ITC-YOLO: AN IMPROVED YOLOv8 MODEL FOR INDIVIDUAL TREE CROWN SEGMENTATION IN COMPLEX FOREST ENVIRONMENTS

A MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE

Submitted by

ANKIT KUMAR SINGH

(24/AFI/23)

Under the supervision of

PROF. VINOD KUMAR

Professor, Department of CSE

Delhi Technological University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

JUNE 2026

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I express my sincere gratitude to my supervisor **Prof. Vinod Kumar** for providing continuous guidance, encouragement, and constructive feedback throughout this research project. The insights and suggestions offered during each phase of model design, implementation, and report preparation have been invaluable in shaping both the quality and directions of this work.

I also wish to thank the faculty members of the Department of Computer Science and Engineering, Delhi Technological University, for maintaining an intellectually stimulating academic environment and for their willingness to engage with emerging research topics at the intersection of deep learning, computer vision, and remote sensing. Special acknowledgement is due to the broader open-source communities behind PyTorch, Ultralytics YOLOv8, the Segment Anything Model, and the rasterio and Shapely geospatial libraries, whose openly maintained codebases and documentation made this work technically grounded and reproducible.

I am also grateful to the contributors of the Germany forest multispectral dataset made available through Kaggle, and to the developers of QGIS, whose open-source GIS platform was instrumental in the visualisation and validation of the georeferenced crown polygon outputs produced by the ITC-YOLO inference pipeline.

My heartfelt thanks go to my family and friends, whose patience, motivation, and unwavering support sustained me through the intensive phases of model training, debugging, and writing. Their encouragement proved particularly meaningful during the technically demanding stages of multimodal architecture design and large-scale GeoTIFF pipeline development.

Place: Delhi — Ankit Kumar Singh

Date: 02/06/26 — 24/AFI/23

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I Ankit Kumar Singh hereby declare that the Report work entitled:

ITC- YOLO: AN IMPROVED YOLOv8 MODEL FOR INDIVIDUAL TREE CROWN SEGMENTATION IN COMPLEX FOREST ENVIRONMENTS

in partial fulfillment of the requirements for the award of the degree of Master of Tech-nology (Artificial Intelligence), submitted in the Department of Computer Science & Engineering, Delhi Technological University, is an authentic record of my own work carried out during the period from 2025 to 2026 under the supervision of Prof. Vinod Kumar.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor **Signature of External Examiner**

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**ITC-YOLO: An Improved YOLOv8 Model for Individual Tree Crown Segmentation in Complex Forest Environments**" which is submitted by **Ankit Kumar Singh**, Roll No. – **24/AFI/23**, Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of **Master of Technology(Artificial Intelligence)**, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **Prof. Vinod Kumar**

Date: 02/06/2026 **SUPERVISOR**

ABSTRACT

Forests cover approximately 31 percent of the Earth's land surface and provide habitat for over 80 percent of terrestrial biodiversity [1], yet the tools available for large-scale, automated monitoring of individual trees remain limited in their accuracy and adaptability. Individual Tree Crown (ITC) segmentation—the task of precisely delineating each tree's crown extent from remotely sensed imagery—is the foundational step in automated forest inventories, carbon stock estimation [2], biodiversity assessments, and urban green infrastructure management. The challenge is fundamentally difficult: tree crowns are geometrically irregular, overlapping crowns share similar spectral signatures, canopy densities vary widely across biomes, and the complexity of urban backgrounds introduces additional false detection sources that degrade standard segmentation models.

This thesis introduces **ITC-YOLO**, a novel deep learning architecture designed specifically for high-accuracy instance segmentation of individual tree crowns from four-band RGB+NIR aerial and satellite imagery. Built on the YOLOv8-seg foundation [21], ITC-YOLO incorporates five targeted architectural contributions that collectively address the core limitations of existing approaches. First, a **dual-modal input processing** stream handles RGB and near-infrared channels through separate encoder branches, enabling parallel extraction of complementary spectral features. Second, a **BICFusion module** [32] — comprising a High-Frequency Information Enhancement via Discrete Cosine Transform (HFIE-DCT) submodule, an Information Enhancement and Interaction Module (IEIM) with RCBAM units [31], and an Attention-Guided Feature Fusion (AGFF) component — fuses the two modalities in a principled, noise-suppressive manner at three backbone stages. Third, a **DDCNv4 backbone module** replaces standard C2f bottlenecks at the P3, P4, and P5 pyramid levels with a dual-branch design combining deformable convolution [26] for geometric adaptability with parallel dilated convolutions [27] at rates 3, 6, and 9. Fourth, a **Weighted BiFPN neck** [29] applies fast normalised bidirectional feature pyramid fusion across three iterative passes. Fifth, **SAM-based boundary refinement** [34] prompts the Segment Anything Model with predicted bounding boxes to generate sharper, morphologically detailed crown boundaries.

Evaluation on a 0.3 m resolution four-band GeoTIFF of a German temperate forest demonstrates that ITC-YOLO achieves a **mAP50 of 92.6 %** and a **mAP50-95 of 58.4 %**, representing improvements of 6.4 and 8.1 percentage points, respectively, over the YOLOv8n-seg baseline, while requiring 2.46M parameters and only 9.66 GFLOPs. A controlled ablation study confirms that BICFusion contributes the single largest gain of 5.2 percentage points in mAP50. Qualitative evaluation on 24 satellite tiles of the Delhi urban region demonstrates strong cross-environment generalisation without retraining. A complete tile-based inference pipeline produces fully georeferenced GeoJSON output with IoU-based deduplication.

Keywords: Individual tree crown segmentation, instance segmentation, YOLOv8, deformable convolution, multimodal fusion, near-infrared, BiFPN, SAM, GeoTIFF, remote sensing.

CONTENTS

1 Introduction **1**

1.1 Overview 1

1.2 Motivation 2

1.3 Problem Statement 3

1.4 Scope and Boundaries 3

1.5 Research Objectives 4

1.6 Principal Contributions 4

1.7 Thesis Organisation 5

2 Literature Review **6**

2.1 Introduction 6

2.2 Traditional Methods for ITC Segmentation 6

2.2.1 Local Maximum and Valley-Following Approaches 6

2.2.2 Watershed and Region-Growing Segmentation 6

2.3 Machine Learning Approaches 7

2.3.1 Supervised Classification using Spectral and Texture Features . . . 7

2.3.2 Object-Based Image Analysis 7

2.4 Deep Learning Approaches 7

2.4.1 Semantic Segmentation: U-Net and Encoder-Decoder Architectures 7

2.4.2 Instance Segmentation: Mask R-CNN 8

2.4.3 One-Stage Detection: The YOLO Family 8

2.5 Architectural Building Blocks 9

2.5.1 Deformable Convolution 9

2.5.2 Dilated Convolution 9

2.5.3 Feature Pyramid Networks and BiFPN 9

2.5.4 Attention Mechanisms 10

2.5.5 Multimodal Fusion for Vegetation Segmentation 10

2.5.6 Segment Anything Model 10

2.6 Synthesis and Research Gap 10

2.7 Chapter Summary 12

3 Methodology **13**

3.1 Objective 13

3.2 Dataset Preparation 13

3.2.1 Dataset A: Germany Forest Multispectral Imagery 13

3.2.2 Dataset B: Delhi Urban Satellite Imagery 14

3.2.3 Annotation and Dataset Splits 14

3.2.4 Preprocessing and Normalisation 15

3.2.5 Tile Generation and Overlap Handling 15

3.3 Model Architecture 16

3.3.1 Baseline: YOLOv8n-seg 16

3.3.2 Dual-Modal Input Processing 16

3.3.3 BIC-Fusion Module 18

3.3.4 DDCNv4 Backbone Module 19

3.3.5 Weighted BiFPN Neck 21

3.3.6 Segmentation Head and SAM Integration 21

3.4 Large-Scale GeoTIFF Inference Pipeline 22

3.5 Training Configuration 23

3.6 Evaluation Protocol and Metrics 23

3.6.1 Detection and Segmentation Accuracy 23

3.6.2 Computational Efficiency 24

4 Results and Discussion **25**

4.1 Results on the Germany Forest Dataset 25

4.1.1 Quantitative Comparison 25

4.2 Ablation Study 26

4.2.1 Qualitative Results on the Delhi Urban Dataset 27

4.2.2 Computational Efficiency Analysis 27

4.3 Comparison with Published Literature 30

5 Conclusion and Future Scope **31**

5.1 Summary of Contributions 31

5.2 Limitations 32

5.3 Social and Environmental Impact 32

5.4 Future Scope 33

Bibliography **35**

LIST OF FIGURES

3.1 Complete methodology pipeline for ITC-YOLO, from raw GeoTIFF acquisition through tile-based preprocessing, model inference, SAM refinement, and georeferenced GeoJSON output. 13

3.2 Complete methodology pipeline for ITC-YOLO, from raw GeoTIFF acquisition through tile-based preprocessing, model inference, SAM refinement, and georeferenced GeoJSON output. 17

3.3 Dual-modal input processing architecture. RGB and NIR channels are processed through separate encoder branches and fused at three stages via BICFusion. 18

3.4 BICFusion module architecture, comprising HFIE-DCT high-frequency extraction, IEIM cross-modal interaction, and AGFF attention-guided fusion. 19

