\hat{y}_i$	Predicted values

Transparency Statement

In preparing this thesis, I used Grammarly to check the grammar and sentence structure of my own writing.

My supervisor and supervisory committee have approved the use of the above technologies for the described purposes. I confirm that no AI technologies other than the one 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.

1. Introduction

1.1. Overview

One of the most significant tree species in Canada is the spruce tree. It provides robust, high-quality timber for construction, pulp and paper production, and furniture making [1]. Spruce trees play a significant role in nature since they help maintain biodiversity and capture carbon, forming the base of the boreal forest's ecosystem [2]. Despite these advantages, forests remain vulnerable to natural hazards such as wildfires, and pest infestations, highlighting the need for proper resource management [3]. Accurate measurement and monitoring of tree growth and biomass, tree species composition, diversity forest density, and canopy cover are examples of key parameters that provide effective management strategies to ensure forest sustainability, biodiversity, and resilience to environmental changes. Measuring tree crown features such as the crown diameter and area is essential because it is closely linked to biomass volume, tree health, and overall forest productivity [4].

New technologies in remote sensing and machine learning let us use aerial images to find and measure tree features automatically, while traditional methods of field measurements are labor-intensive and time-consuming for large areas. The unmanned aerial vehicles (UAV), or drones, that are equipped with advanced cameras and GPS, can efficiently map and monitor large areas to provide valuable data for analysis. Machine learning is applied to identify regions of interest, object detection, and segmentation to extract data from aerial photos [5]. However, achieving high accuracy in image processing remains challenging due to factors like resolution, lighting, and object overlap.

This study uses aerial photos to accurately separate spruce tree crowns with the YOLOv5 object detection model, and the Watershed Segmentation Technique (WST) to conduct image processing. The methodology introduces significant contributions and innovations, advancing the field of automated tree measurement in forestry. One of the key innovations is the capability to detect spruce trees in mixed-species environments. Unlike other methods that focus exclusively on single-species forests, this study addresses the challenge of identifying spruce trees among different species and provides accurate crown detection and diameter measurement for the desired species. This expands the applicability of the method to diverse and complex forest conditions.

Another significant contribution is the integration of YOLOv5 and watershed segmentation. By combining object detection and image segmentation, the proposed method tackles common challenges in crown detection. Previous research using watershed segmentation often faced issues where background objects and foreign elements were mistakenly recognized as tree crowns. This project eliminates such errors by first detecting only spruce trees using YOLOv5. This step guarantees that only relevant objects proceed to the segmentation stage, where watershed segmentation is used to precisely measure the crown dimensions. This streamlined approach reduces the need for additional filtering or manual corrections, making the process more efficient and reliable.

The study also introduces a comprehensive dataset of spruce trees, supported by precise field measurements of crown diameters and DBH. A regression-based correction mechanism aligns aerial predictions with field measurements, significantly improving the accuracy of diameter estimation. This ensures that the proposed method is not only innovative but also robust and applicable in real-world forestry management scenarios. The findings demonstrate the method’s reliability in complex forest conditions and its potential to improve the accuracy of tree crown measurement.

Finally, the study establishes a strong relationship between crown diameter and Diameter at Breast Height (DBH) through a linear regression model, enabling indirect biomass estimation. By utilizing crown measurements obtained from aerial imagery, this approach reduces the need for extensive ground-based measurements. This finding has practical implications for forestry management, particularly in resource-limited settings where field data collection is challenging.

Notably, the results from this methodology demonstrate a higher accuracy than those achieved with top-view aerial drone imagery techniques referenced in the literature, as well as those obtained using LiDAR for similar applications. The innovations introduced in this study provide a scalable and efficient solution for forest monitoring and management, with potential extensions to mixed forests and other tree species.

1.2. Research Objectives

The aim of this research is to develop an automated method that uses aerial data to predict spruce tree crown diameters. The study uses data from over 2,000 spruce trees in Saskatchewan, Canada, where drones captured top-view images at two different heights, 50m and 70m over the ground.

I will modify a YOLOv5 model to detect spruce trees among other tree species in aerial photos, marking their locations. Then, watershed segmentation separates spruce tree crowns and measures their diameter and area based on pixel value. The accuracy of present model is evaluated with field measurements, and finally, linear regression is used to find the correlation between crown diameter and Diameter at Breast Height (DBH), which enables biomass estimation.

This research aims to answer some key questions; How well can the YOLOv5 model detect spruce trees among other species in aerial images? Can watershed segmentation accurately measure tree crowns in crowded forest areas? How closely does the correction technique match the predicted crown diameters with real field measurements? How is the measured crown diameter linked to DBH and biomass estimates?

1.3. Structure of this Thesis

The rest of this thesis is organized into four chapters. Chapter 2 reviews the existing literature on tree crown measurement methods, the use of machine learning in forestry, and applied tools. It points out gaps in previous methods for predicting crown diameters, which this research aims to address. Chapter 3 explains the methodology, covering data collection, setting up the YOLOv5 model, using watershed segmentation for crown detection, and applying linear regression for corrections. It also describes how the accuracy of the model is tested. Chapter 4 presents the results by showing the detection and segmentation accuracy of the proposed method. Furthermore, the strengths and weaknesses of the approach, and its implication for forest management are discussed. Chapter 5 summarizes the main findings and suggests potential future improvements and applications for different tree species and regions.

2.Literature Review

In this chapter, the segmentation of individual tree crowns (ITCs) is divided into two categories based on the dataset used, those that rely on top-view aerial drone imagery and those utilizing LiDAR data. The choice of dataset and segmentation method has a significant impact on both the accuracy and effectiveness of the results. LiDAR-based studies utilize high-resolution point cloud data to capture detailed canopy structures, though often at a high cost due to the equipment required. Alternatively, drone-based methods provide a more accessible option, particularly with developments in image processing techniques. The literature review examines six related studies, focusing on the methodologies and accuracy metrics achieved in research.

2.1. Top-View Drone Imagery as a Data Source

Study 1:

Fan et al. (2024) [6] conducted a study concentrating on individual tree crown (ITC) segmentation to enhance the accuracy of tree-level monitoring and management in forest environments. Their primary aim was to address the complexities posed by crown heterogeneity, overlapping crowns, and varying image quality, which traditionally hinder the segmentation of individual trees using remote sensing data. To achieve this, the authors developed a hybrid approach, combining classical and deep learning methods to improve segmentation accuracy while reducing computational costs.

The proposed method included a three-step approach. First, they used a Faster R-CNN model to detect initial bounding boxes for tree locations, based on a custom COCO dataset, which is a large-scale object detection dataset containing diverse images with annotated objects for training and testing machine learning models. COCO, short for Common Objects in

Context, includes not only object annotations but also contextual information about their surroundings, making it ideal for tasks requiring object localization and classification. Its extensive variety of categories and real-world scenarios enhances the robustness and generalizability of machine learning models trained on it. This step established an initial detection that helped reduce the over-segmentation often caused by traditional seed point methods. Following this, an MCWST (Marker Controlled Watershed Transformation) approach was applied, which provided individual tree segmentation based on marker-controlled watershed transformation techniques. Finally, Mask R-CNN was implemented for pixel-level instance segmentation, enhancing segmentation detail by refining the initial bounding boxes.

The authors reported significant improvements in model performance through quantitative evaluations, noting that Mask R-CNN achieved a precision of 67.72% and a recall of 70.14%. These metrics represented a marked improvement over the MCWST approach, which exhibited lower segmentation accuracy. For example, MCWST achieved a precision of 52.9% and recall of 34.2%, illustrating the enhanced effectiveness of Mask R-CNN in segmenting individual crowns in dense forest environments. In addition, the F1-score for the Mask R-CNN model was reported at 0.67, indicating a balanced performance between precision and recall.

An important finding was the method’s adaptability in segmenting tree crowns across different forest types. The authors applied their technique to diverse datasets from regions with coniferous and mixed forest areas, achieving high segmentation accuracy. In one segment, they reported segmenting up to 15477 tree crowns in coniferous-dominant areas, showing the model's robustness even in densely forested environments. These findings underscore the hybrid approach's applicability across various forest conditions and its potential for large-scale forest monitoring.

In conclusion, they demonstrated that integrating Mask R-CNN with traditional watershed approaches can substantially improve ITC segmentation accuracy, particularly in challenging conditions with overlapping and heterogeneous crowns. Their work contributes to advancements in tree-level monitoring tools, enabling better forest management and environmental assessments.

Study 2:

Chen et al. (2024) [7] aimed to improve the precision of individual tree crown segmentation using an optimized model, Att-Mask R-CNN, which incorporates a fused attention mechanism. This model employs UAV based remote sensing images to identify and segment tree canopies, specifically targeting six species in Xiong County, China. The study was motivated by the need for more accurate and efficient methods of tree canopy measurement, which is essential for understanding forest stand structure and facilitating forest management.

The proposed method began with the collection of 1,200 images using a DJI Phantom 3 drone, covering six distinct tree species. These images were then annotated and used to train the Att-Mask R-CNN model, which was an enhancement over the original Mask R-CNN and MS R-CNN models. The primary innovation in Att-Mask R-CNN is the addition of a fused attention mechanism, combining channel and spatial attention. This feature enables the model to better capture complex forest details, allowing it to differentiate tree crowns more precisely.

In terms of results, the optimized Att-Mask R-CNN achieved substantial improvements in segmentation accuracy. Specifically, it obtained a mean average precision (mAP) of 65.29% and a mean intersection over union (mIoU) of 80.44% across all species. When compared with other models, Att-Mask R-CNN outperformed both Mask R-CNN (mAP of 62.41%, mIoU of 74.36%) and MS R-CNN (mAP of 64.62%, mIoU of 77.84%), confirming its enhanced accuracy due to the attention mechanism. Additionally, individual tree species results showed that Sabina chinensis achieved the highest segmentation accuracy with an F1 score of 95.83%, while Salix matsudana had the lowest at 74.32% due to crown overlap and shading issues that challenged the segmentation process.

A notable finding from the study was the ability of the Att-Mask R-CNN model to provide accurate estimates of the tree crown's vertical projected area. Using a mask pixel-based calculation, the model achieved an R^2 value of 85% and a relative root mean square error (rRMSE) of 12.81% in estimating the crown area. These metrics highlight the model's capacity to generate reliable canopy projections, which can be essential in forest resource assessments and ecological studies.

2.2. LiDAR Data as a Resource for Analysis

Study 3:

In their study, Liu et al. (2024) [8] aimed to improve the accuracy of individual tree segmentation from airborne LiDAR point clouds, particularly addressing challenges related to over-segmentation and under-segmentation. Their primary goal was to combine the strengths of traditional segmentation techniques with more advanced clustering methods to achieve higher precision.

Liu et al. applied marker-controlled watershed segmentation, which is a well-known method for separating objects in an image based on intensity gradients. This technique allows for the initial identification of individual tree clusters. However, to refine these clusters, particularly in cases where trees are closely packed or under-segmented, the authors introduce spectral clustering.

They tested their method on six different forest plots, including both coniferous and broad-leafed forest types. The results show outperform in comparison with traditional Canopy Height Model (CHM) segmentation techniques, since they achieved an average recall of 0.854 (the proportion of correctly detected tree crowns among all actual crowns), a precision of 0.937 (the proportion of correctly detected crowns among all detections made), and an F1-score of 0.892 (the harmonic means of precision and recall, balancing both metrics). These metrics indicate that their method not only detects most trees correctly but also minimizes false detections, making it highly reliable for individual tree segmentation in complex forest environments.

A key finding of the study is the ability of their method to handle diverse forest structures. Traditional methods, such as CHM approaches, often struggle in forests with mixed species or dense canopies. However, the marker-controlled watershed combined with spectral clustering performed well across different forest types and can achieve high segmentation accuracy even in forests with overlapping crowns and varying crown shapes.

Another important aspect of the study is its reliance on airborne LiDAR data. LiDAR offers several advantages over traditional aerial imagery, as it provides detailed three-dimensional information about tree height, crown width, and overall forest structure.

They also compared their method with other segmentation approaches. They found that CHM methods work well for simple tree canopies. However, these methods become less accurate in more complex areas, like forests with dense or overlapping tree crowns. In contrast, their method maintained high accuracy across all forest types. They also highlighted that point-based methods, like K-means clustering, have limitations. These methods often miss the small details of individual tree shapes. By combining the best aspects of CHM and point-based methods, the proposed method provides a more robust solution to the segmentation problem [8].

Study 4:

Fu et al. (2024) [9] conducted a study to enhance individual tree segmentation using a hybrid approach combining UAV-LiDAR data with multiscale adaptive filtering and clustering techniques. The authors aimed to address the challenges of over-segmentation and under-segmentation in forests, especially when dealing with varying tree densities and species.

The proposed method first employed a multiscale adaptive local maximum (LM) filter for crown detection, which adapts box sizes based on the crown size and type of tree (coniferous or broad-leaved). This filter dynamically adjusted to the different characteristics of the forest plots, resulting in more accurate identification of crowns, even in high-density areas.

They introduced a fuzzy C-means clustering method to refine the segmentation, using supervoxel-weighted clustering to separate tree crowns, particularly where crown boundaries overlap. This approach resulted in high segmentation precision, as seen in the various metrics reported from the experiments. Specifically, Fu et al. achieved a precision of 88.27% and a recall of 86.31% during crown detection.

In addition to the precision and recall, the study reported an overall F1-Score of 0.892 across multiple datasets. The hybrid segmentation method was applied to synthetic datasets, which varied in density from 233 to 811 stems per hectare and had crown areas ranging from 3.40 to 112.45 m^2 for coniferous plots. These results demonstrate the flexibility of the method in managing both sparse and dense forest environments. Furthermore, the approach was tested on a real UAV-LiDAR dataset, and it successfully segmented individual tree crowns.

The study's findings suggested that this hybrid approach can significantly improve forest inventory practices by offering more precise individual tree segmentation, which is critical for estimating tree biomass, carbon stock, and monitoring forest health [9].

Study 5:

Yu et al. (2024) [10] focused on improving individual tree segmentation using UAV-LiDAR data through an adaptive crown-shaped algorithm. The goal of this study was to accurately segment trees by detecting seed points and applying a region growing method for individual tree segmentation. The key challenge was reducing errors associated with over-segmentation and under-segmentation in complex forest environments.

The method introduced in this study involved a multi-step process, beginning with the extraction of crown surface points. They divided point clouds into blocks along the X-axis and projected them onto different planes to isolate the crown boundaries. Afterward, they applied an adaptive algorithm to detect the seed points, representing tree crown. This approach was designed to improve segmentation accuracy by focusing on the real tree crown shapes, avoiding issues caused by fixed window sizes in traditional methods.

In their results, Yu et al. (2024) demonstrated significant improvements in seed point detection, achieving a precision of 95.9%, recall of 91.6%, and an overall F1-score of 0.94. These metrics show that the method was highly effective in detecting individual trees across different forest types. Additionally, the accuracy of tree segmentation varied depending on forest type and topography. For example, in a planted coniferous forest, the accuracy reached

87.7%, while in mixed and broadleaf forests, the accuracy was lower, with 73.2% and 70.5% respectively.

The key findings from this study highlight the algorithm's ability to reduce over segmentation and under segmentation errors, particularly in natural forest environments where tree crowns often overlap. The adaptive approach allowed for more flexible seed point detection, adapting to different forest structures and improving the precision of tree delineation compared to traditional CHM-based methods.

In conclusion, the study by Yu et al. (2024) provides a robust method for improving individual tree segmentation using UAV-LiDAR data. Their adaptive crown shaped algorithm showed superior performance across a range of forest types, offering valuable insights for future applications in forest management and monitoring [10].

Study 6:

Zu et al. (2024) [11] presented an approach for improving the segmentation of individual rubber tree crowns using UAV-mounted LiDAR data. The study area is in Danzhou City, Hainan Island, China, known for extensive rubber production. The experimental plot, approximately 3 hectares in size, contains four rubber tree clones aged 15 years, namely ReKen 523, ReYan 72059, ReYan 73397, and PR 107. These clones exhibit various biological differences in terms of trunk thickness, crown size, and wind resistance.

The LiDAR data was collected using a DJI Matrice 600 Pro drone equipped with a Velodyne HDL-32E laser scanner. The drone flew at an altitude of 70 meters above ground level, maintaining a flight speed of 5 km/h. The Velodyne HDL-32E laser scanner was setup for the generation of high density point clouds using Simultaneous Localization and Mapping (SLAM) technology, providing rapid scanning and point cloud registration in real time. The system produced an average ground point density of approximately 3500 points per square meter, making it highly effective for detailed spatial mapping.

The LiDAR data facilitated the analysis of tree crowns and trunks within the rubber tree plantation. The trees were planted systematically, with 7 meters between rows (north-south orientation) and 3 meters between lines (east-west orientation). The different rubber tree clones showed varying growth characteristics, such as height, crown width, trunk diameter, and leaf area index (LAI).

They introduced a deep learning network that distinguishes photosynthetic (leaves) from non-photosynthetic parts (branches). The network architecture includes two feature abstraction layers, which perform sampling and feature aggregation, and two feature propagation layers, which restore point clouds to their original resolution for classification. A graph based approach was used for individual tree crown (ITC) segmentation. The process begins by creating an Inverted Canopy Height Model (CHM) from point cloud data, after ground points are removed using Cloth Simulation Filtering (CSF). Rubber trees, which exhibit high canopy density and small angle branching, are segmented based on bottom-up growth patterns to account for crown shapes and complex structures.