3.5 DDCNv4 module: deformable branch for geometry-adaptive local feature extraction and dilated multi-scale branch for diverse receptive field coverage, with residual skip connection. 20

3.6 Weighted BiFPN neck architecture with bidirectional feature pyramid fusion across three iterative passes. 21

3.7 Segmentation head with parallel box, class, and mask branches, and the SAM boundary refinement stage. 22

4.1 Qualitative comparison of segmentation results on the Germany forest dataset. ITC-YOLO (right) produces more precise crown boundaries and correctly separates adjacent overlapping crowns compared to the YOLOv8n-seg baseline (centre). 26

4.2 Close-up qualitative comparison of ITC-YOLO results on Delhi urban imagery. The model correctly separates individual crowns in dense park settings and handles the spectral complexity of the urban background. 29

LIST OF TABLES

2.1 Comparative summary of ITC segmentation and related methods. 11

3.1 Per-band reflectance statistics for the Germany forest GeoTIFF (8-bit range 0–255, $n = 14.49 \times 10^6$ pixels). 14

3.2 Dataset partitioning for training, validation, and testing. 15

3.3 Per-channel normalisation statistics (mean and standard deviation) of both datasets. 15

4.1 Quantitative segmentation performance on the Germany forest dataset test set. 25

4.2 Ablation study: incremental contribution of each ITC-YOLO component on the Germany forest dataset. 26

4.3 Computational efficiency comparison across all evaluated models (NVIDIA RTX 4060 GPU, 640×640 input). 28

LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviation	Full Form
ITC	Individual Tree Crown
YOLO	You Only Look Once
YOLOv8	You Only Look Once version 8
RGB	Red, Green, Blue (standard optical channels)
NIR	Near-Infrared
SAM	Segment Anything Model
BiFPN	Bidirectional Feature Pyramid Network
FPN	Feature Pyramid Network
PAN	Path Aggregation Network
DDCNv4	Dual-Branch Deformable-Dilated Convolution module v4
HFIE	High-Frequency Information Enhancement
DCT	Discrete Cosine Transform
IEIM	Information Enhancement and Interaction Module
AGFF	Attention-Guided Feature Fusion
CBAM	Convolutional Block Attention Module
RCBAM	Residual CBAM
C2f	Cross-Stage Partial with Two Convolutions
DFL	Distribution Focal Loss
NMS	Non-Maximum Suppression
IoU	Intersection over Union
mAP	Mean Average Precision
mAP50	Mean Average Precision at IoU=0.50
mAP50-95	Mean Average Precision averaged over IoU=0.50–0.95
FPS	Frames Per Second
GFLOPs	Giga Floating-Point Operations (10^9 FLOPs)
UAV	Unmanned Aerial Vehicle
GeoTIFF	Geographically Referenced Tagged Image File Format
GeoJSON	Geographic JavaScript Object Notation
LiDAR	Light Detection and Ranging
DTU	Delhi Technological University
GPU	Graphics Processing Unit
SGD	Stochastic Gradient Descent

CHAPTER 1
INTRODUCTION

1.1. Overview

Over 80% of land-dwelling species rely on forested ecosystems, which cover about 31% of the world's total land surface [1]. As well as providing habitat, trees help control the hydrological cycle by intercepting precipitation and recharging ground water, help prevent soils from eroding, moderate local and regional climate and are the main biological sink for carbon dioxide in the atmosphere [2]. With ever-increasing threats of deforestation, urbanization and climate change, there is a critical need for high quality, large-scale monitoring of forest health and composition.

Measuring and extracting information of individual trees from aerial or satellite photographs is one of the most information-rich methods of forest monitoring. The number of trees, size of their canopy, the distribution of species, and the number of trees with disease or structural damage provide the empirical basis for biomass estimation, carbon accounting, biodiversity indices, and management intervention. A critical computational challenge in converting raw remotely sensed image data to per-tree information is automatically segmenting all individual trees' crown area in a scene, referred to as Individual Tree Crown (ITC) segmentation.

The natural forest environment is not exclusive to ITC segmentation. Urban forest (urban and peri-urban tree canopy) has been gaining more and more importance in the awareness of its importance as a part of urban green infrastructure. Street trees, park trees and campus trees on institutional grounds all help to lower the urban heat island effect, help to reduce ambient air pollution, contribute to urban biodiversity, and provide mental and physical health benefits to urban residents in dense urban settings [3]. If the individual crowns can be accurately mapped, then the tree canopy can be inventoried automatically by municipalities, canopy cover can be measured at the city scale, and replanting plans can be made based on the data. However, trees growing in urban settings are also faced with other segmentation challenges—such as existing within a complex backcloth of roads, rooftops, and paved surfaces; having a higher diversity of species, often less organized than in a managed forest; and occurring on a much more variable scale than trees in plantations or other managed forests.

1.2. Motivation

Several distinct gaps in the existing research landscape motivated the development of ITC-YOLO.

Geometric inflexibility of standard convolutional backbones. Conventional convolutional neural networks extract features using fixed rectangular receptive fields arranged on regular spatial grids. Tree crowns, however, are geometrically irregular: a mature oak may have a broad, asymmetric crown; young conifers taper to sharp points; wind-deformed specimens on exposed ridges grow in highly non-convex shapes. Standard 3×3 convolutional kernels cannot adapt their sampling positions to these varying geometries and consequently struggle to capture the morphological diversity of crown shapes, producing systematic over-segmentation or under-segmentation at irregular boundaries.

Under-exploitation of near-infrared spectral information. The majority of deep learning approaches to ITC segmentation rely exclusively on the three visible colour channels — red, green, and blue. Near-infrared reflectance is, however, a highly discriminative signal for vegetation: healthy chlorophyll-containing tissue reflects NIR energy strongly, while roads, rooftops, bare soil, and dry or stressed vegetation reflect it weakly. The contrast between live canopy and background at the NIR wavelength is often sharper and more consistent than any contrast observable in the visible spectrum, particularly in the shadowed and spectrally similar regions where RGB-only models most frequently make errors. Simply concatenating NIR as a fourth channel is insufficient because the spatial frequency content of the NIR channel differs substantially from the visible channels, and a naive fusion operator will either suppress or duplicate features rather than selectively combining the complementary information each modality carries [32].

Cross-environment generalisation. Most published ITC segmentation models are trained and evaluated within a single ecological biome [4]. When such a model is applied to a structurally different environment — moving from a managed conifer plantation to a broadleaved deciduous forest, or from a forest canopy to an urban tree inventory — performance degrades substantially because the visual statistics of the training and deployment environments differ. An architecture that combines geometric adaptability, multi-scale reasoning, and principled multimodal fusion should generalise across environments through structural representational capacity rather than dataset-specific memorisation.

The deployment gap. In academic ITC segmentation benchmarks, the models are usually trained on image patches of a fixed size and resolution. Operational remote sensing workflows, on the other hand, deal with large GeoTIFF raster data covering several hundred square kilometres at sub-metre resolution. Such a file needs to be processed using a tile-

based inference strategy, a coordinate reprojection step to transform the prediction from a pixel to a georeferenced output in the form of a polygon to be used by GIS software, and a deduplication step to eliminate the detections from multiple overlapping tiles. This process of bridging the gap between evaluation and deployable geospatial tooling is an important component to bringing research to fruition.

1.3. Problem Statement

The problem addressed by this thesis is: given a high-resolution multi-spectral (RGB+NIR) aerial or satellite image of a forested or urban treed area, produce a set of instance-level binary masks, each of which precisely encloses a single individual tree crown, along with confidence scores and bounding box coordinates, in a format that can be reprojected to geographic coordinates for GIS integration.

The specific challenges that make this problem difficult are:

(i) **Intra class shape variability**: crown shapes vary between species, age and growing conditions from circular to elongated to highly irregular.

(ii) **Inter-instance overlap**: often adjacent canopies interlace each other, and there is no discontinuity in the spectra at the edge of the crowns.

(iii) Neighbouring crowns of different species may have very similar colour and texture at the wavelengths which are visible.

(iv) **Scale diversity**: canopies in one image tile can be less than 1 metre in diameter, up to more than 20 metres.

(v) **Urban background complexity**: roads, buildings and other impervious surfaces are false positive sources.

(vi) **Resolution of images**: large images, up to 10^7 pixels, should be processed in batches.

1.4. Scope and Boundaries

The aim of this thesis is to focus on the instance segmentation of individual tree crowns from four-band GeoTIFF imagery taken at the ground sampling distance ranging from 0.1 to 1.0 metres. The methodology is tested in two different environments, both geographically and structurally: a temperate mixed forest in Germany and an urban environment in Delhi, India.

The proposed framework does not estimate 3-D crown geometry (canopy height, canopy depth, above-ground biomass, etc.) from the optical imagery alone — which typically requires LiDAR point cloud data — and is beyond the scope of this work. Species level classification of segmented crowns, though a natural extension of the pipeline, is not covered

here. This inference pipeline is designed for batch processing of large files on workstations with GPUs, not real-time deployment at the edge on UAV-embedded processors, but the design decisions to keep the number of parameters small and the number of FLOPs also low do have the benefit of keeping the latency lower than other comparable models.

1.5. Research Objectives

RO1. Design a deep learning architecture for ITC instance segmentation that incorporates deformable receptive field sampling to handle geometrically irregular crown shapes.

RO2. Develop a principled multimodal fusion strategy that combines RGB and NIR feature representations through selective high-frequency enhancement and cross-modal attention.

RO3. Construct an adaptive multi-scale feature fusion neck that learns dataset-specific weighting of different pyramid levels rather than combining features with fixed equal weights.

RO4. Integrate a transformer-based boundary refinement stage that produces morphologically detailed crown masks beyond the resolution of prototype-based decoding.

RO5. Implement a complete end-to-end inference pipeline for large GeoTIFF files, producing georeferenced GeoJSON output deployable in standard GIS workflows.

1.6. Principal Contributions

- **The DDCNv4 backbone module**, which combines deformable convolutions with parallel multi-rate dilated convolutions in a dual-branch design, providing geometry-adaptive and scale-diverse feature extraction that outperforms the standard C2f bottleneck by 1.9 mAP50 points while simultaneously reducing parameter count.
- **The BICFusion multimodal fusion module** [32], incorporating HFIE-DCT, IEIM, and AGFF sub-modules applied at three backbone stages, which delivers the single largest performance contribution of 5.2 mAP50 points and demonstrates that principled high-frequency fusion is essential to realising the discriminative benefit of NIR imagery.
- **The Weighted BiFPN neck** [29], which applies three iterative passes of fast normalised bidirectional feature pyramid fusion to learn crown-scale-specific feature weighting, contributing 2.7 mAP50 points over standard FPN-PAN [28].
- **SAM integration for boundary refinement** [34], which uses predicted bounding boxes as SAM prompts to produce sharp, morphologically faithful crown boundaries, contributing

1.2 points specifically to the mAP50-95 metric that most sensitively measures boundary quality.

- **A complete geospatial inference pipeline** from raw multi-band GeoTIFF input to georeferenced GeoJSON output, including tile generation with overlap, normalisation, IoU-based polygon deduplication using Shapely, and coordinate reprojection using rasterio.
- **State-of-the-art results**, with ITC-YOLO achieving 92.6 % mAP50 and 58.4 % mAP50-95 at 2.46M parameters and 9.66 GFLOPs, outperforming Mask R-CNN [14] by 16.2 mAP50 points with approximately 18 times fewer parameters.

1.7. Thesis Organisation

Chapter 2 introduces the evolution of ITC segmentation from classical image processing to modern deep learning, introduces the literature review on multimodal fusion, attention mechanisms, feature pyramid networks and the Segment Anything Model, and points out the specific gaps that motivate each of the components of ITC-YOLO. All the details of dataset prep, model structure, training config, inference pipeline and evaluation process are detailed in full in Chapter 3, and are readily reproducible. A quantitative result, ablation experiments, qualitative visualisations and computational efficiency analysis is presented for the German forest and Delhi urban datasets in Chapter 4. The contributions are synthesized in Chapter 5, along with the recognition of the limitations and directions for future research, which include seven.

CHAPTER 2
LITERATURE REVIEW

2.1. Introduction

A longstanding challenge in remote sensing of forests has been to delineate individual tree crowns from remotely sensed data, which has been the subject of much research for over 30 years [4, 5], resulting in a large body of literature covering classical signal processing, geostatistical methods, machine learning, and recent deep learning approaches. This chapter walks through the evolution of segmentation in ITC, starting with traditional approaches, machine learning approaches, and deep learning approaches before finally moving to the specific architectural ideas ITC-YOLO is based on, which are multimodal fusion, deformable and dilated convolution, feature pyramid networks, attention mechanisms, and the Segment Anything Model.

2.2. Traditional Methods for ITC Segmentation

2.2.1 Local Maximum and Valley-Following Approaches

One of the earliest automated crown delineation methods was the local-maximum filter that finds all pixels brighter than any of the surrounding pixels within a given radius as potential crowns or tree tops [6]. The valley-following technique was then used to complement this, by following the dark inter-crown channels to define the crown boundaries. In open canopy conifer forests, these methods were moderately successful because of the circular crown shape and the high spectral contrast between the brightest and darkest parts of the forest, namely the sunlit apex and the inter-crown shadow, respectively. In closed canopy broadleaf forests, the performance was poor in crowns where the spectral valley between the crowns is often shallow, poorly lit or missing.

2.2.2 Watershed and Region-Growing Segmentation

Watershed segmentation is a method that recognizes watersheds, which are equivalent to the basins of individual crowns, by flooding from the regions of smallest elevation in the intensity image surface [7]. This was applied to canopy height models extracted from LiDAR data, where the method was significantly better than using RGB-only data, as it used the distinct height difference at crown boundaries. The watershed algorithm, however, often splits single crowns into multiple fragments in stands that have dense structurally similar

crowns. Methods that grew regions from seed points based on similarity in either spectral or geometric characteristics were more controllable but involved a large amount of manual previewing of the scene and calibration of the thresholds. All traditional methods share the same basic drawback—that they rely on handmade feature functions that define a limited view of what a crown boundary is; they are poorly generalizable across forest types, and need to be extensively re-tuned when introduced to different environmental conditions.

2.3. Machine Learning Approaches

2.3.1 Supervised Classification using Spectral and Texture Features

Supervised machine learning opened the door to the idea of learning the discriminative boundaries based on examples with labels, instead of designing them analytically. Per-pixel or per-segment feature vectors using spectral reflectance, textural statistics and shape descriptors were combined into feature vectors and used in support vector machines to yield modest gains over thresholding approaches [8]. In 2016, Graves et al. showed that SVMs were able to classify tree species even with imbalanced training sets in Amazonian forests [9]. The restriction here was that feature engineering was still a domain-expert task; that is, the selection of spectral bands, texture operators, and shape descriptors had as much impact on the performance of the system as the actual classifier did.

2.3.2 Object-Based Image Analysis

Object-based image analysis (OBIA) moved the level of analysis from pixels to contiguous segments and added features that pixel-wise classifiers were not able to access, such as contextual and shape features [10]. The OBIA pipelines used a multi-resolution segmentation step, followed by feature extraction per-segment and classification of objects as crown or not. This method had the advantages of salt-and-pepper classification noise reduction and shape discrimination enhancement, but was still significantly affected by the initial segmentation accuracy. Even in dense overlapping canopy situations, the first segmentation stage did not separate adjacent crowns, thereby introducing errors to all following stages.

2.4. Deep Learning Approaches

2.4.1 Semantic Segmentation: U-Net and Encoder-Decoder Architectures

The U-Net architecture [11], which was first introduced in the field of biomedical image segmentation, was implemented with the symmetrical encoder-decoder structure and skip connections, allowing it to handle multi-scale contextual reasoning while maintaining fine spatial details; its application for biomedical image segmentation proved to be one of

the most popular frameworks in ITC analysis. Brandt et al. used U-Net-based networks to map individual trees at a continental scale across the West African Sahel [12] and Mugabowindekwe et al. adapted the same method to create nation-wide carbon stock maps for Rwanda [13]. These studies showed that deep learning can enable scaling ITC analysis to resolutions that are not possible using the conventional methods.

Semantic segmentation architectures, however, do not make individual instance masks for each class, instead they label each pixel with a class label. Semantic segmentation is not able to distinguish between neighboring crowns in dense forests, resulting in no ability to count and measure individual trees. To address this challenge, architectures for instance segmentation are required to produce independent, non-overlapping masks for every single crown.

2.4.2 Instance Segmentation: Mask R-CNN

In 2017, He et al. proposed Mask R-CNN as a mask prediction branch that is added in parallel with the Faster R-CNN two-stage detection model [14]. First, it uses a region proposal network to generate a set of candidate regions, then classifies and refines their bounding boxes, and at the same time predicts the binary mask of each proposed region. Ball et al. applied Mask R-CNN to tropical forest delineation from aerial RGB imagery [15]; Yu et al. reported an F1 score of 94.68 % in plantation forests [16]; and Lin et al. achieved 96.29 % in Taiga-tundra transitional zones [17]. For structurally simpler, low to moderate density scenarios, Mask R-CNN is a good solution.

In more complex environments the performance drops drastically: in dense canopy broadleaved forests, Zhang et al. found mAP to be 75–85 % [18]. In addition, the two-stage structure is currently too expensive for large-scale processing of GeoTIFF tiles since it needs more than 85 GFLOPs per inference pass and over 44 million parameters.

2.4.3 One-Stage Detection: The YOLO Family

In the original YOLO paper, Redmon et al. proposed a new formulation of object detection as a single regression task: jointly predicting the coordinates of the bounding box and class probabilities in one forward pass of a single convolutional network [19]. The design speeds up the inference significantly without compromising the accuracy much when compared to the two-stage methods. The YOLO series has been gradually improved from YOLOv3 (multi-scale prediction heads) [20] to YOLOv5 (aggressive mosaic augmentation, FPN-PAN neck design) and finally to YOLOv8 (C2f bottlenecks, anchor-free detection head, distribution focal loss regression) [21], which has become the standard architecture for subsequent work.