The experiment was conducted using the PyTorch deep learning framework on a Windows 10 system. It was executed on a PC equipped with an Intel i7-8550U CPU, 16 GB of RAM, and a NVIDIA GTX 1070 GPU. The hyperparameters chosen included a batch size of 16, a weight decay rate of 0.0001, an input size of 2048, a learning rate of 0.001, and the Adam optimizer. The Adam optimizer is a widely used optimization algorithm in machine learning that combines the benefits of momentum based optimization and adaptive learning rates. It computes individual adaptive learning rates for each parameter using estimates of first and second moments of gradients, enabling faster convergence and better handling of sparse gradients. These hyperparameters were selected to optimize performance while maintaining a reasonable training time. The loss curve converged in 81 epochs.

To evaluate the trained network, results for tree trunk recognition at two levels “instance level” and “point level” are compared. the network demonstrated strong accuracy, with a mean recall of 94.6%, precision of 96.2%, and F1-score of 95.4% at instance level, while the corresponding values for “point level” are 80.6%, 78.0%, and 79.2%, respectively. Their method was compared with marker-controlled watershed algorithm and cluster-based method in all five metrics, and demonstrated its strength over the other algorithms [11].

3.Research Method

This study introduces a method for predicting spruce tree crown diameters from aerial images using a combination of image processing and machine learning techniques. The process begins with the collection of field data in Saskatchewan, Canada, followed using the YOLOv5 model to identify spruce trees. The process involves camera calibration to enhance the accuracy of measurements, thereby improving the precision of the model in estimating final results. After tree detection step, watershed segmentation is used to refine crown shapes, particularly in densely populated areas. Non-Maximum Suppression (NMS) is then tuned to minimize overlapping detections and enhance result accuracy. The segmented pixels are utilized for measuring crown diameters and areas, and linear regression is used to connect crown diameters to DBH measurements. Figure 3-1 presents a comprehensive workflow diagram that summarize the key steps involved in the methodology.

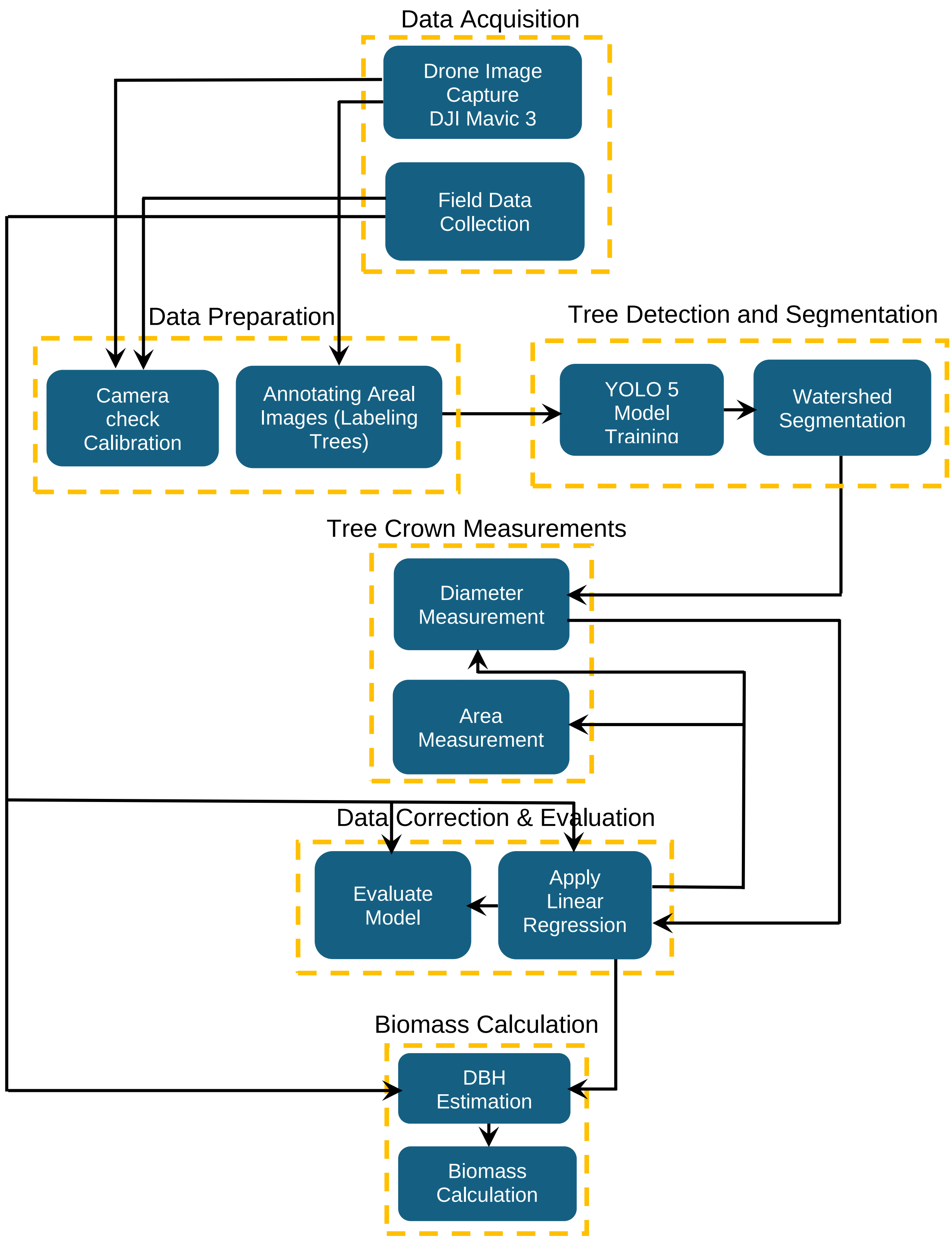

Figure 3-1: Flowchart of methodology

3.1. Equipment Used for Capturing Aerial Images

For this research, a DJI Mavic 3 Classic drone [12] with a 4/3 CMOS Hasselblad camera is used to capture high-resolution top-view images of the area. This camera has a 20MP sensor and can record videos up to 5.1K at 50fps and 4K at 120fps. The camera has a variable aperture from f/2.8 to f/11, allowing better control over depth of field. It also captures 12.8 stops of dynamic range, ensuring clear images even in difficult lighting. The drone uses DJI's O3+ transmission system, providing a 1080p/60fps live feed up to 15 kilometers away, making it great for large-scale aerial surveys. With a 46-minute maximum flight time and omnidirectional obstacle sensing, the Mavic 3 Classic ensures safe and effective data collection in different environments. Figure 3-2 shows images of the drone and its remote controller used in this study [12].

a) b) c)

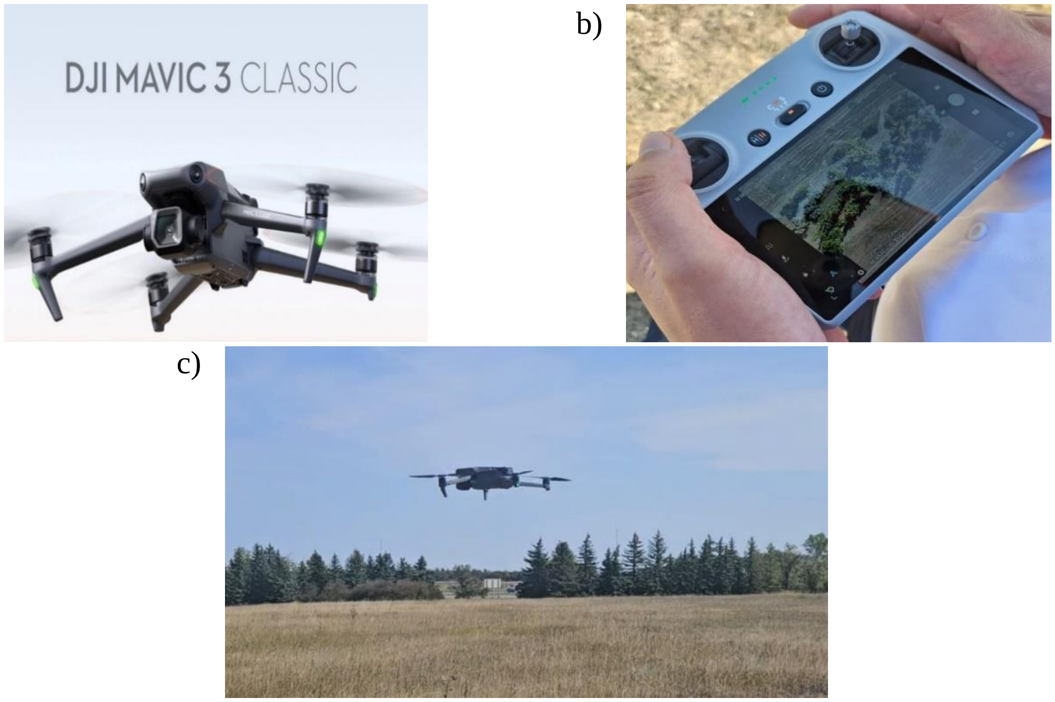

Figure 3-2: Drone and its remote controller a) DJI Mavic 3 Classic drone b) Drone remote controller c) Flying drone in the field of study

3.2. Methods of Measuring Object Dimensions in Images

This research presents two methods for determining the real world value of each pixel using camera parameters. The first method involves calculating the Ground Sample Distance (GSD), which directly links pixel values to actual distances. The second method uses the Field of View (FOV) to determine the image's real world width, allowing for the conversion of pixel values into real world measurements.

In the first method, the real world value for each pixel is calculated using camera settings, such as the sensor width, focal length, and image width in pixel and flight altitude. The GSD helps convert the pixel values into real world measurements [13]. GSD shows the ground distance that matches the space between two pixels in an image. The formula for GSD is [14]:

$$GSD = \frac{SensorWidth \times FlightAltitude}{FocalLength \times ImageWidth} \tag{3.1}$$

A smaller GSD means higher image resolution, with more details in each pixel. A larger GSD means each pixel covers more ground, which lowers the resolution. Once the GSD is calculated, the real world distance for any object in the image can be determined by counting the pixels and multiplying by the GSD.

The second approach is based on using the FOV, which is an important camera parameter that determines how much of the scene can be captured by the camera [15]. It is essential to clarify that the Field of View (FOV) specified by the manufacturer represents the total angle covered by the camera, which is measured across the diagonal of the image. However, in this study, FOV refers specifically to the horizontal field of view (HFOV) of the image. Therefore, all FOV values discussed here relate to the image's width and should not be mistaken for the diagonal FOV given by the manufacturer. Note that the horizontal FOV can be derived from the total diagonal FOV since the image aspect ratio (3:4) is known, or alternatively, it can be calculated using the sensor width and the following formula [16]:

$$FOV = 2 \times arctan\left(\frac{SensorWidth}{2 \times FocalLength}\right) \tag{3.2}$$

Once the FOV is known, it can be used to calculate the width of the image in the real world.

$$Image\ Width\ (real\ world) = 2 \times FlightAltitude \times tan\left(\frac{FOV}{2}\right) \tag{3.3}$$

After finding the image width in real world terms, the real world value for each pixel can be found by dividing the real world image width by the number of pixels along the width of the image. This method is particularly useful for aerial or satellite imagery where need to estimate the extent of ground coverage without directly relying on GSD. It is an alternative to calculating GSD directly. A wider FOV captures a larger part of the scene, which increases the

GSD, meaning each pixel covers more ground. On the other hand, a narrower FOV captures a smaller portion of the scene, resulting in a lower GSD and higher resolution.

To measure the distance between two points in an image, the pixel distance is calculated using the Pythagoras theorem. If the coordinates of the points are ($x1$, $y1$) and ($x2$, $y2$), the distance in pixels is given by the following formula:

$$Distance_{pixels} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.4}$$

This pixel distance is then multiplied by the GSD to convert it into a real world measurement:

$$Distance_{meters} = Distance_{pixels} \times GSD \tag{3.5}$$

When comparing both methods, the GSD connects the pixel resolution to real world distances, while the FOV controls how much of the scene is captured. Accurate GSD and FOV are important for reliable measurements, to this, camera check calibration plays a significant role.

3.3. Camera Calibration Check Process

Camera calibration check is essential to ensure that object sizes and dimensions in images are measured accurately, since minor imperfections in manufacturing and lens distortion cause inaccuracies. Factors such as temperature variations and camera positioning also introduce slight shifts, leading to distorted object appearances in images. Camera calibration corrects these discrepancies to ensure that pixel measurements correspond accurately to real world dimensions. Performing a camera calibration check establishes whether recalibration is necessary prior to conducting any measurements [17].

In camera calibration check, some reference objects with known dimensions are located in the scene. Two images are captured at two altitudes of 50 m, and 70 m. Then, their real world dimensions are calculated and compared with known dimensions. The result of the comparison represents the accuracy of camera and measuring method. If required, camera parameters, such as Field of View (FOV), sensor width, and focal length are updated.

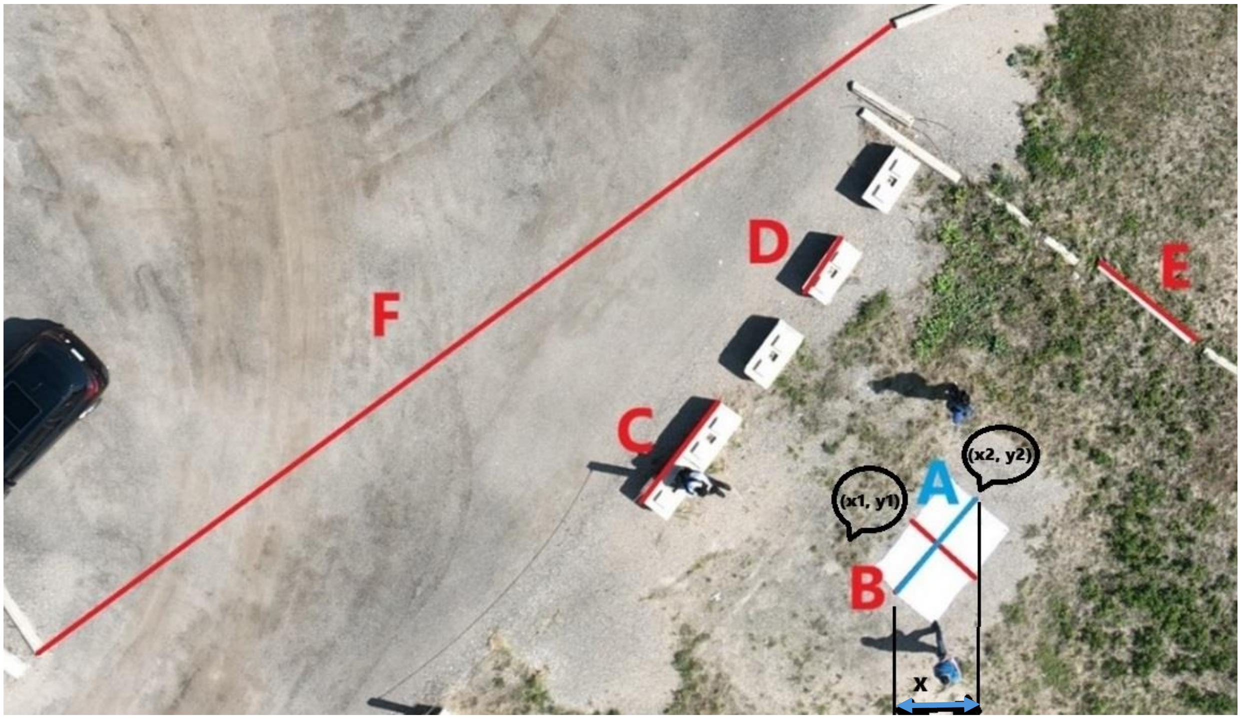

Figure 3-3: Coloured sample distances (Objects)

In Figure 3-3, six sample lines labelled A, B, C, D, E, and F are displayed. The real-world distances of these lines are accurately measured. The pixel coordinates of the object's endpoints, (x_1, y_1) and (x_2, y_2), are identified. Then, the length of X is calculated, and the number of its corresponding pixels are counted. Considering that the number of pixels along the width edge is known, by proportional relationship the image width is calculated.

According to 3.6 and 3.7 formulas, the image width and flight altitude are used to update the camera parameters. The flight altitude is also obtained from the image properties, captured by the drone.

$$FOV = 2 \times arctan\left(\frac{Image\ Width\ (real\ world)}{2 \times Flight\ Altitude}\right) \quad (3.6)$$

This formula isolates the FOV, making it possible to be calculated by using the real world sample size and the altitude at which the image is taken.

$$Sensor\ Width = \frac{GSD \times Focal\ Length \times Image\ Width\ (in\ pixels)}{Flight\ Altitude} \quad (3.7)$$

The 3.8 formula is used to calculate the accuracy:

$$Acc(\%) = 100\% - \left(\frac{|MD - AD|}{AD} \times 100\%\right) \quad (3.8)$$

Where Acc (%) is the accuracy, MD (m) is the measured dimension, and AD (m) is the actual dimension.

3.4. Data Acquisition

3.4.1. Aerial Image

Aerial image capture is conducted at 50 meters and 70 meters altitudes. Shown in Figure 3- 4, the study area is in green spaces of the University of Regina (3737 Wascana Parkway, Regina, SK). The area is chosen to represent diverse forest conditions, including both pure spruce regions and mixed species areas with a wide range of tree crown shapes, sizes, and forest densities.