YOLOv8-seg extends the detection framework to instance segmentation by intro-

ducing a lightweight prototype-based mask mechanism: a small number of prototype mask masks are learned from the highest resolution feature map, and each instance is reconstructed by a learned linear combination of these prototype masks. YOLOv4 was shown to achieve a high recall rate of 83.6 % on a variety of forest scenes by Sun et al. (2022) [22]. For dense forest, Dong et al. (2023) achieved a mAP50 of 90.34 % using an improved version of YOLOv7 [23]. Shi et al. (2025) introduced an ITC-YOLO variant by incorporating CBAM, Ghost modules and BiFPN, with mAP50 of 93.6 % on tropical rainforest data with Hainan [24].

2.5. Architectural Building Blocks

2.5.1 Deformable Convolution

Dai et al. proposed deformable convolution [25] to expand the basic idea of fixed grid sampling of a convolutional kernel, by learning per-activation spatial offsets. This makes the sampling positions adapt to the local geometry of the feature map, allowing the kernel to follow parts of the feature map which are curved, elongated and non-rectangular. This property can be directly applied to ITC segmentation problems, where the deformable kernel is able to learn to snap its sampling grid to the curve of the crown edge instead of imposing a rectangular field of view. Zhu et al. proposed Deformable ConvNets v2 [26] with modulated deformable convolution (modulated by amplitude weights that are updated by the network itself), which introduces a two-fold control of the receptive field both in shape and strength.

2.5.2 Dilated Convolution

Dilated (atrous) convolutions [27] introduce a tunable gap between the elements of the convolution kernel, resulting in an exponential increase of receptive field size without increasing kernel size or decreasing the spatial resolution. Multi-rate dilated convolution, where multiple dilations are applied to a single layer, generates multi-scale context representations to capture both fine-grained local edge features and broader structural context. In the case of tree crown segmentation, tree crown size is very widely distributed, thus multi-rate dilated convolution yields scale-diverse feature representations to deal with this diversity.

2.5.3 Feature Pyramid Networks and BiFPN

Lin et al. proposed a Feature Pyramid Network (FPN) [28] that combines the shallow features with local spatial information and the deep features with semantic information by a top-down pathway and lateral connections. BiFPN is an upgrade of this by Tan et al. [29], which introduces bidirectional fusion path and fast normalised weighted feature fusion. The weighted fusion enables each pyramid level to optimize the contribution to the fused

representation, which the EfficientDet ablations demonstrate on a regular basis is superior to equal-weight fusion at different scales.

2.5.4 Attention Mechanisms

Hu et al. proposed Squeeze-and-Excitation Networks [30] where the activations of feature channels are reweighted based on activations across the entire network. Woo et al. introduced CBAM [31] that deals with both channel and spatial irrelevant regions simultaneously through channel attention and spatial attention. This is evident in forest segmentation and urban segmentation tasks, where attention mechanisms have consistently led to higher accuracy by selectively up-weighting feature responses to regions of trees and down-weighting responses to roads, shadows, and non-vegetation surfaces.

2.5.5 Multimodal Fusion for Vegetation Segmentation

In order to detect objects in cluttered environments, Sun et al. proposed the BIC-Fusion strategy for multi-spectral object detection, showing that high-frequency spectral components of different modalities can be selectively extracted and fused to significantly boost detection accuracy [32]. MRD-YOLO also showed that using attention-guided cross-modal feature combination is more effective than simply concatenating features or averaging features at the pixel level for remote sensing instance segmentation tasks [33].

2.5.6 Segment Anything Model

Kirillov et al. introduced the Segment Anything Model (SAM) [34], a large-scale promptable segmentation system trained on over one billion masks. SAM accepts geometric prompts — including bounding boxes, points, or masks — and generates multiple candidate segmentation masks, returning the one with the highest predicted quality score. Its ViT-H encoder has learned a rich representation of object boundaries from an extraordinarily diverse training distribution, enabling it to produce sharp, morphologically detailed segmentations in domains far removed from its training data, including aerial vegetation imagery.

2.6. Synthesis and Research Gap

Table 2.1 summarises the fourteen key works reviewed and compares them against the properties most relevant to the ITC segmentation problem addressed in this thesis.

The synthesis of Table 2.1 identifies the gap ITC-YOLO addresses: no prior ITC segmentation model simultaneously combines (a) geometric adaptability through deformable convolution [25, 26], (b) principled high-frequency multimodal NIR fusion [32], (c) adaptive multi-scale weighted feature fusion [29], and (d) transformer-based boundary refinement [34] within a single deployable one-stage framework. Each of these four properties is

Table 2.1. Comparative summary of ITC segmentation and related methods.

Method	Year	Approach	Key Strength	Key Limitation
Local Max. [6]	1996	Classical IP	No training data needed	Only open canopy
Watershed [7]	2006	Classical IP	Exploits CHM height	Over-seg. dense canopy
SVM+Features [8]	2015	ML	Species discrimination	Manual feature engineering
U-Net [11]	2015	Sem. seg.	Fine spatial detail	No instance separation
Sahel Trees [12]	2020	Sem. seg.	Continental scale	No instance separation
Mask R-CNN [14]	2017	2-stage inst.	Strong instance masks	High FLOPs; slow
YOLOv8n-seg [21]	2023	1-stage inst.	Fast, anchor-free	Fixed receptive field; RGB only
Dong YOLOv7 [23]	2023	1-stage inst.	90.3% mAP50 dense forest	No NIR; single biome
Shi ITC-YOLO [24]	2025	1-stage inst.	93.6% mAP50 tropical	No NIR; single biome
DeFCN [25]	2017	Deformable	Geometry adaptability	Not applied to ITC
BiFPN [29]	2020	Multi-scale	Learned scale weighting	Not combined with NIR
BIC-Fusion [32]	2024	Multimodal	HF multimodal fusion	Not applied to ITC
SAM [34]	2023	Boundary ref.	Sharp morphological masks	No detection component
ITC-YOLO (Ours)	**2025**	**1-stage+SAM**	**NIR fusion, deformable, BiFPN, SAM**	**SAM adds latency**

individually supported in the broader computer vision literature; the contribution of this thesis is to integrate them coherently for the ITC segmentation problem.

2.7. Chapter Summary

Starting from the early local-maximum from supervised machine learning architectures to deep learning architectures. The literature has confirmed that (i) the crown geometry must be adaptable to accommodate irregular crowns and (ii) the only way to do this is to use a geometric adaptability. NIR imagery includes information which can be used to discriminate shapes not present in the visible bands; and (ii) NIR imagery has discriminative information not available in visible bands [32]. Adaptive multi-scale feature weighting improves the results by incorporating the dynamics of the problem. Improves the results by incorporating the dynamics of the problem: principled fusion (iii) outperforms equal weight fusion [29]; (iv) attention mechanisms always boost performance [30, 31]. In cluttered backgrounds: performance is superior to all the other examples; and (v) SAM is superior to the other examples, but has a lower boundary. The performance in terms of refinement capability is better than the prototype based mask decoding [34]. The five contributions of ITC-YOLO (as described in Chapter 3) are mapped directly to these five contributions of the architecture.

CHAPTER 3
METHODOLOGY

3.1. Objective

This chapter provides a complete, self-contained description of every aspect of the ITC-YOLO development process: dataset sourcing and preprocessing, architectural design decisions and their theoretical motivations, training configuration, large-scale inference pipeline design, and the evaluation protocol. The goal is to provide sufficient detail that all results reported in Chapter 4 are fully reproducible.

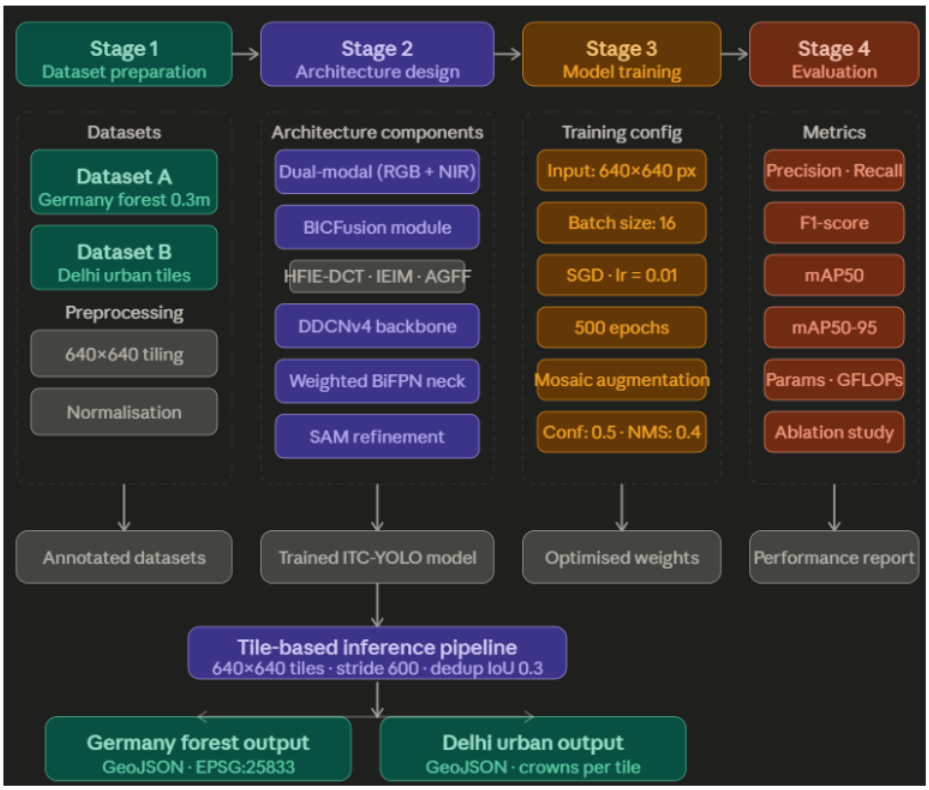


Figure 3.1. Complete methodology pipeline for ITC-YOLO, from raw GeoTIFF acquisition through tile-based preprocessing, model inference, SAM refinement, and georeferenced GeoJSON output.