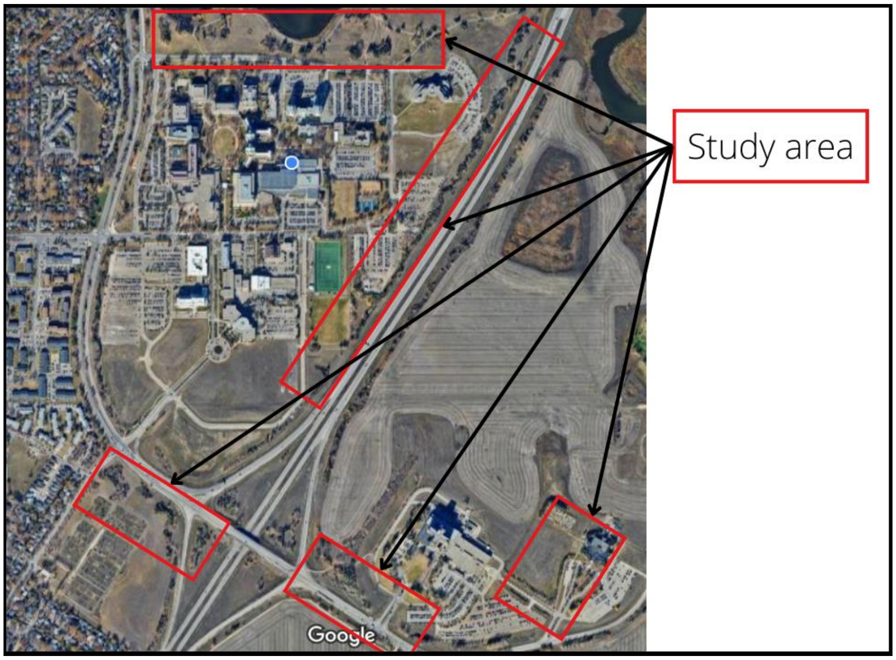

Figure 3-4: Aerial view of study area

To ensure complete coverage of the study area, systematic flight paths are planned to capture across the entire region, regardless of tree density or terrain. Flights are conducted at both altitudes, with images taken randomly at different locations, allowing the model to handle varying image resolutions and scales. This approach ensured that the model would be tested in both dense and sparse tree environments, helping to improve its accuracy across different tree densities and forest conditions. Abnormal or partially infected trees are also included in the study area. It is important to assess the model's ability to handle real world scenarios, where infected trees have irregular shapes or can be partially obscured.

3.4.2. Field Data Collection

In this research, the identification and measurement of spruce trees, specifically white spruce and blue spruce, are the main objectives. Since the study area contained a mix of species, careful identification is needed to make sure non-target species are excluded from the dataset. Each tree is tagged and labelled with a unique identifier to ensure that the field data is accurately matched with the aerial images.

One of the main measurements taken in the field is Diameter at Breast Height (DBH). This measurement, recorded at a standard height of 1.3 meters from the ground, gives important information about a tree's size, health, and potential biomass [18]. A specialized tape is used to measure DBH, ensuring that the tape is placed perpendicular to the trunk to avoid errors. The DBH data is integrated with aerial imagery to explore any correlations between crown diameter and tree size, aiding in the development of the machine learning model.

Haglof Vertex V Hypsometer [13] is used to measure the crown diameter. This ultrasonic tool provides accurate measurements of the widest span of each tree's crown, with a precision of ±1 cm. The device works by sending out ultrasonic pulses and calculating the horizontal distance between two points on the tree crown. It is especially helpful in dense forest areas where visual measurements are difficult due to obstructions. Figure 3-5 shows T4 Transponder and Haglof Vertex V Hypsometer.

a) b)

Figure 3-5: a) T4 Transponder and Haglof Vertex V Hypsometer [13] b) Sample data on the Haglof Vertex V Hypsometer

In the current study, two diameters are measured for every tree crown. First, the farthest points on the outer edge of the crown are identified, and the distance between them is measured as the primary crown diameter [19]. This captures the maximum horizontal spread of the crown. To account for any irregularities in the crown shape, a second diameter is measured by drawing a line perpendicular to the first one, intersecting the crown edge at two points. The distance between these points is recorded as the second diameter. Measuring both diameters provided a more realistic picture of the crown’s size, especially when the crown is not perfectly round [20].

The average of the two measured diameters is then calculated, representing the corresponding diameter. The corresponding area of the crown is calculated by:

$$Crown\ Area = \pi \times \left(\frac{Average\ Diameter}{2}\right)^2 \tag{3.9}$$

Additionally, the condition of each tree's crown is assessed during field measurements to evaluate its health and vitality. Healthy crowns, which are usually full and green, show strong photosynthesis, while signs of stress or disease, such as thinning, discoloration, or dieback, are noted, especially in areas with abnormal or partially infected trees [21]. These observations are important because after obtaining the model, if unusual errors in measured crown diameters are seen, it is possible to trace the disease trees.

The drone’s built-in GPS is used to automatically generate geospatial data for each image, to ensure precise mapping and accurate alignment between the field data and aerial images. The GPS mapping enables creating a spatially coherent dataset for further analysis.

Finally, standardized protocols are followed during data collection to guarantee consistency and repeatability across locations and time periods, which minimize human error and make it replicable.

The field data and aerial imagery provide essential datasets for training and evaluating the model. However, dataset preparation and refinement are required to ensure compatibility with the YOLO model's framework.

3.4.3. Dataset Preparation and Annotation for YOLOv5 Training

Several tree maps are created using aerial images and each tree is given a unique identifier. The field data collected for each tree, including crown diameters, Diameter at Breast Height (DBH), tree type (white or blue spruce) and health condition are organized and linked to the unique identifiers.

For the annotation process, the "LabelImg" [22] tool is used to enclose each tree crown in the aerial images with a bounding box, clearly showing their positions. Each bounding box is saved in a format that can be interpreted by YOLO. The format includes information such as the class, center coordinates, box width, and box height.

It is a time-consuming process since it requires the precise annotation of bounding box locations and accurate labelling to correspond with the unique tree identifiers. This precision is important because it allows for reliable results obtained through applied model, helping improve accuracy. Additionally, the presence of untargeted tree species in the field makes the task more challenging. It is important to correctly identify and exclude these species from the target spruce trees in the maps. Furthermore, extra efforts are required to avoid mislabeling for the overlapping crowns.

To show the annotation process, Figure 3-6 presents a drone image with annotated spruce trees, demonstrating how each tree is identified and labelled in the dataset.

Figure 3-6: Annotated spruce trees

The total dataset for this study contains 2,155 labelled images of white and blue spruce trees. To optimize model training and evaluation, the dataset is divided into three groups.

Training Set: 70% of the total dataset is allocated to the training, allowing the YOLOv5 model to learn key features. This portion (1,508 images) is used to train the model, allowing it to recognize patterns and learn how to predict crown diameters.

Validation Set: 15% of the data (323 images) is used as a validation set to fine tune the model and evaluate its performance during training. The validation set provides feedback to adjust model parameters, helping prevent overfitting by testing the model on data that hasn't been seen during training.

Test Set: The remaining 15% (324 images) forms the test set, used to assess the model's accuracy by comparing predicted crown diameters with actual measurements. This set is strictly reserved for final evaluation after training is complete, ensuring the model's generalization on new and unseen data.

This structured approach to dataset division is crucial for developing a reliable machine learning model for precise tree crown detection [23]. The 70%/15%/15% split is a standard protocol in machine learning projects because it provides an appropriate balance between training, validation, and testing. The 70% for training is often enough to allow the model to learn effectively, while still leaving a sufficient amount of data for both validation and testing. However, this split can be adjusted based on specific needs and the nature of the dataset.

A way to assess whether the split is acceptable is by examining the model's performance during both training and validation. If the model consistently underperforms or has high variance between training and validation performance, this could be an indication that more training data is needed. A "small" dataset may not provide enough examples for the model to generalize effectively, leading to overfitting or underfitting. This is often seen when the model’s accuracy increases significantly on the training set but fails to improve on the validation or test sets.

In contrast, a sufficiently large dataset results in better generalization, with more stable accuracy rates across both training and validation sets. If the model reaches a performance plateau despite the inclusion of additional training data, it may suggest that further increases in the dataset size no longer produce significant improvements.

In cases where the dataset is small, it is beneficial to allocate a larger portion for training (for example, 80%) to maximize the learning potential. On the other hand, a larger, more diverse dataset allows for a more balanced distribution of data across the training, validation, and test sets, which helps the model to generalize better and reduces the risk of overfitting.

Therefore, the size and diversity of the dataset, combined with the model's performance, should guide decisions regarding the allocation of data across training, validation, and testing phases.

Based on the results to be presented in the Result and Discussion chapter, the model demonstrates good generalization during training and validation, with no signs of overfitting or underfitting. As the model's performance begins to plateau, further increases in training data do not significantly improve accuracy. Given this, the initial dataset split of 70% for training, 15% for validation, and 15% for testing is found to be appropriate. Therefore, there is no need to adjust the original split, as it provides a balanced and effective distribution of data for model training and evaluation.

3.5. YOLOv5 Model

YOLOv5 (You Only Look Once version 5) is an advanced object detection model known for its real-time performance and high accuracy. Unlike traditional methods that use multiple steps, like region proposal, classification, and bounding box regression, YOLOv5 processes the entire image in one step, predicting both bounding boxes and class probabilities at the same time. The ability to detection in one step makes YOLOv5 useful particularly for large and complex tasks, such as detecting tree crowns in aerial imagery [24]. The versatility and generalization of YOLOv5 allow it to perform well even with differences in tree crown size, shape, and environmental conditions where trees may be partially hidden or have irregular shapes [25].

YOLOv5 Model Architecture

The architecture of YOLOv5 is made up of three main parts: the Backbone, Neck, and Head, each playing a specific role in object detection [24]. The Backbone extracts important features from the input image, turning it into feature maps that show details like texture, shape, and object boundaries. CSPDarknet53, a modified version of Darknet53, improves computation efficiency while keeping high accuracy by splitting the feature maps to enhance gradient flow. The Neck combines features from the Backbone at different levels of detail. The Path Aggregation Network (PANet) helps merge multi-scale features, which is essential for detecting spruce trees of various sizes, especially when there are a lot of scale variations. The Head makes the final object detection predictions. It provides the coordinates for the bounding boxes, class probabilities, and confidence scores for each detected object, in this case, spruce trees.

This architecture allows YOLOv5 to manage the challenges of tree detection, including irregular shapes, overlapping crowns, and background noise, making it a strong tool for real-time forestry applications. Figure 3-7 shows the architecture of YOLOv5 and its key components [26].

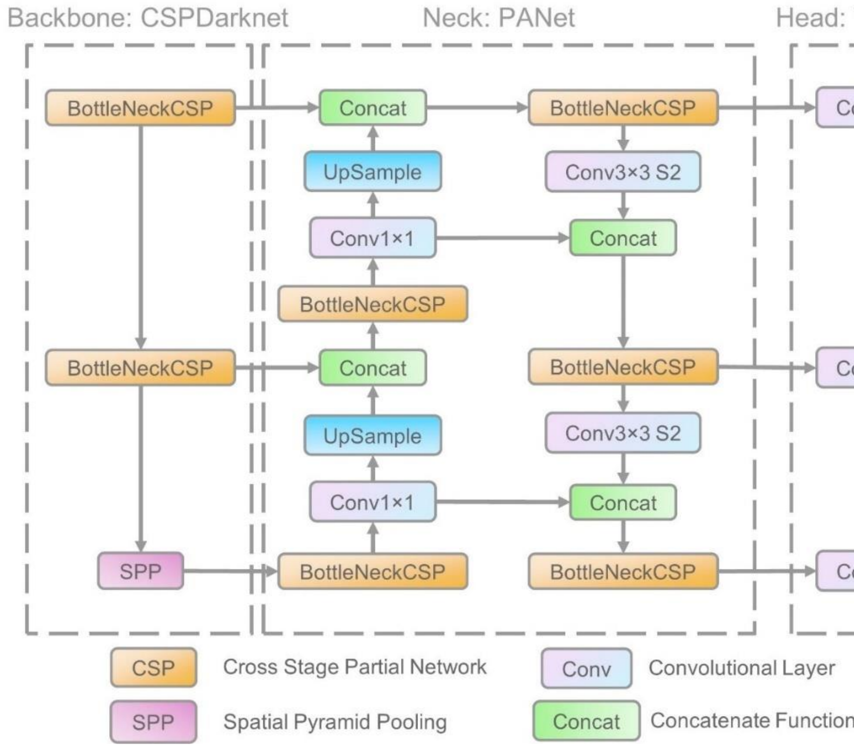

Figure 3-7 : YOLOv5 architecture overview [19]

To provide a detailed view of the YOLOv5 model used in this study, Table 3-1 offers a layer-by-layer breakdown of its architecture. It lists each layer's index, the number of trainable parameters, the module type (such as Conv, C3, and SPPF), and specific configuration details, including input/output channels, kernel sizes, strides, and padding.

The first value in the arguments shows the input channels, indicating how many feature maps enter the layer. The second value represents the output channels, showing how many feature maps leave the layer after processing. The third value is the kernel size, which defines the size of the filter used during convolution. The fourth value is the stride, which shows how far the filter moves across the input data. Lastly, the fifth value refers to padding, which decides if extra pixels are added around the input to maintain spatial dimensions, when the padding is 0, no extra pixels are added.

Table 3-1: YOLOv5 model architecture layer breakdown

Layer	Params	Module	Arguments
0	5,280	Conv	[3, 48, 6, 2, 2]
1	41,664	Conv	[48, 96, 3, 2]
2	65,280	C3	[96, 96, 2]
3	166,272	Conv	[96, 192, 3, 2]
13	1,182,720	C3	[768, 384, 2, 0]
14	74,112	Conv	[384, 192, 1, 1]
17	296,448	C3	[384 192, 2, 0]
18	332,160	Conv	[192, 192, 3, 2]
23	4,134,912	C3	[768, 768, 2, 0]
24	24,246	Detect	[1, [[10, 13, 16, 30, 33, 23] …. [192, 384, 768]]

The parameters represent the trainable weights in each layer, which adjust during training to improve the model's performance in detecting spruce trees. This breakdown explains the complexity and depth of YOLOv5's architecture, showing how each part contributes to its accuracy and efficiency in detecting and segmenting spruce trees from aerial images. This structure demonstrates YOLOv5's strength and its suitability for large-scale areal image processing. The model summary shows that 291 layers, with a total of 20,871,318 parameters, are trained using 48.2 GFLOPs of processing power.

3.5.1.Model Training Process

The YOLOv5 model is trained using "Google Colab" [27], which provides access to an NVIDIA Tesla A100 GPU. This GPU speeds up the processing of the large aerial image datasets. Colab's 12 GB of GPU RAM and 50 GB of storage are sufficient to handle the data and run the model efficiently. The model is implemented using the "PyTorch 1.x" framework, which is supported in Colab. PyTorch's flexible and dynamic structure allow the machine learning tasks to run smoothly and make it easy to train and modify the YOLOv5 model. Colab's Linux-based system also makes sure that the required libraries and dependencies are installed and managed easily. YOLOv5 is used with CSPDarknet53 as the backbone for feature extraction [24]. Pre-trained weights from yolov5s.pt are applied to accelerate the training process [28].

To initiate performance optimization, key hyperparameters are initially set to YOLOv5's default settings. The learning rate is set to ensure stable progress without overshooting during training, and the batch size is chosen to balance memory limits with effective training [25]. It also has a batch size of 16 with resized input images of 640x640 pixels and uses Stochastic Gradient Descent (SGD) as the optimizer. This approach allows the model to benefit from general features and patterns in the pre-training while adapting to the specific characteristics of the new dataset [29]. Other key hyperparameters used during training are shown in Table 3- 2.

Table 3-2: Key hyperparameters used during YOLOv5 training **[25]**

Hyperparameter	Value
Learning Rate	0.01
Momentum	0.937
Weight Decay	0.0005
Warmup Epochs	3.0
Box Loss Gain	0.05
Classification Loss Gain	0.5
Object Loss Gain	1.0
Optimizer	Stochastic Gradient Descent (SGD)

To identify the most effective model for accurate detection, three training stages are structured, enhancing the model's performance with each stage. Initially, a baseline model is trained from scratch using YOLOv5 without any pretrained weights. This stage delivers a control, providing a foundational performance measure to assess the model's capacity to learn directly from the dataset without relying on prior knowledge from other datasets. This approach allows for a clearer evaluation of the effectiveness of advanced techniques applied in later stages.

In the next phase, a pretrained YOLOv5s, previously trained on the COCO dataset, is used. Pretraining on COCO provides a solid range of object recognition skills. Furthermore, fine-tuning on a specific dataset enables quicker convergence and higher starting accuracy, as the model already has basic object detection knowledge.

Finally, more improvements are achieved by training the pretrained YOLOv5s model on an augmented version of the tree crown dataset, incorporating transformations such as Scale Augmentation and Hue, Saturation, and Value (HSV) Adjustments.

Scale Augmentation adjusts the size of objects in images, making the model more resilient to variations in object scale. This adjustment is particularly valuable for aerial images, where changes in drone altitude can affect the apparent size of tree crowns. As a result, the model becomes better at recognizing trees at various scales, enhancing its ability to detect objects consistently despite size variations.

HSV Adjustments, which modify the hue, saturation, and brightness of images, add strength to color and lighting variations. This is especially useful in natural settings like forests, where lighting conditions can impact how trees appear. By exposing the model to a range of colors and lighting scenarios, HSV Adjustments help it focus on structural features of tree crowns and make it less reliant on specific colors.

After analyzing the results obtained through the three stages, the pretrained YOLOv5s model with augmentation is chosen for its effectiveness. This decision is based on key metrics such as mean average precision, precision, and recall. Higher mean average precision (mAP@0.5 and mAP@0.5:0.95) indicates better detection accuracy, while strong precision and recall scores show the model's ability to identify true positives and reduce false detections. The augmented model also avoids overfitting.

The final model is trained twice, once for 60 epochs and for 120 epochs, and after a comparison, the 120 epochs is chosen as the final tuning.

3.5.2. Evaluation Metrics

Evaluating the performance of the YOLOv5 model in this research requires several key metrics that measure the accuracy and reliability of spruce trees detection in aerial images [30]. These metrics are Precision, Recall, F1-Score, Intersection over Union (IoU), and Mean Average Precision (mAP).

In this research, Precision is used to measure the proportion of correctly identified spruce trees among all detected trees. A high precision means fewer false positives, indicating that the model makes fewer incorrect detections. Precision is calculated as [30]:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)} = \frac{Correct\ Detections}{Detected\ Trees} \tag{3.10}$$

Recall that is also called sensitivity, measures the percentage of actual spruce trees that the model correctly detected, can be calculated by [30]:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} = \frac{Correct\ Detections}{Total\ Trees\ (Ground\ Truth)} \tag{3.11}$$

The F1-Score is the harmonic mean of Precision and Recall, offering a single metric that balances both. It is especially useful when considering both false positives and false negatives. The F1-Score is derived as [30]:

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.12}$$

In this research, IoU measures the overlap between the predicted bounding boxes and the actual bounding boxes. It shows how well the predicted trees match the real ones and is used as a threshold to examine if a detection is correct. In this study, a detection is considered correct if the IoU value is 0.5 or higher. It is given by the following formula:

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} = \frac{Overlap\ between\ Predicted\ and\ Actual\ Bounding\ Box}{Combined\ Area\ of\ Predicted\ and\ Actual\ Bounding\ Box - Overlap\ Area} \tag{3.13}$$

Mean Average Precision (mAP), an important metric in object detection, combines these evaluations by averaging Precision across different IoU thresholds, providing an overall accuracy measure. Higher mAP values indicate better detection accuracy. mAP is calculated as follows:

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{3.14}$$

Where AP_i represents the Average Precision at each IoU threshold, and n is the number of IoU thresholds.

Additionally, during training, the model's performance is tracked by observing three key indicators: training loss, validation loss, and convergence behavior. The loss function, which evaluates the accuracy of the model's bounding box predictions, is monitored continuously, with lower training loss values indicating improved performance on the training set. However, if validation loss starts to increase while training loss continues to decrease, it suggests that the model may be overfitting, learning details specific to the training data rather than generalizable patterns. Convergence behavior is assessed by plotting training and validation losses over time; smooth and gradual reductions in these values indicate stable learning, whereas erratic patterns or a plateau without further improvement suggest the model may have reached its optimal performance.

3.5.3.Model Fine-Tuning and Improvements

After training the model, fine-tuning techniques and adjustments are applied to improve its performance in detecting spruce trees. YOLO models apply Non-Maximum Suppression (NMS) by default as part of their output processing. NMS enhances detection quality by refining key parameters, such as confidence thresholds and Intersection over Union (IoU) thresholds, to retain only the most confident bounding boxes, and reduce overlapping detections. By calculating the IoU between pairs of bounding boxes resulting from overlapping crowns, NMS identifies and removes redundant boxes that overlap beyond a specified threshold. This process guarantees that each detected object is represented by a single, high-confidence bounding box, which is particularly useful for avoiding duplicate detections during segmentation.

To further minimize overlap during the segmentation process, careful tuning of NMS parameters is crucial. Adjusting the IoU threshold controls the amount of overlap allowed between bounding boxes before suppression. For example, setting a moderate IoU threshold ensures that valid boxes are retained while still suppressing redundant detections. Similarly, fine-tuning the confidence threshold helps filter out low-quality predictions before NMS is applied, reducing unnecessary computation and improving segmentation accuracy.

In this research, when overlapping tree crowns occur and the YOLO model detects multiple bounding boxes for a single tree in the image, NMS effectively minimizes the impact of redundant bounding boxes. Properly tuned NMS parameters improve precision by removing unnecessary overlaps and recall by maintaining valid detections.

Early tests showed that using the default threshold of 0.25 led to false positives, where non-tree objects are incorrectly detected as spruce trees. By increasing the threshold to 0.5, the model becomes more selective in its predictions, reducing false positives and improving precision.

The second adjustment involves fine-tuning of the Intersection over Union (IoU) threshold. Initially, a threshold of 0.5 was used, but this caused missed detections in some cases, especially where tree crowns overlapped. To improve it, the IoU threshold was lowered slightly to 0.4, allowing the model to capture more trees, particularly in dense forests where overlap is common. This approach improves the model's recall without sacrificing precision, as confirmed by the performance metrics.

3.6. Watershed Segmentation

The watershed algorithm is a well-known image segmentation technique inspired by geographical watersheds, where natural divides separate regions. The principle of the watershed transform involves treating intensity values as hills and basins. For segmentation, the basins can be "flooded" to merge corresponding regions. Figure 3-8 illustrates this process, showing how the flooding helps to combine regions for more accurate segmentation. This technique is especially useful for separating overlapping or closely packed objects, such as tree crowns in dense forests [31].

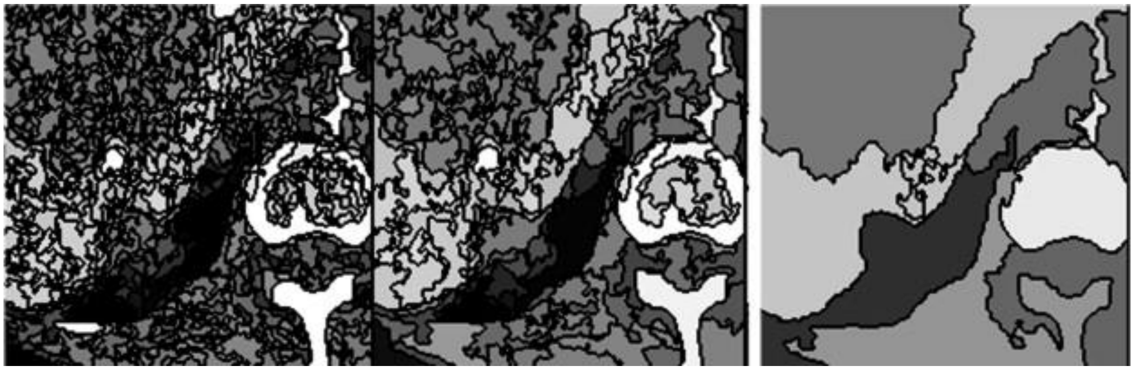

Figure 3-8: Merging of basins into larger regions in Watershed Segmentation

In this study, watershed segmentation is applied after the initial tree detection to refine the boundaries of the detected trees. This approach is crucial for separating individual crowns, which is essential for precise crown diameter and area measurements. The process starts with bounding box loading and conversion.

3.6.1. Watershed Bounding Box Loading

The process begins by loading the original images from the dataset, along with their corresponding annotation files generated by YOLOv5. In these files, the locations of the predicted spruce trees are given as pixel-based coordinates of the bounding boxes, representing the actual positions in the image. Since image segmentation algorithms, like watershed, operate at the pixel level, these pixel-based coordinates enable precise mapping of detected regions onto the original image. To verify alignment, bounding boxes are visualized on the images, allowing rough estimates of the crown areas of each detected tree. As illustrated in Figure 3-9, each spruce tree is labelled and numbered, allowing visual confirmation of the initial detections.

Figure 3-9: Detected labeled spruce trees

At this stage, each identified spruce tree is isolated as an individual image for watershed segmentation, with all other tree species and unrelated objects removed. This approach ensures that watershed segmentation can concentrate on refining the crown boundaries of each spruce tree with maximum accuracy.

3.6.2. Segmentation Preprocessing

To prepare the cropped images for processing, they are converted from color to grayscale. Grayscale simplifies the image by reducing the three primary color channels: red, green, and blue (RGB) to a single intensity value per pixel, leading to lowering computational effort. Grayscale is particularly beneficial when working with large datasets, as it decreases the data load the model must process [32]. Moreover, converting to grayscale helps minimize noise by focusing on intensity rather than color, facilitating the application of filters such as Gaussian blurring. Since color information is less critical for tree crown detection, using grayscale ensures the algorithm concentrates on the shape and structure, ultimately enhancing segmentation accuracy.

Subsequently, Gaussian blurring is applied to the grayscale image to reduce noise that could otherwise negatively impact segmentation accuracy. By averaging pixel values in a region, Gaussian blurring smooths out minor variations and less prominent edges, resulting in a cleaner image for precise segmentation. [33].

In the next step, Sobel filtering is applied to detect edges in the image. The Sobel filter calculates the intensity gradient at each pixel, identifying areas with sharp changes in brightness, which usually indicate the edges of objects. It helps to clearly define the boundaries of the tree crowns, making them more visible for further processing. By detecting the edges, the algorithm can more accurately separate the tree crowns from the background in the image [34].

Following, the image is converted into a binary format using Otsu's thresholding method. A binary image contains only two-pixel values: 0 (black) for the background and 1 (white) for the objects of interest (tree crown in current research). Otsu’s method automatically calculates the optimal threshold to distinguish the tree crown from the background, ensuring that the tree crowns are clearly highlighted for the next stage of processing [35].

3.6.3. Watershed Segmentation for Crown Refinement

Segmentation begins by applying the distance transform to the binary image, which calculates the distance of each pixel from the nearest background pixel. This transform assigns higher values to pixels at the centers of regions, highlighting these core areas. The mathematical formula for the distance transform is:

$$D(x,y) = min_{(x_b,y_b)\in B}\left||(x,y) - (x_b,y_b)\right|| \tag{3.15}$$

Where $D(x, y)$ represents the distance value for pixel (x, y) , and (x_b, y_b) represents the coordinates of the nearest background pixel. This step makes the centers of objects more prominent, setting the stage for accurate segmentation [36].

Next, local maxima within the distance-transformed image are identified. These maxima, which correspond to the central points of objects, serve as markers for the watershed algorithm. The algorithm then initiates the "flooding" process from these markers, expanding outward to define separate regions. This process effectively separates overlapping regions from each other and from the background, achieving clear segmentation boundaries [36].

After applying watershed segmentation, the contours of each segmented region are extracted, representing the outer boundaries of the identified regions. At this stage, noise reduction is applied to emphasize the most relevant shapes in the image. By filtering out irrelevant regions and focusing on the largest remaining area, the segmentation isolates the primary region of interest, which corresponds to the tree crown [36].

3.6.4. Crown Diameter and Area Calculation

The crown diameter and area are essential measurements that provide valuable information about tree size, biomass, and forest structure [37]. After refining the segmentation and extracting the contours, the farthest points along each crown’s boundary are identified to calculate the crown diameter. The distance between the two farthest points is calculated using their coordinates. This pixel distance is then converted into real world units by multiplying it with the Ground Sample Distance (GSD). Since the GSD changes depending on the altitude of the image, the appropriate GSD value is applied to each image.

After calculating the crown diameter, the next step is to measure the crown area. Since the contour forms a closed boundary around the tree crown, the area is determined by counting the number of pixels within the contour. Each pixel corresponds to a known area based on the Ground Sample Distance (GSD). By multiplying the number of pixels inside the contour by the area of one pixel, the actual crown area is calculated.

This process is repeated for all detected trees in each image. Once all trees in an image are processed, the same steps are applied across the entire dataset. This dataset will be used for statistical analysis, comparisons, and model training, Figure 3-10 presents a sample image featuring several trees, each with its detected crown outlined in red contours.

Figure 3-10: Detected tree crowns outlined in red contours

3.7. Linear Regression for Improving Crown and Area Diameter

In this section, a linear regression model is applied to correct the discrepancies between the predicted crown diameters from the YOLOv5 detection and watershed segmentation processes, and the actual field-measured diameters, however, there are systematic errors, particularly in underestimating crown sizes. Linear regression is chosen because of its simplicity, interpretability, and effectiveness in identifying and correcting systematic biases in predictions. This approach provides a straightforward mapping between predicted and actual values, which is crucial when dealing with systematic underestimations that need a consistent correction across the dataset. Additionally, its mathematical foundation ensures robustness and ease of implementation, making it a practical choice for forestry applications where simplicity and reliability are key. Therefore, to improve the accuracy of crown diameter predictions, linear regression is used to derive the relationship between the predicted crown diameter, $D_{\text{corrected}}$, and the actual field-measured diameter, $D_{\text{predicted}}$. The basic form of the regression equation is:

$$D_{corrected} = m \cdot D_{predicted} + b \tag{3.16}$$

Where m represents the slope of the line, indicating how much the predicted diameter needs to be adjusted for each unit increase in the predicted value, while b is the intercept, showing the baseline correction applied to all predictions.

By applying this model, the systematic underestimation of crown sizes is corrected. The regression model identifies patterns in the discrepancies and adjusts the predictions, resulting in more accurate crown diameter estimates. The reason for using linear regression lies in its ability to model a straightforward relationship between variables while minimizing the impact of noise in the data. This is particularly useful when systematic errors dominate, as it directly corrects systematic errors without overcomplicating the methodology. After training the model on the differences between predicted and measured values, it is applied to all predicted crown diameters. The training process minimizes the sum of squared differences (Errors) between the predicted diameters and the actual field-measured diameters. This approach is known as least squares regression and is commonly used to find the best fitting line. which can be expressed as:

$$m = \frac{n\sum(x_i y_i) - (\sum x_i)(\sum y_i)}{n\sum(x_i^2) - (\sum x_i)^2} \tag{3.17}$$

$$b = \frac{\sum y_i - m\sum x_i}{n} \tag{3.18}$$

$$Error = \sum_{i=1}^{n}(x_i - y_i)^2 \tag{3.19}$$

Where m is the slope, n is the number of data points, x_i is the data points for the independent variables or actual field-measured diameters, y_i is the data points for the dependent variables or predicted diameters by YOLO model and b is the intercept.

These formulas are derived by minimizing the Error or sum of squared differences between the actual values and the predicted values. Once m and b are computed, the best-fit line for the dataset is determined. In other words, by minimizing this error, the model learns the best relationship between the predicted and actual diameters, leading to the corrected formula:

$$Corrected\ Diameter = 0.53 \times Predicted\ Diameter + 2.21 \tag{3.20}$$

In addition to the crown areas calculation from the aerial images, actual crown areas are measured in the field, manually (details in 3.4.2 Data Field Collection). The field-measured crown areas provide a dataset for validating the predictions calculated by using the aerial images, and its reliability can be examined by comparing the predicted and actual areas.

To correct any discrepancies between the predicted and actual crown areas, the empirical regression model, which is originally used to adjust crown diameter predictions, is applied to the crown area predictions. Since the regression model had already aligned the pixel-based measurements with real world dimensions for crown diameters, applying the same correction to the crown areas is maintained consistency across both sets of predictions. This approach eliminates the need to create a separate regression model for area adjustments.

Despite its advantages, linear regression has limitations. It assumes a linear relationship between variables, which may not always reflect the true complexity of the discrepancies. For example, external factors such as tree age, species variation, canopy density, and environmental influences may introduce nonlinear patterns that the linear model cannot address. Moreover, the model’s reliance on the assumption of homoscedasticity (equal variance across predictions) can be problematic if prediction errors vary based on crown size. This may lead to under- or over-corrections for certain size ranges. Additionally, the regression's accuracy depends heavily on the dataset's size and quality. A small or noisy dataset may limit the model’s ability to identify accurate relationships, potentially resulting in less effective corrections. Such issues highlight the importance of having a comprehensive and high-quality dataset to train the model.

Even with these challenges, the correction process improved the accuracy of the crown area measurements, bringing them into closer alignment with the field data. This consistency between the predicted and field-measured values enhances the reliability of the model, ensuring it is suitable for practical applications in forestry management.

3.8. Evaluation of Models for Diameter Prediction

The accuracy of the model for detecting crown diameter is evaluated using several statistical metrics to assess the performance and reliability of the predicted crown measurements. These evaluations are essential for verifying the model’s capacity to accurately reflect the true crown dimensions. Five key metrics are employed to evaluate the accuracy of the regression models.

In the following formulas, n represents the number of observations, y_i denotes the actual values, and $\widehat{y_i}$ denotes the predicted values.