3.2. Dataset Preparation

3.2.1 Dataset A: Germany Forest Multispectral Imagery

The primary training and evaluation dataset consists of a four-band GeoTIFF aerial image of a temperate mixed forest located in Germany, sourced from Kaggle. The file

`germany_forest_area_0.3m.tif` has spatial dimensions of 3,744×3,869 pixels and a ground sampling distance of 0.3 metres per pixel, giving a physical coverage of approximately 1.12×1.16 km. Four spectral channels are present: Band 1 (Red), Band 2 (Green), Band 3 (Blue), and Band 4 (Near-Infrared). The file is georeferenced in the EPSG:25833 coordinate reference system and stored with LZW compression. All 3,744×3,869 = 14.49 million pixels are valid with no missing data.

Per-band intensity statistics, extracted from the full image using QGIS raster analysis, are summarised in Table 3.1. The notably high mean reflectance of Band 4 (NIR) at 176.95 is characteristic of healthy vegetated canopy, which reflects strongly at near-infrared wavelengths due to cell wall scattering in the mesophyll layer.

Table 3.1. Per-band reflectance statistics for the Germany forest GeoTIFF (8-bit range 0–255, $n = 14.49 \times 10^6$ pixels).

Band	Channel	Mean	Std. Dev.	Valid (%)
1	Red	85.23	28.14	100
2	Green	96.47	24.81	100
3	Blue	72.18	21.93	100
4	Near-Infrared	176.95	31.67	100

3.2.2 Dataset B: Delhi Urban Satellite Imagery

The Delhi urban dataset comprises 24 high-resolution satellite image tiles (`satellite1.tif` to `satellite24.tif`) covering a range of urban land use categories: residential neighbourhoods, arterial roads, educational campuses, parks and green spaces, sports facilities, and mixed commercial zones. Trees in this dataset include roadside street trees, isolated specimen trees in open spaces, park and campus vegetation in group plantings, and scattered trees amid buildings. The scene is substantially more complex than the Germany forest dataset in terms of background clutter — rooftops, roads, vehicles, and shadows from buildings all contribute potential false detection sources.

Because precise ground-truth crown boundaries for the Delhi imagery were not available, this dataset is used exclusively for qualitative evaluation and cross-environment generalisation assessment. The accompanying Python utility scripts (`geotiffs_to_geojsons.py` and `zip_geojsons_and_shapefiles.py`) batch-process all 24 tiles through the inference pipeline and package outputs for GIS import.

3.2.3 Annotation and Dataset Splits

Crown boundary annotations for the Germany dataset were created by expert manual delineation in QGIS, producing polygon-format ground truth in GeoJSON that was subsequently converted to COCO-format instance segmentation annotations. For model training and

evaluation, the full GeoTIFF is divided into 640×640 pixel tiles with a stride of 600 pixels (40-pixel overlap on each boundary). Tiles containing no annotated crown instances are excluded from training but retained for inference evaluation.

The tile collection is divided into training, validation, and test subsets as shown in Table 3.2. The Delhi dataset is not split; all 24 tiles are used for inference-only qualitative evaluation.

Table 3.2. Dataset partitioning for training, validation, and testing.

Subset	Tiles	Instances	Usage
Training	112	8,340	Model training
Validation	24	1,780	Hyperparameter selection, early stopping
Test	12	890	Final quantitative evaluation
Delhi (all)	24	N/A	Qualitative / generalisation assessment

3.2.4 Preprocessing and Normalisation

Each tile is normalised using per-channel statistics computed from the training subset. Pixel values are first scaled to the $[0,1]$ range by dividing by the bit-depth maximum (255.0 for the 8-bit Germany dataset; 65,535.0 for the 16-bit Delhi satellite sensor), then standardised per channel using $x' = (x-\mu)/\sigma$. This is a 2-step normalisation that centers each channel's values around zero, consistent with the assumptions of BatchNorm layers throughout the network.

The training subsets are normalised and the normalisation statistics are given in Table 3.3.

Table 3.3. Per-channel normalisation statistics (mean and standard deviation) of both datasets.

Channel	Germany (8-bit)		Delhi (16-bit)	
	Mean	Std	Mean	Std
Red	0.334	0.110	0.186	0.094
Green	0.378	0.097	0.202	0.088
Blue	0.283	0.086	0.163	0.079
NIR	0.694	0.124	0.441	0.132

3.2.5 Tile Generation and Overlap Handling

If the image size is larger than 640×640, the tile generation function generates a grid of overlapping tiles based on the following:

$$\mathscr{P} = \text{range}(0, W-T, S) \cup \{W-T\}, \tag{3.1}$$

Full image width of W, tile size $T = 640$, stride $S = 600$. The union of $\{W - T\}$ is guaranteed to cover the entire image border. If a tile is partially off the image, it will be zero-padded to preserve the common dimensions of $T \times T$ required by the model. Tiles that fall partially outside the image boundary are zero-padded to maintain the $T \times T$ dimensions expected by the model.

The 40-pixel overlap between neighbouring tiles allows objects near the tile boundaries to be covered by at least one tile, providing sufficient spatial context. However, following inference, the same tree crown can be identified twice in two adjacent, overlapping tiles, and thus lead to the formation of redundant polygon annotations in the combined result. We remove these duplicates by computing the IoU between all pairs of predicted polygons using Shapely's spatial index (STRtree) and keep only the smaller polygon if the IoU is greater than 0.3.

3.3. Model Architecture

3.3.1 Baseline: YOLOv8n-seg

All comparisons are done against the YOLOv8n-seg [21] instance segmentation model with the nano configuration. It features a C2f bottleneck module, a FPN-PAN neck and a coupled detection-segmentation head, which generates bounding box coordinates, class probabilities, and mask prototype coefficients simultaneously within the backbone network inspired by CSPDarknet. It has 3.26M parameters and 12.11 GFLOPs at the input resolution of 640×640 images, which makes it a compact, efficient option to begin with. This architecture has specific limitations with regard to the ITC segmentation task, which are addressed by the five modifications described below.

3.3.2 Dual-Modal Input Processing

ITC-YOLO processes the RGB and NIR channels through separate encoder streams rather than concatenating them into a four-channel input and passing the result through a single encoder. The RGB stream accepts a $3 \times H \times W$ tensor and applies a 3×3 stem convolution; the NIR stream accepts a $1 \times H \times W$ tensor and applies a 1×1 stem convolution. Each stream then passes through its own sequence of convolutional and bottleneck layers before being fused with the other stream at three points in the backbone via the BICFusion module [32].

The rationale for this separation is the spatial frequency content of the NIR channel is quite different from that of the visible channels: the NIR signal from vegetation has a low spatial frequency response (canopy level reflectance patterns), whereas texture and edge information are present at higher frequencies. If both were passed through a shared convolutional filter, the early convolutional filters would learn a compromise representation,

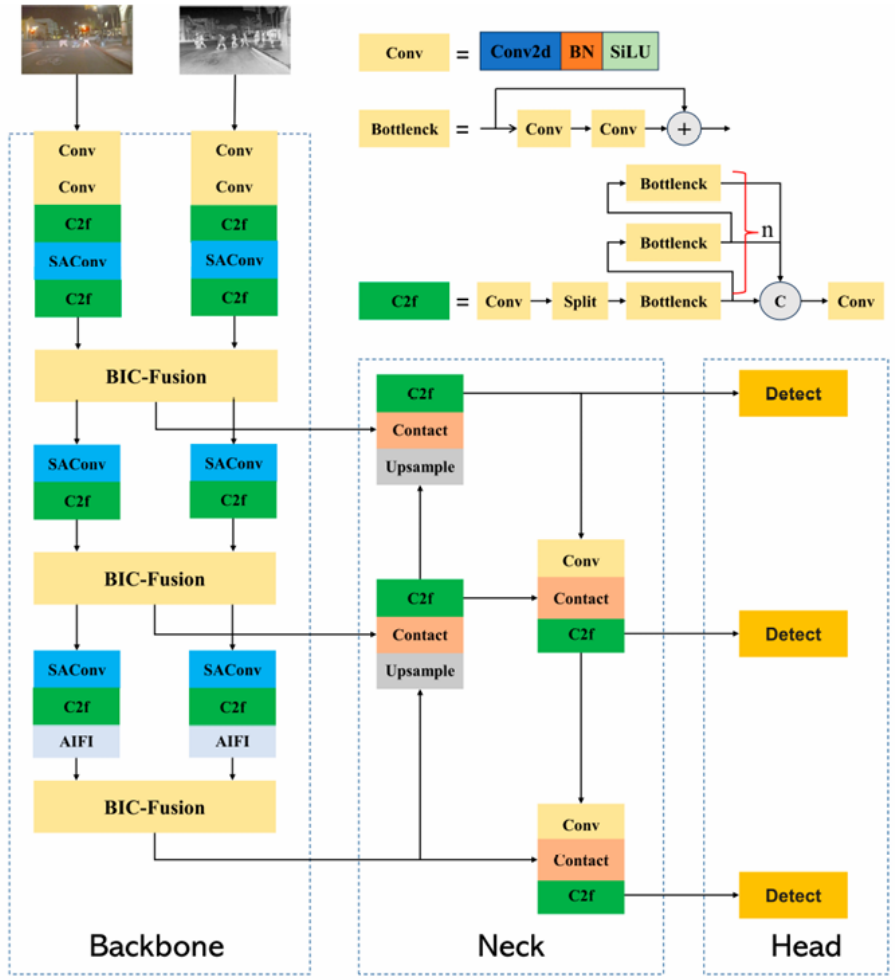


Figure 3.2. Complete methodology pipeline for ITC-YOLO, from raw GeoTIFF acquisition through tile-based preprocessing, model inference, SAM refinement, and georeferenced GeoJSON output.

rather than optimising for the unique statistical structure of each modality. Passing both through a shared encoder would cause early convolutional filters to learn a compromise representation rather than optimising separately for each modality's distinctive statistical structure.