The Mean Absolute Error (MAE) formula is used to calculate the average magnitude of the errors between the predicted and actual values. It is valuable for giving a simple, easy-to-interpret measure of error that does not overly emphasize large errors. The formula for MAE is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} /y_i - \widehat{y_i}/ \tag{3.21}$$

The Mean Squared Error (MSE) formula is applied to assess the average squared difference between predicted and actual values. By squaring the errors, it places increased importance on larger deviations, making it particularly sensitive to significant prediction errors. The formula for MSE is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y_i})^2 \tag{3.22}$$

Root Mean Squared Error (RMSE), calculated as the square root of the average squared differences, provides an easily interpretable measure of error in the same units as the target variable. It is especially useful for giving greater weight to larger errors, similar to MSE. The formula for RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y_i})^2} \tag{3.23}$$

Mean Percentage Error (MPE) is measured the average percentage error between the predicted and actual values. This metric is particularly beneficial for understanding how significant the errors are in relation to the actual values, making it ideal when relative accuracy is important. The formula for MPE is:

$$MPE = \frac{100}{n}\sum_{i=1}^{n}\frac{y_i - \widehat{y_i}}{y_i} \tag{3.24}$$

Accuracy, derived from the MPE, is employed to quantify how closely the predicted values match the actual values in percentage terms. This measure is easy to interpret and provides a straightforward understanding of how well the model is performing. The formula for accuracy is:

$$Accuracy = 100\% - MPE \tag{3.25}$$

3.9. Estimating Biomass Through Crown Diameter and DBH Correlation

Although not the primary focus of this research, the linear relationship between crown diameter and Diameter at Breast Height (DBH) is discussed to demonstrate the practical applications of the study's findings. In forestry, DBH is often strongly correlated with crown diameter, which allows foresters to predict DBH using crown measurements obtained from aerial imagery. This relationship has significant implications for estimating tree biomass and forest health without the need for ground-based measurements [37].

The biomass estimation formula used is:

$$B \ = \ \alpha \times (DBH)^2 \times H \tag{3.26}$$

Where B represents the estimated biomass, DBH is the Diameter at Breast Height (measured or predicted), H is the height of the tree (less critical and assumed equal for all trees in forest), α is a species-specific coefficient that accounts for tree form and wood density.

This formula highlights how the crown diameter detected in this study can indirectly aid in biomass estimation by using the predicted DBH and an assumed average tree height. By establishing the relationship between crown diameter and DBH, foresters can estimate biomass more efficiently using aerial imagery, reducing the need for extensive field measurements.

4. Results and Discussion

This chapter presents the results of the calibration check process, the YOLOv5 object detection model, the watershed segmentation algorithm, regression models, and research applications.

4.1. Camera Calibration Check Performance

The drone manufacturer has provided camera parameters such as sensor width (17.3 mm), focal length (12 mm), and the FOV (84°). These parameters along with captured images altitude and image size (5280 × 3956 pixels), are used to calculate the GSD and consequently the image real world width. Table 4-1 shows real world image width, the GSD, and FOV at altitudes of 50 m and 70 m, using formulas 3.1 to 3.5. As can be seen in table, at the altitude of 50 m, the GSD is 0.01365 (m/pixel), meaning each pixel in the image corresponds to approximately 1.36 cm of ground distance. At the altitude of 70 m, the GSD increased to 0.01911 (m/pixel), covering more ground distance, but reducing image resolution. In addition, results reveal that the real world image width at the altitude of 50 m and 70 m are 71.98 m and 100.78 m, respectively.

Table 4-1: GSD and image width at varying altitudes

Flight Altitude (m)	Real World Image Width (m)	GSD (m/pixel)	HFOV	FOV
50	71.98	0.01365	71.5°	84°
70	100.78	0.01911	71.5°	84°

Table 4-1 also shows the importance of considering both GSD and FOV when analyzing aerial images, as those directly affect the level of details and accuracy in the measurements.

The overall accuracy, calculated by averaging the accuracy of each measurement, is 95.7%. The minimal measurement error and high accuracy achieved after the calibration check indicate that further adjustments to the camera parameters are unnecessary [38]. The current

Field of View (FOV) and Ground Sample Distance (GSD) settings have been verified and will be applied to accurately calculate the tree crown size and area.

4.2. Field Measurements

The field measurements produce a comprehensive dataset that includes both DBH and crown diameters for 2,155 individual spruce trees. The crown diameter distribution, shown in Figure 4-1, provides insights into the variation of tree crown sizes in the study area. The diagram shows the crown diameters split into ranges on the x-axis, from 2.0 to 7.5 meters, and the number of trees in each range on the y-axis.

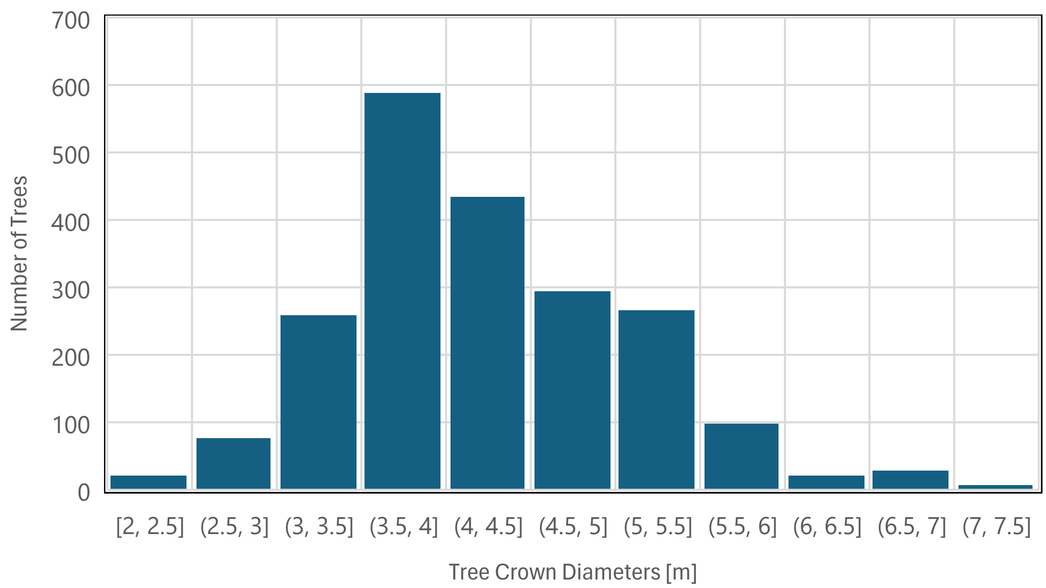

Figure 4-1: Crown diameter distribution of measured trees

This distribution of crown diameters highlights the structural diversity of the tree population in the study area. The variation in crown sizes provides valuable information for improving the machine learning model, especially in distinguishing between different tree sizes and health conditions.

4.3. Results and Analysis of YOLOv5

The results of the three model stages, baseline YOLOv5, pretrained YOLOv5s (COCO), and pretrained YOLOv5s with data augmentation, are analyzed to determine the most effective model for spruce tree detection. To provide an assessment, four primary metrics are chosen: mAP@0.5:0.95, F1 Score, Precision, and Recall. These metrics are selected for their comprehensive coverage of detection accuracy and the model's ability to balance true and false detections.

The mAP@0.5:0.95 metric evaluates detection accuracy across a range of IoU thresholds, providing insights into each model's effectiveness in identifying spruce tree at varying levels of overlap. F1 Score, which balances precision and recall, is essential for this task as it reflects the model's overall success in detecting true positives while minimizing false positives. Precision measures the proportion of correctly identified spruce trees out of all detections, making it particularly useful in minimizing false detections. Recall, on the other hand, assesses the model's capability to detect all true spruce trees within the images, highlighting its sensitivity to actual positives. Together, these four metrics offer a well-rounded view of model performance, illustrating the benefits achieved at each stage through pretraining and augmentation. Figure 4-2 illustrates the F1 score progression over 120 epochs across three models. The pretrained model with augmentation consistently achieves the highest F1 Score, indicating a strong balance between precision and recall.

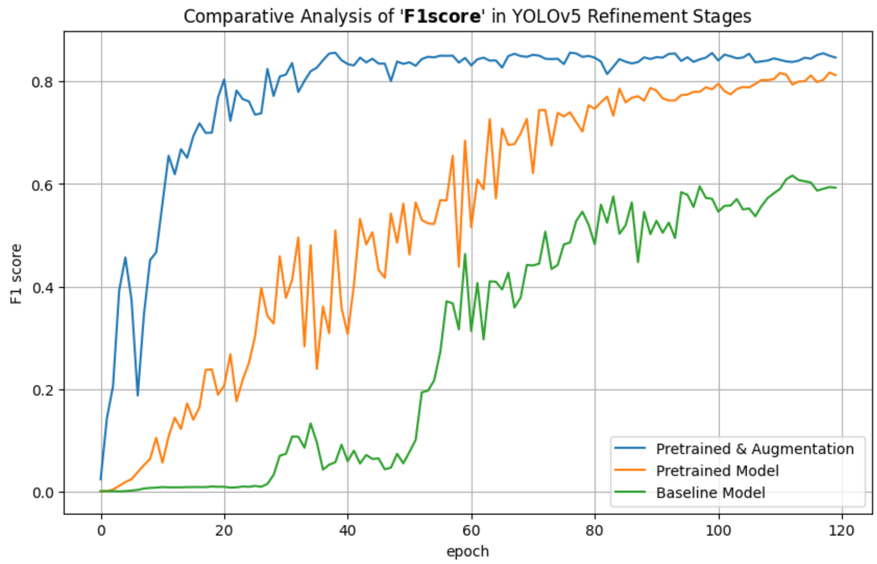

Figure 4-2: F1 Score comparison across training stages

Figure 4-3 presents the mAP@0.5:0.95 metric, showing detection accuracy across varying IoU thresholds. Similar to the F1 Score, the pretrained model with augmentation outperforms the other models, achieving the highest mAP values and demonstrating improved detection accuracy.

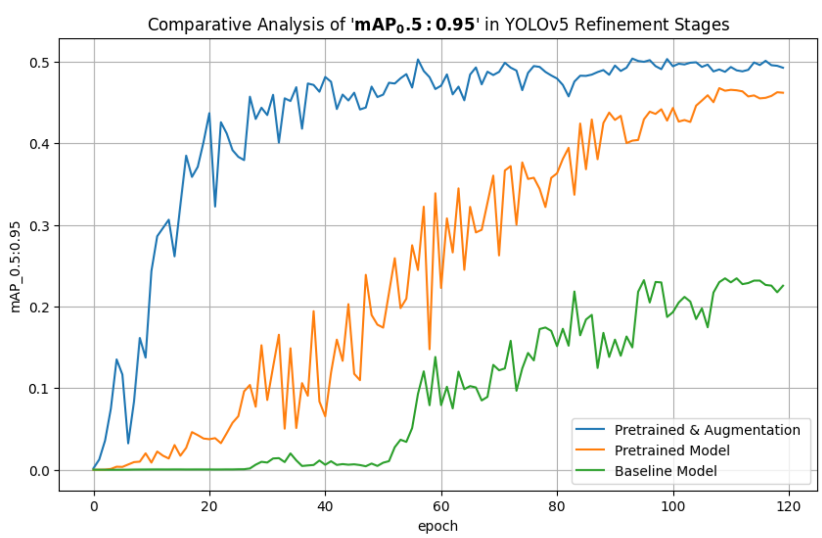

Figure 4-3: mAP@0.5:0.95 comparison across training stages

Table 4-2 summarizes the performance of each model stage across selected key metrics. This comparison emphasises the effectiveness of the pretrained model with augmentation, confirming its selection as the optimal model for this research.

Table 4-2: Performance metrics for each model stage

Model	mAP@0.5	mAP@0.5:0.95	Precision	Recall	F1 Score
Baseline YOLOv5	0.58	0.22	0.61	0.57	0.59
Pretrained YOLOv5s (COCO)	0.86	0.46	0.83	0.78	0.81
Pretrained YOLOv5s + Augmentation	0.88	0.49	0.84	0.84	0.84

The selected model, is trained for 60 epochs with a default confidence threshold of 0.25 and an IoU threshold of 0.5, performed reasonably well, achieving precision, recall, and F1-Score values of around 0.82. However, as shown in Table 4-3, by comparing with the model trained with 120 epochs, the 120-epoch training demonstrates greater improvements in performance metrics. Precision and recall values for 120-epoch trained model stand in 0.84, and 0.85, respectively, and the F1-Score of 0.84 reflects a balance between precision and recall, confirming the model's robustness in this application.

Table 4-3: Performance metrics for YOLOv5 tree crown detection

Metric	**60 Epochs**	**120 Epochs**
Precision	0.82	0.84
Recall	0.83	0.85
F1-Score	0.83	0.84

The convergence behavior of the model during the 120-epoch training process is illustrated in Figure 4-4, which shows the reduction in both train box loss and validation box loss over time. The figure clearly indicates a consistent decline in losses, reflecting the model's improved learning and prediction capabilities as training progressed. This reduction in losses suggests that the model become more accurate in identifying spruce trees and predicting bounding boxes.

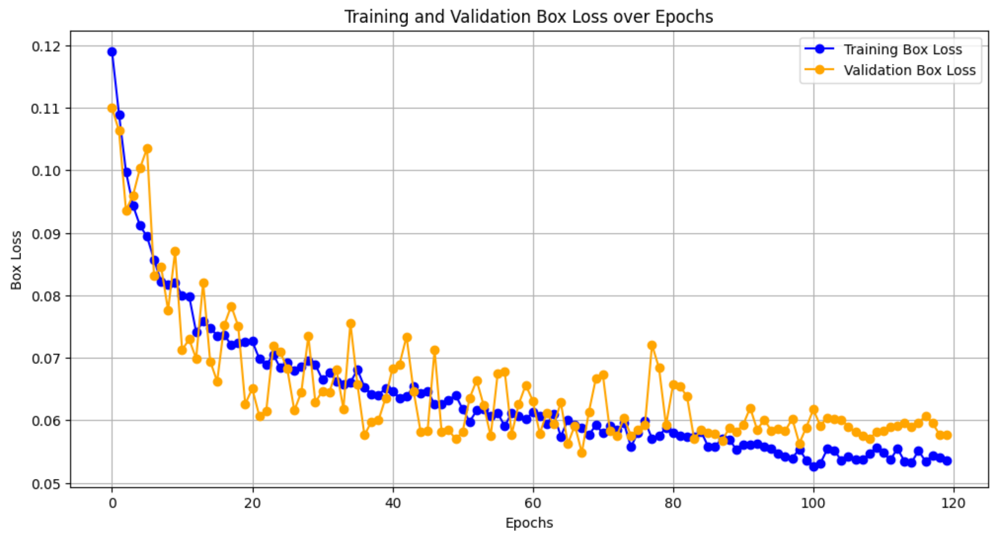

Figure 4-4: Training and validation box loss over epochs

The figure 4-5 shows the "Precision and Recall Graph" where both metrics exhibit a steady increase over time. At 60 epochs, recall is around 0.75, and precision is approximately 0.8. By 120 epochs, recall improves to 0.85, while precision reaches 0.84, indicating a consistent enhancement in the model's accuracy and detection capabilities as training progresses.

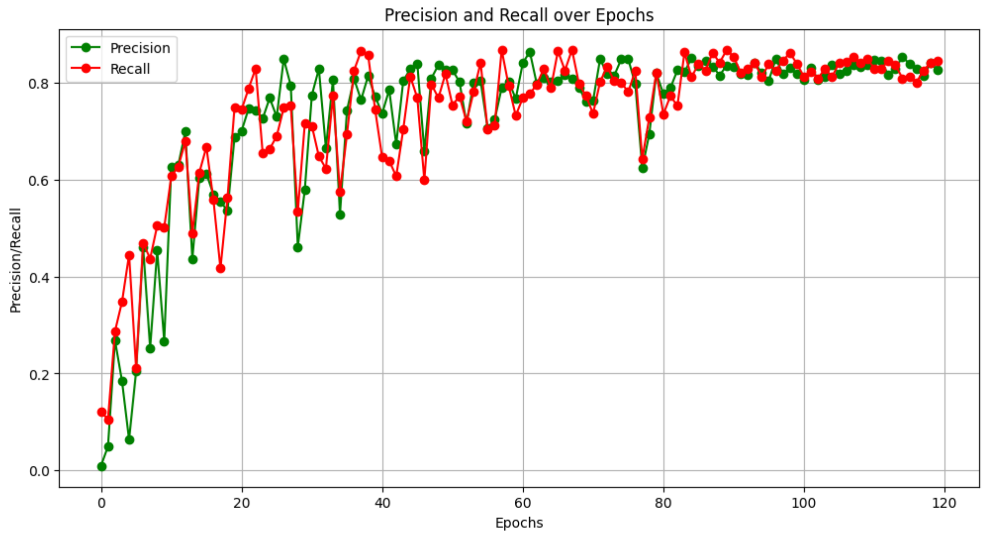

Figure 4-5: Precision and recall progression over epochs

The figure 4-6 illustrates the "F1 Score Over Epochs." The F1 score shows a consistent upward trend as the training progresses. At 60 epochs, the F1 score reaches approximately 0.75. By 120 epochs, the F1 score stabilizes at around 0.84, demonstrating enhanced model performance and reliability as the number of epochs increases. This trend highlights the continued refinement of the model's accuracy through additional training epochs.

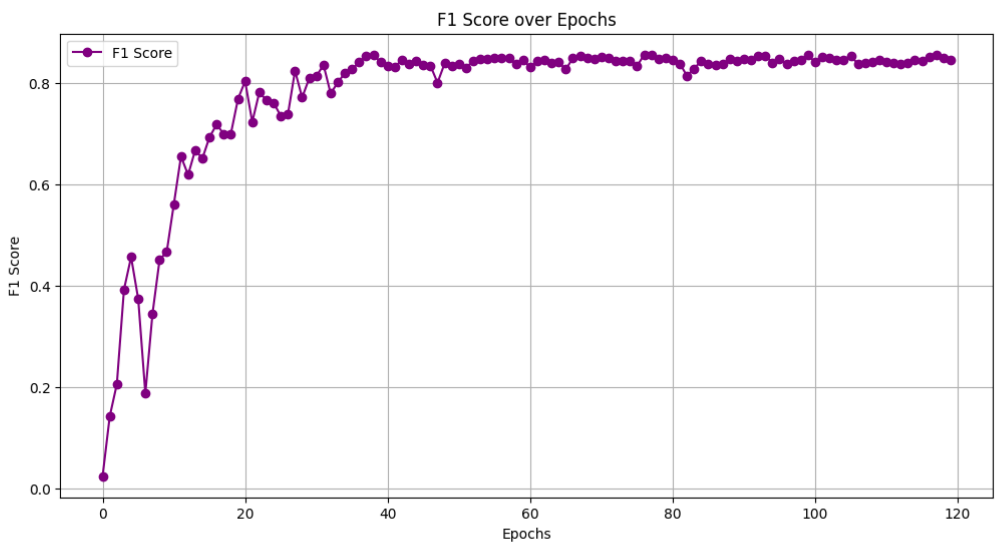

Figure 4-6: F1 Score over epochs

The YOLOv5 model successfully identifies and localizes spruce trees in aerial images using bounding boxes, achieving rapid detection speeds of approximately 0.3 milliseconds per image. Figure 4-7 shows examples of the detection results, with bounding boxes drawn around spruce trees from different tree species. The results are based on drone-captured data at an altitude of 50 meters. In this sample image, the blue boxes represent the manual labels, while the green boxes show the model-predicted labels.

Figure 4-7: Sample detection of spruce trees

Following the fine-tuning adjustments, the model's precision is improved, rising from 0.84 to 0.89 after re-tuning the Non-Maximum Suppression (NMS) thresholds. By increasing the confidence threshold from 0.25 to 0.5, the model becomes more selective, reducing false positives that had previously resulted in non-tree objects being misclassified as spruce trees. Additionally, adjusting the Intersection over Union (IoU) threshold from 0.5 to 0.4 allows the model to capture more overlapping tree crowns, particularly in dense forest areas. Together, these targeted refinements to the NMS thresholds contributed to a notable improvement in detection accuracy, as reflected in the increased precision score.

This study demonstrates a strong performance in individual tree crown segmentation, showing results compared to previous studies. Fan et al. (2024) reported an F1-score of 0.67 for Mask R-CNN, whereas the model presented in this research achieved an F1-score of 0.84 after 120 epochs, reflecting superior accuracy. Additionally, the precision and recall values attained in this study, 0.89 and 0.85, respectively, exceed those of both Fan et al. and Chen et al., with the latter reporting a mAP of 65.29% and an mIoU of 80.44%. These results underscore the effectiveness of the current approach for precise crown detection.

Similarly, the accuracy metrics of the current model demonstrate competitive results when compared to Studies 3, 4, 5, and 6. Liu et al. (2024) and Yu et al. (2024) achieved F1-scores of 0.892 and 0.94, respectively, results that closely align with the 0.84 F1-score reported in this study, even though those studies employed LiDAR data.

4.4. Watershed Segmentation Results

The trees detected by the YOLO model serve as the initial input for the watershed segmentation process, enabling the refinement of tree crown boundaries. Figure 4-8 illustrates three samples of these detected trees. The outcomes of each step in the segmentation process are explained in this section, with the same trees shown repeatedly across the images to highlight the sequence of changes.

Figure 4-8: Individual spruce trees detected by YOLO for watershed segmentation

The conversion of cropped images to grayscale simplifies the data by reducing the three primary colour channels (RGB) to a single intensity value per pixel. The outcome of this grayscale conversion is shown in Figure 4-9, where the images reflect the refined intensity based data ready for further processing.

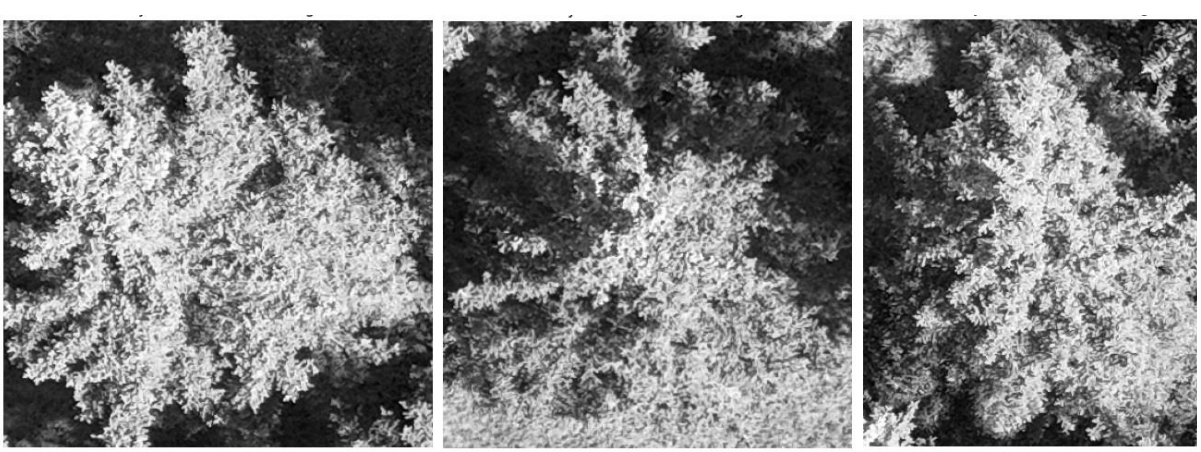

Figure 4-9: Grayscale conversion of cropped images

The application of Gaussian blurring smooths the grayscale image, reducing noise and eliminating sharp edges, which enhances the accuracy of the segmentation process. The result of this blurring step is shown in Figure 4-10.

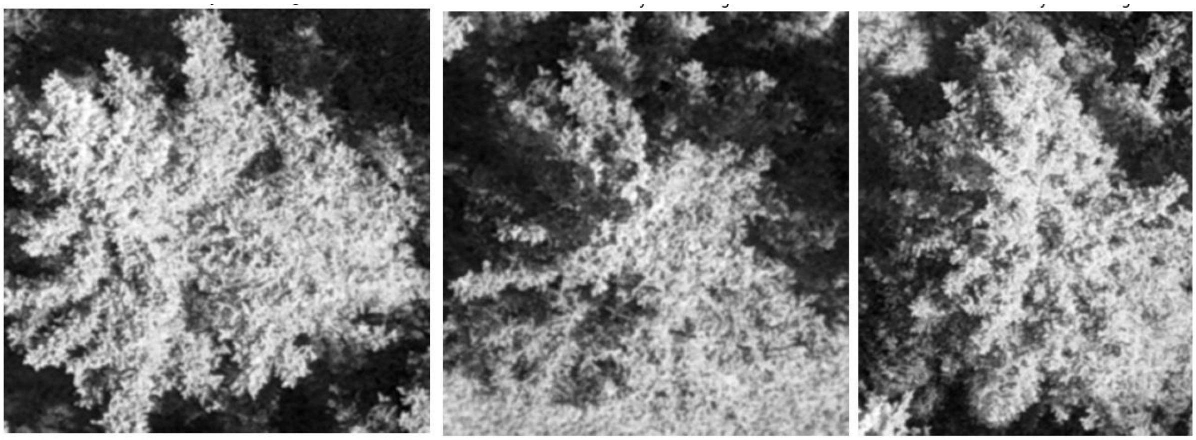

Figure 4-10: Gaussian blurring applied to the grayscale image

In the next step, Sobel filtering detects edges in the image by calculating intensity gradients, making the objects boundaries more distinct and improving objects separation from the background. The result of this edge detection is shown in Figure 4-11.

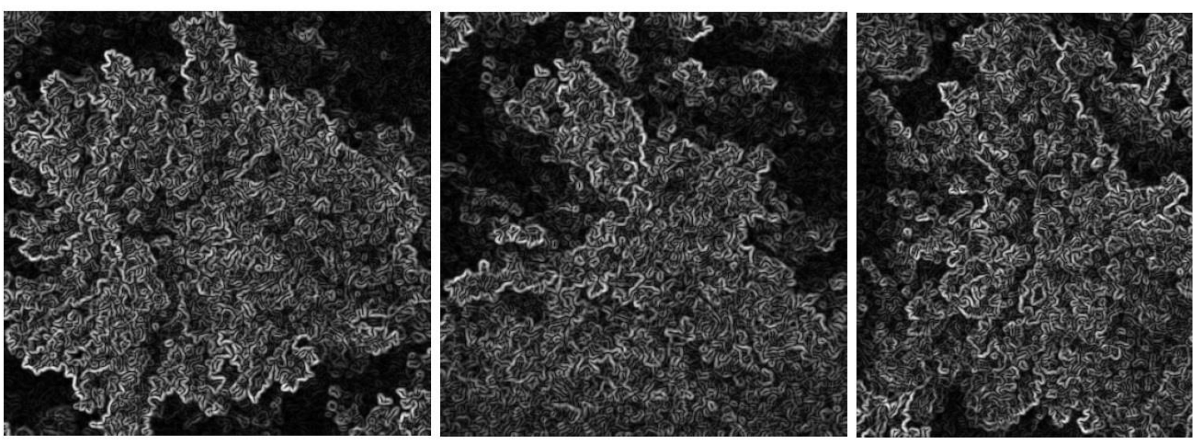

Figure 4-11: Sobel filtering results

The image is converted into a binary format using Otsu's thresholding, clearly distinguishing objects from the background by assigning two-pixel values, with 0 for the background and 1. The resulting binary image, ready for further segmentation, is shown in Figure 4-12.

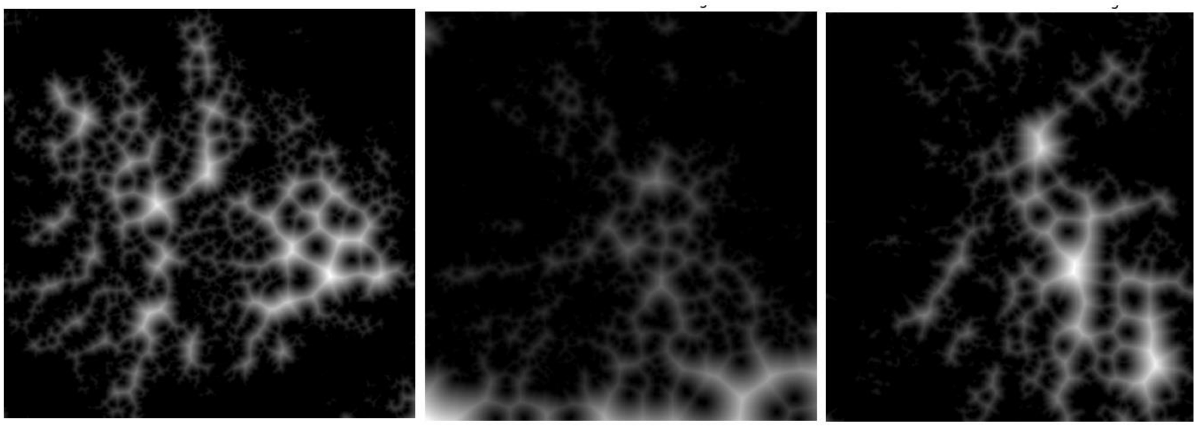

Figure 4-12: Binary image generated using Otsu's thresholding

Segmentation begins by applying the distance transform to the binary image, which assigns higher values to the central areas of the objects, guiding the segmentation process. The local maxima, representing these central points, serve as markers for the watershed algorithm to begin accurate object separation. The result of this process is shown in Figure 4-13.

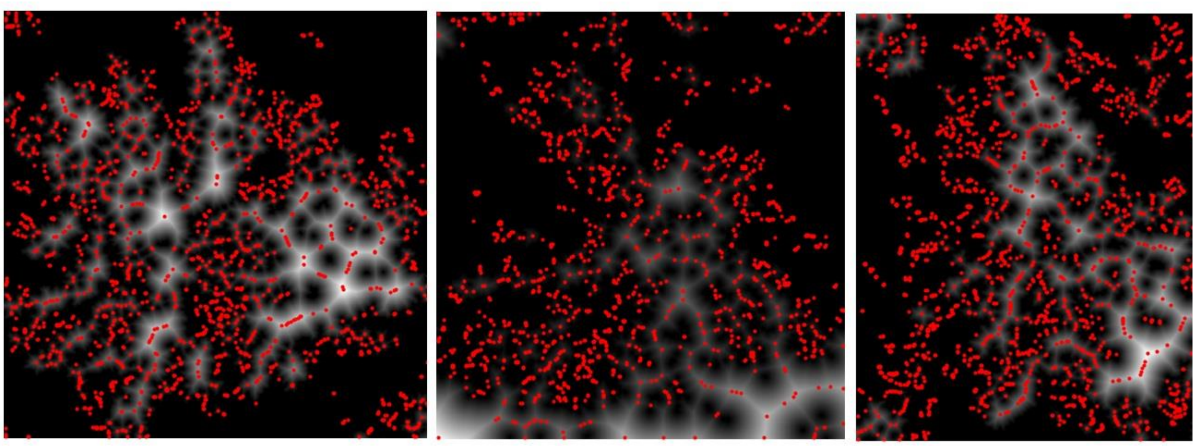

Figure 4-13: Distance transform with local maxima

The watershed algorithm is applied, resulting in the segmentation of tree crowns marked by distinct regions shown in Figure 4-14.

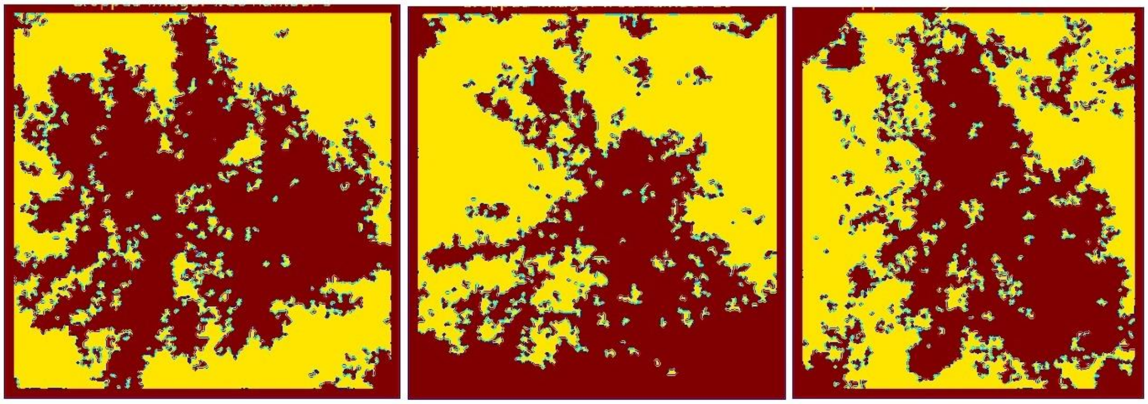

Figure 4-14: Distinct regions using watershed algorithm

Following the segmentation, the contours of each tree crown are extracted, outlining their outer boundaries, as shown in Figure 4-15.

Figure 4-15: Contours of segmented regions

At this stage, noise is filtered out, retaining only the largest regions corresponding to the tree crown. The contour of these segmented crown is shown in Figure 4-16.

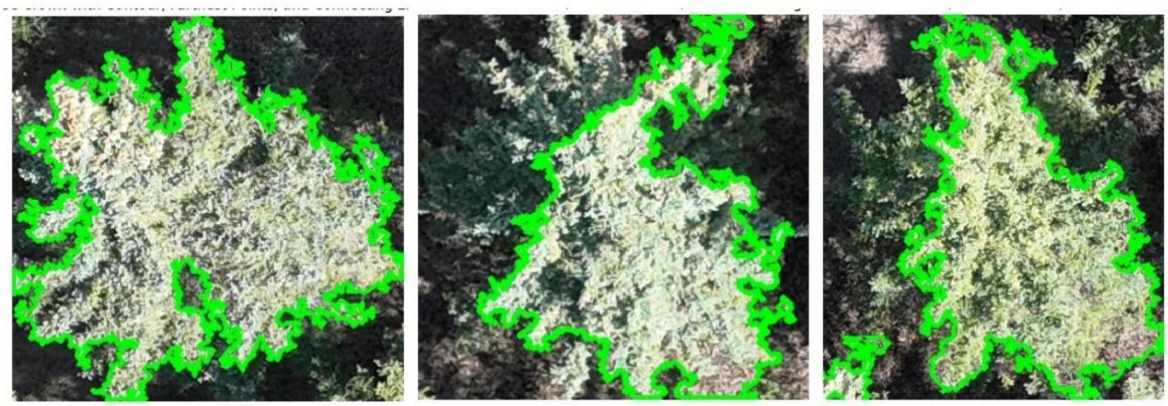

Figure 4-16: Tree crown detection

After refining the segmentation and extracting contours, the crown diameter is calculated by identifying the farthest points along each crown’s boundary, as shown in Figure 4-17. The pixel distance between these points is converted into real world units using the Ground Sample Distance (GSD).