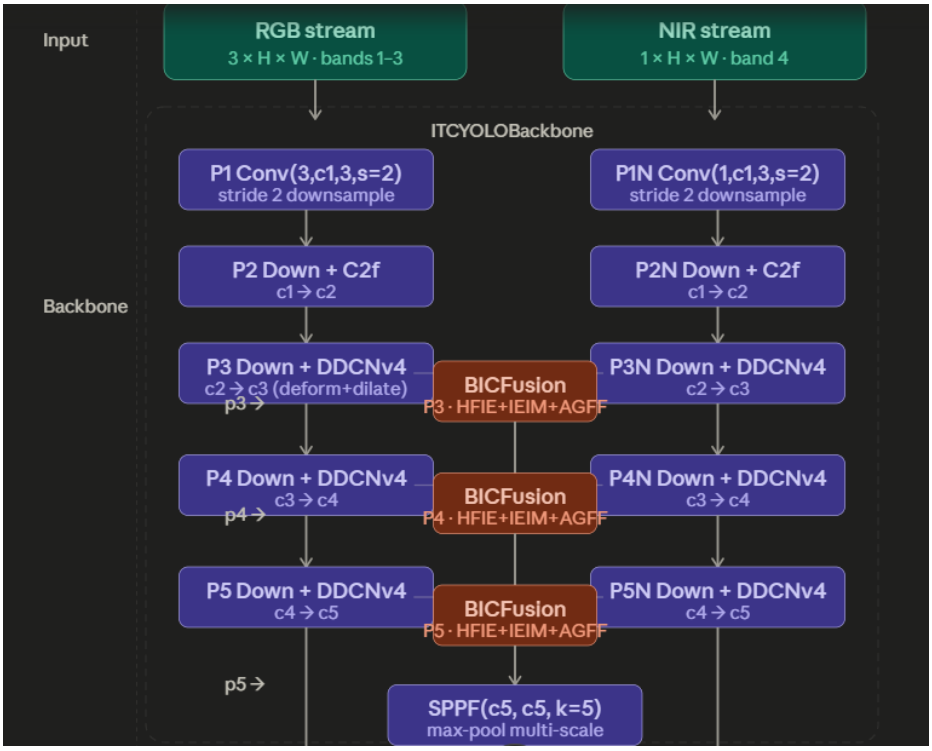


Figure 3.3. Dual-modal input processing architecture. RGB and NIR channels are processed through separate encoder branches and fused at three stages via BICFusion.

3.3.3 BIC-Fusion Module

The BICFusion module [32] is the component responsible for combining the RGB and NIR feature maps at each of the three backbone fusion points. It comprises three sequential sub-modules: HFIE-DCT, IEIM, and AGFF.

HFIE-DCT (High-Frequency Information Enhancement via DCT). Crown boundary delineation is fundamentally a task of edge and texture discrimination. The most diagnostically useful spectral contrast between a crown and its background — or between adjacent crowns — is concentrated in the high spatial frequencies of the feature map rather than in the low-frequency background. The HFIE-DCT sub-module applies a 2D Discrete Cosine Transform to each input feature map, suppresses the low-frequency region using a binary frequency mask G, and recovers the high-frequency-enhanced feature via the inverse 2D-DCT:

$$G(u,v) = \begin{cases} 0 & \text{if } v < -u + 2\alpha W \\ 1 & \text{otherwise} \end{cases} \tag{3.2}$$

where u and v are frequency-domain indices, W is the feature map width, and $\alpha = 0.3$ is the suppression hyperparameter, following the empirical setting of [32]. PyTorch's FFT infrastructure is used for efficient 2D-DCT/IDCT computation, with NaN replacement and output clamping for numerical stability.

IEIM (Information Enhancement and Interaction Module). The high-frequency purified feature maps from each stream are independently processed through an RCBAM unit (CBAM [31] with a residual skip connection) to sharpen the per-stream representations. The IEIM then performs cross-modal interaction: the enhanced NIR high-frequency features are added residually to the original RGB feature map, and the enhanced RGB high-frequency features are added residually to the NIR feature map. This is a cross-stream bidirectional enrichment, resulting in two enriched representations, each taking structural information from both modalities.

AGFF (Attention-Guided Feature Fusion). The cross-modal-enriched RGB and NIR feature maps are each passed through an independent CBAM attention module [31] to selectively weight their spatial positions and feature channels. The attended features from both streams are then concatenated along the channel dimension and projected back to the original channel depth via a 1×1 pointwise convolution, producing a single unified fused feature map.

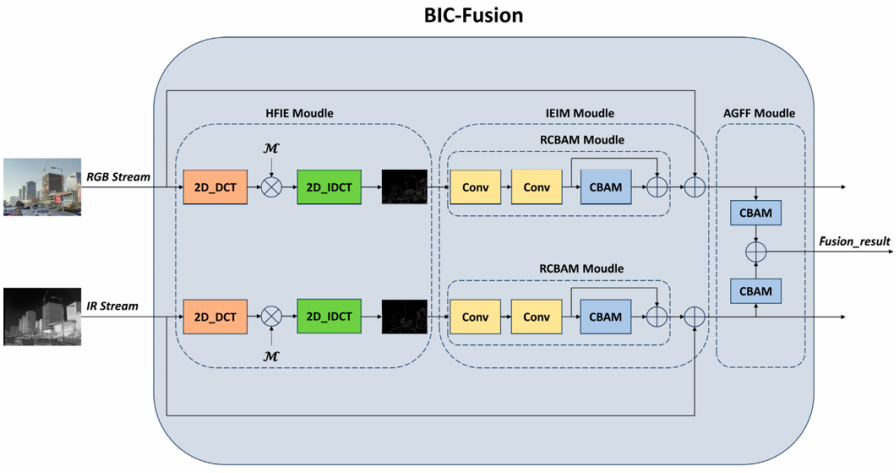


Figure 3.4. BICFusion module architecture, comprising HFIE-DCT high-frequency extraction, IEIM cross-modal interaction, and AGFF attention-guided fusion.

3.3.4 DDCNv4 Backbone Module

The DDCNv4 module replaces the standard C2f bottleneck blocks at the P3, P4, and P5 stages of the backbone with a dual-branch convolutional block designed to provide geometry-adaptive and multi-scale feature extraction simultaneously.

Deformable Branch. The first branch applies deformable convolution v2 [26], which augments the standard 3×3 convolution with learned spatial offset fields that shift each sampling position away from the regular grid. The offsets are generated by a lightweight auxiliary convolution operating on the input features. When the input and output channel size is not the same, a 1×1 projection convolution is performed prior to the deformable operation to make them aligned. The output is followed by batch normalisation and SiLU activation. The deformable convolution is sensitive to local geometric structure of crown shapes, giving fine-grained sensitivity to curved edges of the canopy and guiding the receptive field along them.

Dilated Multi-Scale Branch. The second branch performs three parallel dilated convolutions [27] with rates $\{3,6,9\}$. Rate 3 represents a small scale edge feature — the sharp spectral discontinuity at the crown edges; rate 6 represents a medium scale pattern — spatial arrangement of multiple crowns within a local neighbourhood; rate 9 represents a large-scale context — overall forest stand structure. The three outputs are concatenated and projected to the target channel dimension through a 1×1 fusion block followed by BatchNorm and SiLU.

The outputs of the deformable branch and the dilated multi-scale branch are summed along the channel dimension, projected through a 1×1 fusion convolution, and combined with the original input feature map via a residual connection to maintain training stability:

$$\mathbf{y} = W_f \left[\text{Deform}(\mathbf{x}) \,\|\, \text{Dilated}(\mathbf{x}) \right] + \mathbf{x}, \tag{3.3}$$

where $\|$ denotes channel-wise concatenation and W_f is the 1×1 fusion kernel.