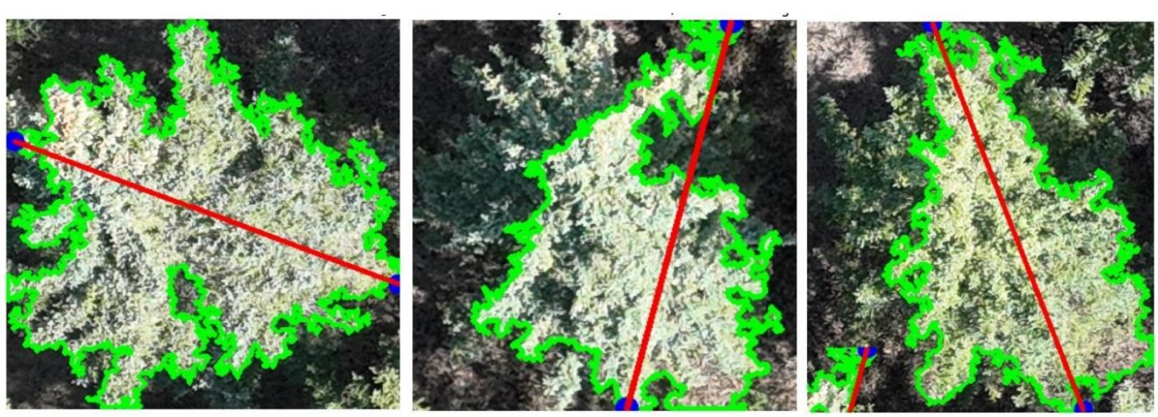

Figure 4-17: Detected diameter lines on tree crowns

After calculating the crown diameter, the crown area is measured by counting the pixels within the contour and converting them into real world units using the Ground Sample Distance (GSD). The result is illustrated in Figure 4-18, with the detected areas highlighted for clarity.

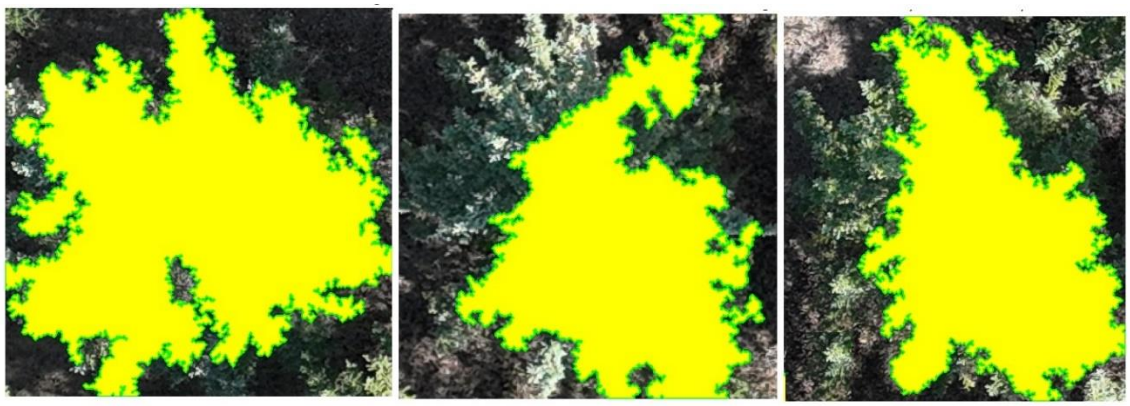

Figure 4-18: Detected tree crowns area

4.5. Evaluation of Regression Model

As shown in Figure 4-19, the discrepancies between the predicted and actual crown diameters reveal a trend of underestimation. The regression model effectively reduces the discrepancies.

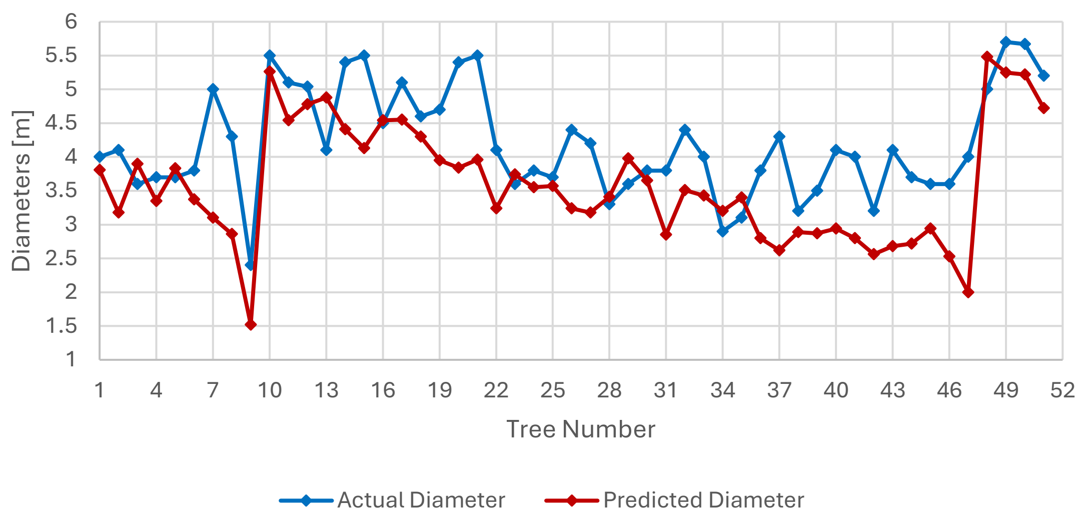

Figure 4-19: Actual and predicted tree crown diameters

By addressing the YOLOv5 and watershed segmentation systematic underestimations, the regression model enhances the accuracy of crown diameter predictions. The effectiveness of the regression model is demonstrated in Figure 4-20 by comparing corrected predicted diameter with the actual diameter.

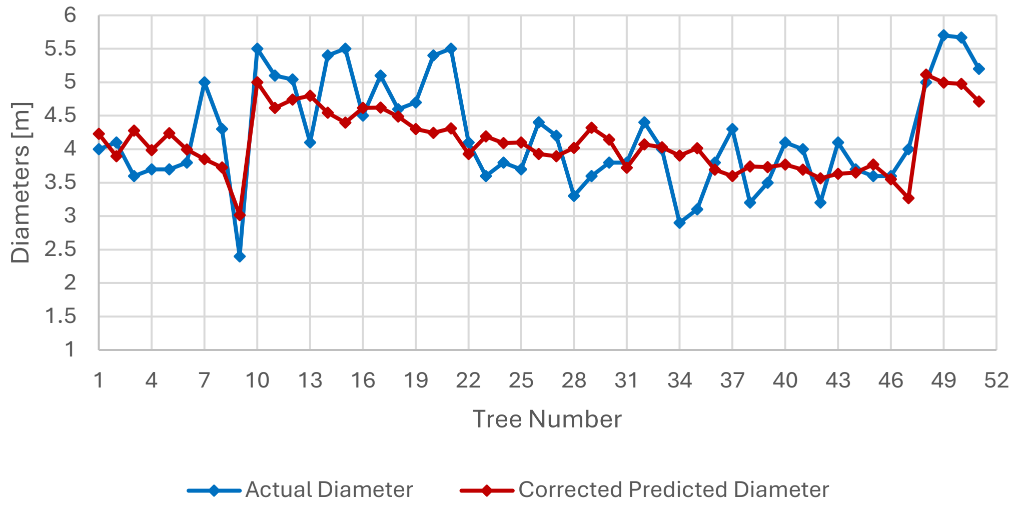

Figure 4-20: Actual vs. Corrected predicted diameter

The accuracy of the regression model is assessed using key statistical metrics at two altitudes, 50m and 70m, which clearly show the improvements achieved through the correction process. The Mean Absolute Error (MAE) improved from 0.73 to 0.46 at 50m, and from 0.80 to 0.44 at 70m. Similarly, the Mean Squared Error (MSE) decreased from 0.90 to 0.34 at 50m, and from 0.99 to 0.35 at 70m. The Root Mean Squared Error (RMSE) is reduced from 0.94 to 0.58, and from 0.99 to 0.59 at 50m and 70m, respectively. The Mean Percentage Error (MPE) also sees significant improvements, dropping from 17.10% to 11.13% at 50m, and from 18.49% to 10.90% at 70m. Lastly, accuracy increases from 82.90% to 88.87% at 50m, and from 81.51% to 89.10% at 70m. Table 4-4 provides a detailed comparison of these metrics before and after correction.

Table 4-4: Estimation accuracy comparison at 50m and 70m

Metrics	50m Before Correction	50m After Correction	70m Before Correction	70m After Correction
MAE	0.73	0.46	0.80	0.44
MSE	0.90	0.34	0.99	0.35
RMSE	0.94	0.58	0.99	0.59
MPE (%)	17.10	11.13	18.49	10.90
Accuracy (%)	82.90	88.87	81.51	89.10

The regression model shows an improvement in crown diameter predictions, bringing accuracy close to 90%. These predictions are essential for practical applications, where precise measurements are critical.

Figure 4-21 presents a comparison between the corrected predicted crown areas and the actual field-measured areas, showing a good agreement between them.

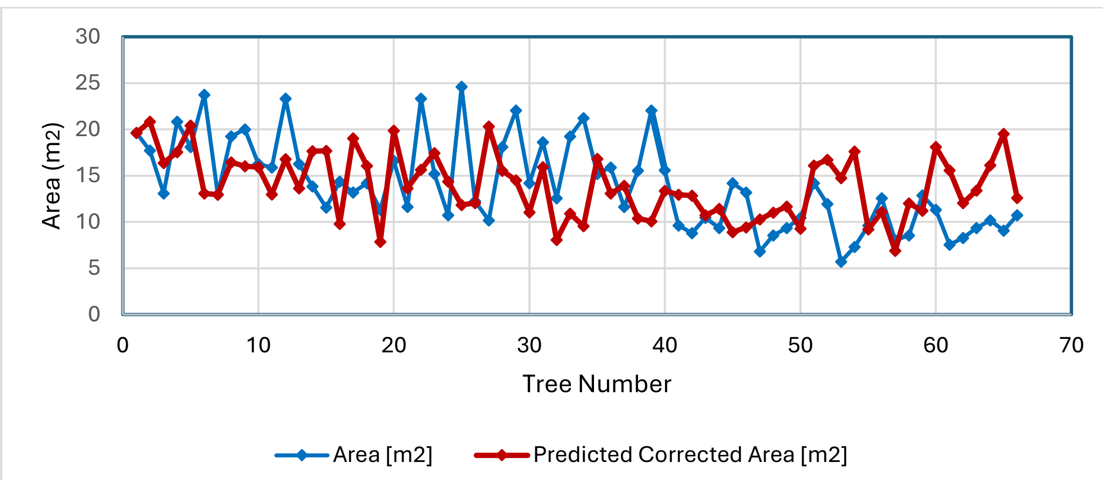

Figure 4-21: Actual vs. Corrected predicted area

4.6. Application of Crown Diameter and DBH

The correlation between field measured DBH and crown diameter derived by linear regression, shown in Figure 4-22, is a valuable finding in this study. It enables DBH to be estimated based on crown diameters, which is a helpful tool in forest management. Predicted

DBH can be used in biomass estimation. This method offers a reliable and scalable solution for biomass estimation using aerial imagery, demonstrating the broader applicability of current research's findings.

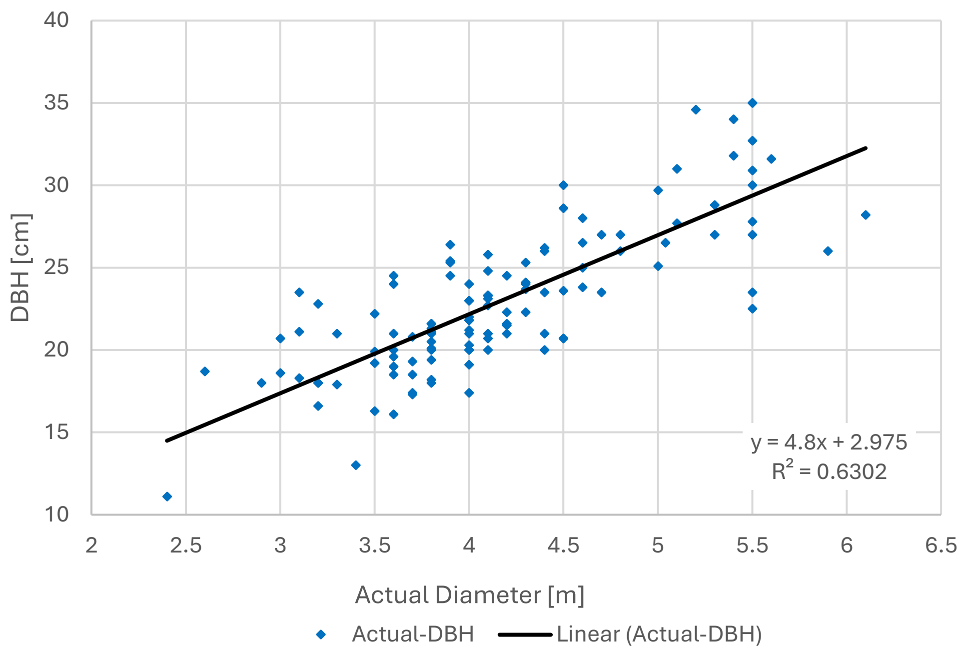

Figure 4-22: Linear relationship between measured DBH and measured crown diameter

4.7. Comparison with Existing Research

This section compares the results of this research with existing studies discussed in the literature review. This study comprises two main parts: using YOLO for detecting the desired trees and applying watershed segmentation for detecting and measuring tree crowns. Each part is evaluated against related research. The comparison focuses on key performance metrics, such as detection accuracy, segmentation precision, and overall efficiency.

Table 4-5 presents the results and methods of three benchmark studies that utilized UAV imagery, comparing them with the two main parts of this research.

Table 4-5: Comparison of benchmark studies on top view drone imagery tree detection and measurement

	Author	Method	Results
Top-View Drone Imagery	Fan et al. (2024)	MCWST	Precision: 52.9% Recall: 34.2%
		Mask R-CNN	Precision: 67.72% Recall: 70.14% F1-Score: 0.67
	Chen et al. (2024)	Att-Mask R-CNN	mAP: 65.29% R^2: 85% rRMSE: 12.81% F1 score: 74.32%
		Mask R-CNN	mAP: 62.41%
		MS R-CNN	mAP: 64.62%
	Fu et al. (2024)	Local Maximum (LM)	Accuracy: 88.27% Recall: 86.31% F1-Score: 0.892
		Fuzzy C-Means Clustering	

For the first part, the YOLOv5-based approach demonstrates superior performance in terms of precision (0.84), recall (0.85), and F1-score (0.84), compared to Fan et al. (2024) [6] using Mask R-CNN, which reported an F1-score of 0.67. Similarly, the results also surpass Chen et al. (2024) [7] using Att-Mask R-CNN, which achieved a slightly higher F1-score of 74.32%. Furthermore, the mAP@0.5 in this research (0.88) significantly exceeds the mAP values of 65.29% and 62.41% reported by Chen et al. (2024). The mAP@0.5:0.95 value of 0.49 in this study, although not directly comparable due to the stricter threshold, this further underscores the robustness of the YOLOv5 model in accurately detecting spruce trees, even in complex environments.

For the second part, focusing on crown detection and measurement using watershed segmentation, the accuracy achieved in this study (89.1%) is on par with the performance reported by Fu et al. (2024) [9], who also achieved an accuracy of 88.27% using Local Maximum (LM) and Fuzzy C-Means Clustering. However, this research additionally reports a Mean Percentage Error (MPE) of 10.9%, providing further insight into the reliability of the crown measurement process. These results confirm the effectiveness of the proposed methodology for both detecting and measuring tree crowns while demonstrating competitive performance compared to existing methods.

The results from this research are also compared with the results of three other benchmark studies, as shown in Table 4-6, that utilized LiDAR datasets.

Table 4-6: Comparison of benchmark studies on LiDAR data
tree detection and measurement

	Author	**Method**	**Results**
LiDAR Data	Liu et al. (2024)	Marker-Controlled Watershed	Recall: 0.854 Precision: 0.937 F1-Score: 0.892
	Zu et al. (2024)	deep learning network	Precision: 78.0% Recall: 80.6% F1-Score: 79.2%.
		Graph-Based ITC Segmentation	
	Yu et al. (2024)	Crown Surface Point Extraction	Precision: 95.9% Recall: 91.6% F1-Score: 94%
		Adaptive Seed Point Detection	

These benchmark studies demonstrate varying performance levels, providing a basis for evaluating the effectiveness of this research. For instance, the precision (0.84), recall (0.85), and F1-score (0.84) achieved by this research are competitive with Liu et al. (2024) [8], who reported an F1-score of 0.892 using Marker-Controlled Watershed. Similarly, this study's precision and recall metrics are comparable to Zu et al. (2024) [11], who achieved an F1-score of 79.2% with a deep learning network combined with Graph-Based ITC Segmentation. While Yu et al. (2024) [10] achieved higher precision (95.9%), recall (91.6%), and F1-score (94%)

using advanced crown surface point extraction techniques, the results of this research highlight the effectiveness of the YOLOv5-based approach in detecting desired trees. These comparisons underscore the robustness of the proposed method, particularly when considering the differences in data types and methodologies used between this research and the benchmark studies.

5.Conclusion

This research introduces a new approach to automatically detect spruce tree crowns and measure their diameters using YOLOv5 and watershed segmentation. The results of this study demonstrate that the YOLOv5 model is highly effective in detecting spruce tree from aerial imagery, achieving a precision of 0.84, a recall of 0.85, and an F1-Score of 0.84 after training for 120 epochs. These metrics highlight the model's robust performance in identifying spruce trees with a balance of accuracy and reliability.

Although some limitations exist during the study, such as detection of overlapping crowns in densely populated areas, results obtained by advanced techniques such as transfer learning and Non-Maximum Suppression (NMS) are satisfactory. Specifically, Table 4-3 illustrates that the inclusion of augmentation techniques further improves model performance, with the highest mean average precision (mAP@0.5) reaching 0.88. These findings validate the effectiveness of the methodology and its applicability to real-world scenarios.

Watershed segmentation plays a crucial role in refining initial bounding boxes by implementation of grayscale, Gaussian blurring, Sobel filtering, and Otsu's thresholding techniques.

The study also employs linear regression after YOLOv5 and watershed segmentation to improve diameter predictions, achieving an overall accuracy of 89.10% compared to field measurements. Additionally, Table 4-5 shows that the corrected predictions align well with actual crown diameters, with a mean absolute error (MAE) of 0.21 meters and a mean percentage error (MPE) of 3.5%, making the dataset more reliable for applications such as biomass estimation and forest management.

6.Future works

Since the dataset is limited to spruce trees, future work should focus on expanding the dataset to include a more diverse range of tree species. By including additional species, the model's ability to handle mixed-species forests will improve, making it applicable to more diverse forestry environments. Increasing the dataset with examples from dense forests, especially those with overlapping crowns, will improve the model's ability to handle the complexities of tree crown detection in high-density areas.

Future efforts should also focus on real-time detection to provide live monitoring of forest and agriculture plant health, allowing faster and more efficient data collection. The implementation of real-time detection can be achieved by utilizing edge computing or onboard drone processors to process data directly during flight. This capability will enable immediate decision-making for forestry management, reducing the delay between data collection and analysis. Additionally, improvements in real-time detection will support applications in emergency situations, such as rapid evaluations after wildfires or pest infestations.

Another area of future research is the integration of LiDAR data with aerial imagery. LiDAR provides detailed three-dimensional information about forest structures, which, when combined with high-resolution aerial images, can significantly improve the accuracy of tree crown and biomass measurements. This combination of data sources could address challenges in dense forests, where overlapping crowns often prevent precise detection.

In addition, future studies should explore methods for creating three-dimensional models of tree crowns from aerial imagery. These 3D models can provide a more complete representation of crown structures, enabling better estimations of biomass and other forestry parameters.

Expanding the time frame of this research is another direction for future work. By integrating seasonal datasets, the model could be adapted for monitoring changes in forest conditions over time. This capability would allow for tracking tree growth, assessing the impacts of environmental challenges, and observing seasonal variations in crown diameter.

In summary, the future works described above aim to develop further the current methodology by addressing its limitations and expanding its applicability to diverse forestry and agricultural contexts. These efforts will not only improve the accuracy and efficiency of tree crown detection and measurement but also contribute to advancing forest monitoring and management technologies on a global scale.

7.References

[1] M. Hassegawa, M. Savard, P. R. N. Lenz, E. Duchateau, N. Gélinas, J. Bousquet, and A. Achim, "White spruce wood quality for lumber products: priority traits and their enhancement through tree improvement," *Forestry: An International Journal of Forest Research,* vol. 93, no. 1, p. 16–37, 2019.

[2] K. K. Gagalova, R. L. Warren, L. Coombe, J. Wong, K. M. Nip, M. M. S. Yuen, J. G. A. Whitehill et al, "Spruce giga-genomes: Structurally similar yet distinctive with differentially expanding gene families and rapidly evolving genes," *The Plant Journal: For Cell and Molecular Biology,* vol. 111, no. 5, p. 1469–1485, 2022.

[3] O. P. Rajora, M. S. Fageria, and M. Fitzsimmons, "Effects of wild forest fires on genetic diversity and population structure of a boreal conifer, white spruce (Picea glauca (Moench) Voss): Implications for genetic resource management and adaptive potential under climate change," *Forests,* vol. 14, no. 1, p. 157, 2023.

[4] H. Pretzsch, P. Biber, E. Uhl, J. Dahlhausen, T. Rötzer, J. Caldentey, T. Koike et al, "Crown Size and Growing Space Requirement of Common Tree Species in Urban Centres, Parks, and Forests," *Urban Forestry & Urban Greening,* vol. 14, no. 3, pp. 466-479, 2015.

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

[6] F. Wen, J. Tian, J. Troles, M. Döllerer, M. Kindu, and T. Knoke, "Comparing Deep Learning and MCWST Approaches for Individual Tree Crown Segmentation," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Vols. X-1–2024, pp. 67-73, 2024.

[7] W. Chen, Z. Guan, and D. Gao, "Att-Mask R-CNN: An Individual Tree Crown Instance Segmentation Method Based on Fused Attention Mechanism," *Canadian Journal of Forest Research,* vol. 54, no. 7, p. 825–838, 2024.

[8] Y. Liu, D. Chen, S. Fu, P. T. Mathiopoulos, M. Sui, J. Na, and J. Peethambaran, "Segmentation of Individual Tree Points by Combining Marker-Controlled Watershed Segmentation and Spectral Clustering Optimization," *Remote Sensing (Basel, Switzerland),* vol. 16, no. 4, pp. 610-, 2024.

[9] Y. Fu, Y. Niu, L. Wang, and W. Li, "Individual-Tree Segmentation from UAV–LiDAR Data Using a Region-Growing Segmentation and Supervoxel-Weighted Fuzzy Clustering Approach," *Remote Sensing (Basel, Switzerland),* vol. 16, no. 4, p. 608, 2024.

[10] J. Yu, L. Lei, and Z. Li, "Individual Tree Segmentation Based on Seed Points Detected by an Adaptive Crown Shaped Algorithm Using UAV-LiDAR Data," *Remote Sensing (Basel, Switzerland),* vol. 16, no. 5, p. 825, 2024.

[11] Y. Zhu, Y. Lin, B. Chen, T. Yun, and X. Wang, "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 (Basel, Switzerland),* vol. 16, no. 15, p. 2807, 2024.

[12] DJI, "DJI Mavic 3 Classic," 2024. [Online]. Available: https://www.dji.com/ca/mavic-3-classic. [Accessed 10 October 2024].

[13] F. J. I. Lamia, J. E. X. Rogi, and D. Tiwow, "Pengukuran Ketajaman Ground Sampling Distance (GSD) Di Berbagai Ketinggian Lahan Sawah Dengan Menggunakan Drone Tipe Mavic 2 Pro Di Desa Matani Kecamatan Tumpaan," *Agri-Sosioekonomi (Manado),* vol. 19, no. 1, 2023.

[14] A. S. Peng, A. Turkmen, B. Eickhoff, M. Finta, and P. Gerads, "Design of a Ground Sampling Distance Graphical User Interface for an Unmanned Aerial Vehicle System," *2019 53rd Annual Conference on Information Sciences and Systems (CISS),* pp. 1-6, 2019.

[15] S. Khare, M. Singh, B. K. Kaushik, and K. Chandra, "Image Processing Technique for Field of View Measurement of Electro-Optical Imaging," *ICOL-2019,* p. 369–372.

[16] T. Riggs, "Simplified Guide to Field of View (FOV Calculation)," Shotkit, 2024. [Online].

[17] J. Kolecki, P. Kuras, E. Pastucha, K. Pyka, and M. Sierka, "Calibration of Industrial Cameras for Aerial Photogrammetric Mapping," *Remote sensing (Basel, Switzerland),* vol. 12, no. 19, p. 3130, 2020.

[18] J. Chave, C. Andalo, S. Brown, M. A. Cairns, J. Q. Chambers, D. Eamus, H. Fölster et al, "Tree Allometry and Improved Estimation of Carbon Stocks and Balance in Tropical Forests," *Oecologia,* vol. 145, no. 1, p. 87–99, 2005.

[19] X. Wu, X. Aijun, and Y. Tingting, "Passive Measurement Method of Tree Height and Crown Diameter Using a Smartphone," *IEEE Access,* vol. 8, p. 11669–11678, 2019.

[20] Z. Zhu, C. Kleinn, and N. Nölke, "Assessing tree crown volume—a review," *Forestry: An International Journal of Forest Research,* vol. 94, no. 1, p. 18–35, 2021.

[21] D. B. Redfern and R. C. Boswell, "Assessment of Crown Condition in Forest Trees: Comparison of Methods, Sources of Variation and Observer Bias," *Forest Ecology and Management,* vol. 188, no. 1, p. 149–160, 2004.

[22] "github," HumanSignal, October 2024. [Online]. Available: https://github.com/HumanSignal/labelImg.

[23] F. Maleki, N. Muthukrishnan, K. Ovens, C. Reinhold, and R. Forghani, "Machine learning algorithm validation: from essentials to advanced applications and implications for regulatory certification and deployment," *Neuroimaging Clinics,* vol. 30, no. 4, pp. 433-445, 2020.

[24] R. Khanam and M. Hussain, "What is YOLOv5: A deep look into the internal features of the popular object detector," no. 10.48550/arxiv.2407.20892, 2024.

[25] J. Redmon, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* p. 779–788, 2016.

[26] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A Forest Fire Detection System Based on Ensemble Learning," *Forests,* vol. 12, no. 2, p. 217, 2021.

[27] Google, "Google Colaboratory," Google, October 2024. [Online]. Available: https://colab.research.google.com/.

[28] M. Hussain, "YOLO-v5 Variant Selection Algorithm Coupled with Representative Augmentations for Modelling Production-Based Variance in Automated Lightweight Pallet Racking Inspection," *Big Data and Cognitive Computing,* vol. 7, no. 2, p. 120, 2023.

[29] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[30] O. G. Ajayi, J. Ashi, and B. Guda, "Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images," *Smart Agricultural Technology,* vol. 5, p. 100231, 2023.

[31] "Watershed Segmentation," [Online]. Available: https://www.sciencedirect.com/topics/computer-science/watershed-segmentation#chapters-articles.

[32] N. Cao and Y. Liu, "High-Noise Grayscale Image Denoising Using an Improved Median Filter for the Adaptive," *Applied Sciences,* vol. 14, no. 2, p. 635, 2024.

[33] T. G. Devi, N. Patil, S. Rai, and C. S. Philipose, "Gaussian Blurring Technique for Detecting and Classifying Acute Lymphoblastic Leukemia Cancer Cells from Microscopic Biopsy Images," *Life (Basel, Switzerland),* vol. 13, no. 2, p. 348, 2023.

[34] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE Journal of Solid-State Circuits,* vol. 23, no. 2, pp. 358-367, 1988.

[35] S. L. Bangare, A. Dubal, P. S. Bangare, and S. Patil, "Reviewing Otsu's method for image thresholding," *International Journal of Applied Engineering Research,* vol. 10, no. 9, pp. 21777-2178310, 2015.

[36] C. J. Kuo, S. F. Odeh, and M. C. Huang, "Image Segmentation with Improved Watershed Algorithm and Its FPGA Implementation," in *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196)*, 2001.

[37] X. Zhou and M. A. Hemstrom, Estimating Aboveground Tree Biomass on Forest Land in the Pacific Northwest: A Comparison of Approaches, 2009.

[38] N. A. Othman, M. U. Salur, M. Karakose, and I. Aydin, "An Embedded Real-Time Object Detection and Measurement of Its Size," 2018.

8. Appendices

8.1. Sample YOLOv5 Code Snippets

YOLOv5

1. Load Data from Google Drive

```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

```

```
Mounted at /content/drive
```

```
# Augmented training dataset
train_image_folder = '/content/drive/My Drive/YOLO/Test1/project_data/split/train/images'
train_label_folder = '/content/drive/My Drive/YOLO/Test1/project_data/split/train/labels'

# Validation dataset)
val_image_folder = '/content/drive/My Drive/YOLO/Test1/project_data/split/val/images'
val_label_folder = '/content/drive/My Drive/YOLO/Test1/project_data/split/val/labels'

# Test dataset
test_image_folder = '/content/drive/My Drive/YOLO/Test1/project_data/split/test/images'
test_label_folder = '/content/drive/My Drive/YOLO/Test1/project_data/split/test/labels'

```

```
import os

# List  in the training dataset
print("Training Images :")
for file_name in os.listdir(train_image_folder):
    print(file_name)

print("\nTraining Labels:")
for file_name in os.listdir(train_label_folder):
    print(file_name)

# List  in the validation dataset
print("\nValidation Images:")
for file_name in os.listdir(val_image_folder):
    print(file_name)

print("\nValidation Labels:")
```

```python
print("\nValidation Labels:")
for file_name in os.listdir(val_label_folder):
    print(file_name)

# List  in the test dataset
print("\nTest Images:")
for file_name in os.listdir(test_image_folder):
    print(file_name)

print("\nTest Labels:")
for file_name in os.listdir(test_label_folder):
    print(file_name)

```

Show hidden output

Split Data into Train, Validation, and Test Sets

```python
import os

# List image and label files for each dataset (train, validation, test)
# Training dataset (augmented)
train_image_files = sorted(os.listdir(train_image_folder))
train_label_files = sorted(os.listdir(train_label_folder))

# Validation dataset
val_image_files = sorted(os.listdir(val_image_folder))
val_label_files = sorted(os.listdir(val_label_folder))

# Test dataset
test_image_files = sorted(os.listdir(test_image_folder))
test_label_files = sorted(os.listdir(test_label_folder))

# Print totals for each dataset
print(f"Total training images: {len(train_image_files)}")
print(f"Total training labels: {len(train_label_files)}")

print(f"Total validation images: {len(val_image_files)}")
print(f"Total validation labels: {len(val_label_files)}")

print(f"Total test images: {len(test_image_files)}")
print(f"Total test labels: {len(test_label_files)}")

# Check if every training image has a corresponding label
print("\nChecking Training Dataset:")
```

```
print("\nChecking Training Dataset:")
for img, lbl in zip(train_image_files, train_label_files):
    print(f"Image: {img} - Label: {lbl}")

# Check if every validation image has a corresponding label
print("\nChecking Validation Dataset:")
for img, lbl in zip(val_image_files, val_label_files):
    print(f"Image: {img} - Label: {lbl}")

# Check if every test image has a corresponding label
print("\nChecking Test Dataset:")
for img, lbl in zip(test_image_files, test_label_files):
    print(f"Image: {img} - Label: {lbl}")


```

Show hidden output

```

#  print the sizes of the already split datasets
print(f"Train size (images): {len(train_image_files)}, Train size (labels): {len(train_label_files)}")
print(f"Validation size (images): {len(val_image_files)}, Validation size (labels): {len(val_label_files)}")
print(f"Test size (images): {len(test_image_files)}, Test size (labels): {len(test_label_files)}")

```

```
Train size (images): 50, Train size (labels): 50
Validation size (images): 11, Validation size (labels): 11
Test size (images): 11, Test size (labels): 11
```

```
# Verify train folder
print(f"Train images: {len(os.listdir(train_image_folder))}, Train labels: {len(os.listdir(train_label_folder))}")

# Verify validation folder
print(f"Validation images: {len(os.listdir(val_image_folder))}, Validation labels: {len(os.listdir(val_label_folder))}")

# Verify test folder
print(f"Test images: {len(os.listdir(test_image_folder))}, Test labels: {len(os.listdir(test_label_folder))}")

```

```
Train images: 50, Train labels: 50
Validation images: 11, Validation labels: 11
Test images: 11, Test labels: 11
```

Set the Architecture of YOLOv5 and Parameters

```
Validation images: 11, Validation labels: 11
Test images: 11, Test labels: 11
```

Set the Architecture of YOLOv5 and Parameters

```
!git clone https://github.com/ultralytics/yolov5
```

Show hidden output

```
%cd yolov5
!pip install -r requirements.txt
```

Show hidden output

```
model_architecture = 'yolov5m'  # You can change this to yolov5s, yolov5l, or yolov5x
```

Extra

```
# New content for data.yaml
data_yaml_content = """
train: /content/drive/MyDrive/YOLO/Test1/project_data/split/train/images
val: /content/drive/MyDrive/YOLO/Test1/project_data/split/val/images
test: /content/drive/MyDrive/YOLO/Test1/project_data/split/test/images

nc: 1  # Number of classes
names: ['tree']  # Class names
"""

# Write the updated content to data.yaml
with open('/content/yolov5/data.yaml', 'w') as file:
    file.write(data_yaml_content)

print("YAML file updated successfully!")
```

```
YAML file updated successfully!
```

```
!cat /content/yolov5/data.yaml
```

```
YAML file updated successfully!
```

```
!cat /content/yolov5/data.yaml
```

Show hidden output

```
epochs = 120  # Number of training epochs
batch_size = 16  # Batch size for training
img_size = 640  # Image size (can adjust to smaller if you have hardware limitations)
```

```
!python train.py --img {img_size} --batch {batch_size} --epochs {epochs} --data data.yaml
    --cfg models/{model_architecture}.yaml --weights 'yolov5s.pt' --name my_yolo_model --project yolov5_project
```

Show hidden output

Train the Model

```
#!zip -r my_yolo_model_results.zip runs/train/my_yolo_model/
!zip -r my_yolo_model_results.zip runs/train/exp/
!zip -r my_yolo_model_results.zip runs/train/
```

Show hidden output

```
from google.colab import files
files.download('my_yolo_model_results.zip')
```

Show hidden output