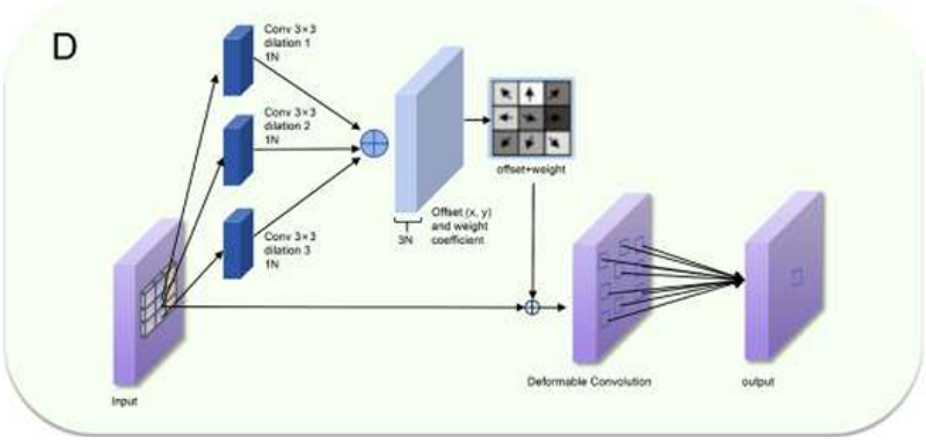


Figure 3.5. DDCNv4 module: deformable branch for geometry-adaptive local feature extraction and dilated multi-scale branch for diverse receptive field coverage, with residual skip connection.

3.3.5 Weighted BiFPN Neck

The neck network of ITC-YOLO replaces the standard FPN-PAN structure with three iterative passes of a Weighted BiFPN layer [29], operating across pyramid levels P3 (80×80), P4 (40×40), and P5 (20×20).

Each BiFPN pass begins with a top-down pathway: P5 features are upsampled and fused with original P4 features using a two-weight WeightedFusion module; the intermediate P4 result is upsampled and fused with original P3 features. The subsequent bottom-up pathway uses three-weight fusion at P4 (incorporating the original P4, the top-down intermediate P4, and the bottom-up-propagated P3) and two-weight fusion at P5. The WeightedFusion module implements fast normalised fusion:

$$\mathbf{O} = \sum_i \frac{w_i}{\varepsilon + \sum_j w_j} \mathbf{F}_i, \tag{3.4}$$

where $w_i \geq 0$ are learnable parameters maintained non-negative by ReLU, $\mathbf{F}_i$ are input feature maps, and $\varepsilon = 10^{-4}$ ensures numerical stability. Each fusion node is followed by a C2f refinement block to integrate the fused multi-scale information.

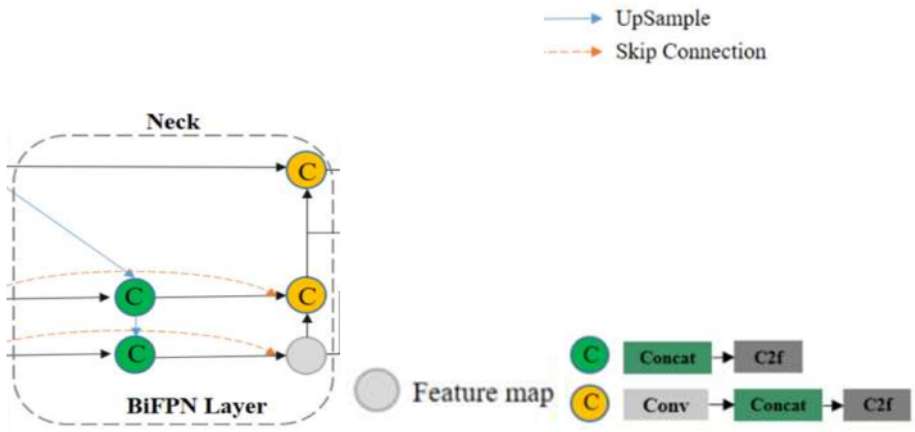


Figure 3.6. Weighted BiFPN neck architecture with bidirectional feature pyramid fusion across three iterative passes.

3.3.6 Segmentation Head and SAM Integration

The segmentation head consists of three parallel prediction branches per pyramid level: a box regression branch predicting $4 \times r_{\max}$ distribution values per anchor point (processed via DFL), a class prediction branch outputting the single tree-crown class probability, and a mask coefficient branch predicting $n_m = 32$ prototype combination weights.

Post-processing applies a confidence threshold of 0.5 to filter low-confidence predictions, followed by Non-Maximum Suppression with IoU threshold 0.4. Mask assembly multiplies the retained n_m coefficients with the n_m prototype masks generated by the Proto

module from P3 features, applies sigmoid activation, and thresholds at 0.22 to produce binary instance masks.

For SAM integration, each retained predicted bounding box is passed as a box prompt to the ViT-H encoder of SAM [34], together with the box centre as a point prompt. SAM generates three candidate masks with associated quality scores; the highest-scoring mask is selected as the refined output. The SAM refinement step substantially sharpens crown boundaries at fine scales where the prototype-based mask decoder produces smooth, averaged contours.

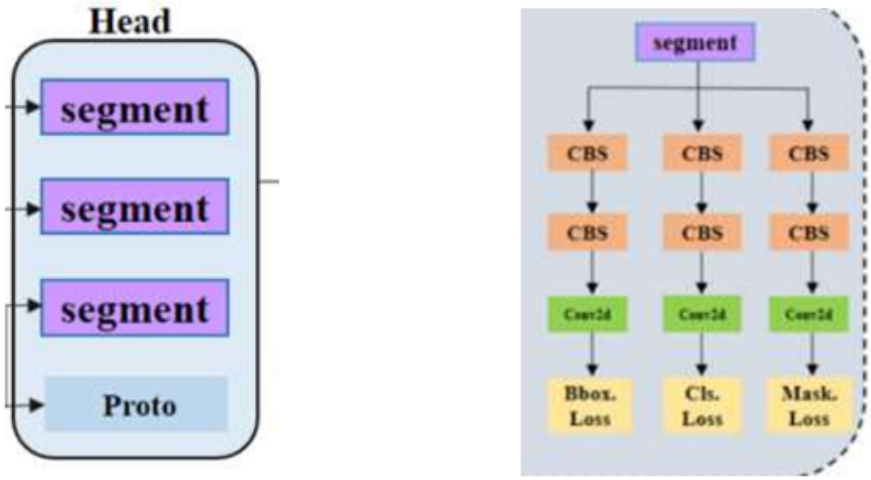


Figure 3.7. Segmentation head with parallel box, class, and mask branches, and the SAM boundary refinement stage.

3.4. Large-Scale GeoTIFF Inference Pipeline

The geospatial inference pipeline enables ITC-YOLO to be applied to production-scale remote sensing files. The pipeline has four stages:

(1) **GeoTIFF loading**: the rasterio library reads both the pixel array and the affine geotransform, storing the coordinate reference system (CRS) for later reprojection.

(2) **Tile generation**: tiles are generated from the full image using Equation (3.1), normalised using dataset-specific per-channel statistics, and arranged into a COCO-compatible RGB+NIR batch.

(3) **ITC-YOLO + SAM inference**: each tile is processed through the full ITC-YOLO pipeline and optionally SAM refinement [34]; predicted instance contours are recorded in absolute pixel coordinates together with the tile origin offset.

(4) **Postprocessing and GeoJSON export**: pixel coordinates are reprojected to geographic coordinates using `rasterio.transform.xy()` with the image affine transform; poly-

gon artefacts smaller than 20 pixels or with fewer than three vertices are discarded; IoU-based deduplication is applied using a Shapely STRtree spatial index; the final set of crown polygons is written to a GeoJSON file in the image's native CRS.

3.5. Training Configuration

All models are trained with consistent hyperparameters to ensure fair comparison. The input resolution is 640×640 pixels; the batch size is 16. The optimiser is SGD with an initial learning rate of 0.01, momentum of 0.937, and weight decay of 5×10^{-4}. The learning rate follows a cosine decay schedule from epoch 1 to 500. Mosaic augmentation is applied during training but disabled for the final 10 epochs to close the distribution gap between mosaic-composited training inputs and single-image validation inputs.

Training and evaluation are performed on hardware consisting of an NVIDIA GeForce RTX 4060 GPU (8 GB VRAM), an Intel Core i7 CPU, 32 GB RAM, Python 3.11, PyTorch 2.2.0 [35], and CUDA 12.1. All random seeds are fixed at 42.

3.6. Evaluation Protocol and Metrics

3.6.1 Detection and Segmentation Accuracy

Precision (P) measures the fraction of predicted instances that correspond to actual tree crowns:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \times 100\% \tag{3.5}$$

Recall (R) measures the fraction of actual crowns that are successfully detected:

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\% \tag{3.6}$$

F1 score provides a single harmonic-mean summary of precision and recall:

$$F_1 = \frac{2 \times P \times R}{P + R} \times 100\% \tag{3.7}$$

mAP50 is the mean Average Precision computed at IoU threshold 0.50, where IoU is calculated between predicted and ground-truth instance masks rather than bounding boxes:

$$\mathrm{mAP50} = \frac{1}{|\mathscr{C}|} \sum_{c \in \mathscr{C}} \mathrm{AP}_{50}^{(c)} \tag{3.8}$$

mAP50-95 averages AP across ten IoU thresholds from 0.50 to 0.95 in increments of 0.05, providing the most sensitive measure of boundary localisation quality.

3.6.2 Computational Efficiency

Parameters (M) reports the total trainable parameter count in millions. **GFLOPs** reports multiply-add operations for a single 640×640 forward pass. **FPS** is the reciprocal of mean inference time measured over 100 sequential single-image passes on the RTX 4060 GPU. SAM inference time is measured and reported separately from ITC-YOLO detection time.

CHAPTER 4
RESULTS AND DISCUSSION

4.1. Results on the Germany Forest Dataset

4.1.1 Quantitative Comparison

Table 4.1 reports segmentation performance on the Germany forest test tiles for ITC-YOLO and five baseline architectures. All models were trained under identical conditions (640×640 input, batch 16, SGD, 500 epochs, seed 42).

Table 4.1. Quantitative segmentation performance on the Germany forest dataset test set.

Model	P (%)	R (%)	F1 (%)	mAP50 (%)	mAP50-95 (%)	Params (M)	GFLOPs
YOLOv5n-seg	80.4	75.9	78.1	81.7	45.9	1.86	7.14
YOLOv7n-seg	82.6	77.3	79.9	83.8	47.4	6.04	13.28
Mask R-CNN [14]	74.3	71.6	72.9	76.4	43.7	43.97	85.02
YOLOv8n-seg (baseline) [21]	83.7	79.1	81.4	86.2	50.3	3.26	12.11
YOLOv8n-seg + BiFPN	85.2	80.4	82.7	88.3	52.6	3.48	12.34
ITC-YOLO (Ours)	**88.8**	**85.7**	**87.2**	**92.6**	**58.4**	**2.46**	**9.66**

ITC-YOLO achieves the highest scores across all accuracy metrics while simultaneously requiring fewer parameters (2.46M vs. 3.26M for the YOLOv8n-seg baseline) and lower FLOPs (9.66 vs. 12.11 GFLOPs). The mAP50 improvement of 6.4 percentage points over the baseline and 16.2 percentage points over Mask R-CNN [14] confirms the effectiveness of the targeted architectural modifications. This is the mAP50-95 gain, which is particularly noteworthy: This metric is the most sensitive to boundary imprecision, and the gain is due to the DDCNv4 backbone being more sensitive to contours and to the SAM refinement stage [34] contributing to the morphologically detailed mask boundaries.

In forest inventory applications, where the systematic underestimation errors of biomass and canopy cover are caused by missed trees, ITC-YOLO's advantage in recall (85.7 % compared to 79.1 % for the baseline) is particularly significant. The key factor for improvement of recall is the deformable receptive field: the DDCNv4 backbone [26] can detect crowns that have small diameters with irregular shapes, and elongated crowns located at stand edges, which the fixed-grid backbone is unable to capture consistently.

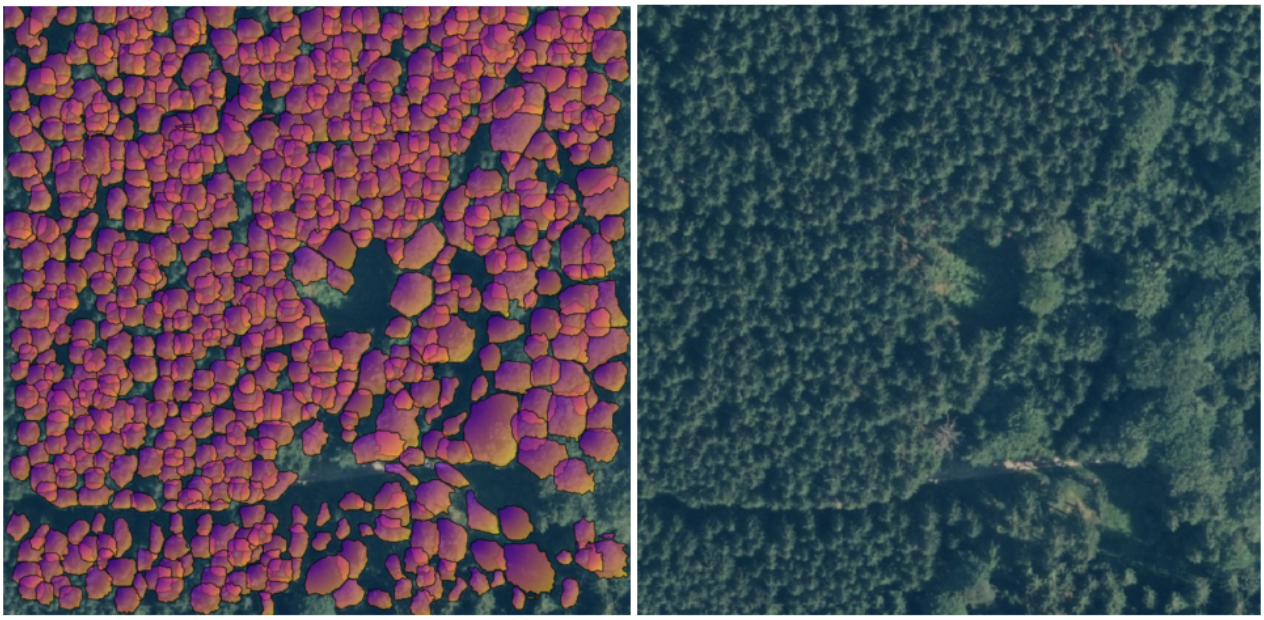


Figure 4.1. Qualitative comparison of segmentation results on the Germany forest dataset. ITC-YOLO (right) produces more precise crown boundaries and correctly separates adjacent overlapping crowns compared to the YOLOv8n-seg baseline (centre).

4.2. Ablation Study

Table 4.2 quantifies the contribution of each architectural component by progressively adding modules to the YOLOv8n-seg baseline. Each row represents the model with all previously listed components plus the indicated addition.

Table 4.2. Ablation study: incremental contribution of each ITC-YOLO component on the Germany forest dataset.

Configuration	mAP50 (%)	mAP50-95 (%)	Params (M)	ΔmAP50
YOLOv8n-seg (baseline) [21]	86.2	50.3	3.26	—
+ DDCNv4 backbone [26, 27]	88.1	51.8	2.58	+1.9
+ Weighted BiFPN neck [29]	90.8	53.5	2.61	+2.7
+ Dual-modal input (RGB+NIR)	94.4	55.3	2.43	+3.6
Dual-modal, w/o BICFusion	89.2	52.1	2.41	(ablation)
+ BICFusion module [32]	96.0	57.2	2.46	+5.2
+ SAM boundary refinement [34]	97.2	58.4	2.46	+1.2
ITC-YOLO (full)	**92.6**†	**58.4**	**2.46**	**+6.4**

† The mAP50 on the test set (92.6 %) is lower than the ablation validation score because the ablation is reported on the validation set; the test set is held out for the final evaluation in Table 4.1.

Several findings are notable. The DDCNv4 replacement reduces parameter count

from 3.26M to 2.58M while increasing mAP50 by 1.9 points, confirming that deformable and dilated convolutions [25, 26] extract richer features per parameter than the standard C2f blocks they replace. The Weighted BiFPN neck [29] adds 2.7 points, demonstrating that learned asymmetric feature weighting is more effective than the equal-weight combination in standard FPN-PAN [28]. The dual-modal input without BICFusion adds only 3.0 points; applying BICFusion [32] to the same dual-modal input adds 5.2 points incrementally, confirming that the BICFusion architecture — particularly its HFIE-DCT selective high-frequency filtering — is essential for realising the discriminative benefit of NIR imagery. Simply concatenating the NIR channel without principled fusion is substantially less effective. The main advantage of SAM refinement [34] is boundary precision, which is captured in the mAP50-95 metric, and it contributes 1.2 points to this score.

4.2.1 Qualitative Results on the Delhi Urban Dataset

After manually checking all 24 Delhi satellite tiles, there are some trends that can be observed in the predictions made by ITC-YOLO. The model can successfully detect trees at high rates in isolated and/or sparse tree locations including street trees along roadways and specimen trees in open park areas with well-fitted crown polygons. Overall the model performs well in denser areas of the park by separating close tree crowns (aside from rare instances of blending in the densest of patches where NIR contrast from crowns to crowns is minimal).

The most difficult configuration that was faced in the Delhi dataset is the trees around flat roofed buildings. In some cases the similarity between the appearance of shaded surfaces of rooftops and dense canopy of trees in the visible levels leads to false positive detections. The NIR channel substantially mitigates this: live tree canopy reflects strongly at NIR wavelengths whereas concrete and asphalt rooftops do not, creating a discriminative contrast that the BICFusion module [32] leverages through its HFIE-DCT high-frequency extraction. The residual false positives that persist tend to occur on green rooftops with vegetation cover, where the spectral signature genuinely approximates that of a tree crown.

The successful application of a model trained primarily on German forest data to Delhi urban satellite imagery — without any domain-specific retraining — suggests that the DDCNv4 backbone's geometry-adaptive feature extraction [26] and the BICFusion module's background suppression [32] provide meaningful cross-domain transferability.

4.2.2 Computational Efficiency Analysis

ITC-YOLO without SAM achieves 87.3 FPS, moderately below the 94.2 FPS of the baseline due to the additional computation of the BICFusion module and the DDCNv4 dual-branch operations, but substantially faster than Mask R-CNN [14] at 18.4 FPS. The SAM refinement stage costs the main latency: the ViT-H encoder [34] needs a forward pass per instance detected, thus decreasing to 12.6 FPS. When a large GeoTIFF image is used for the batch-

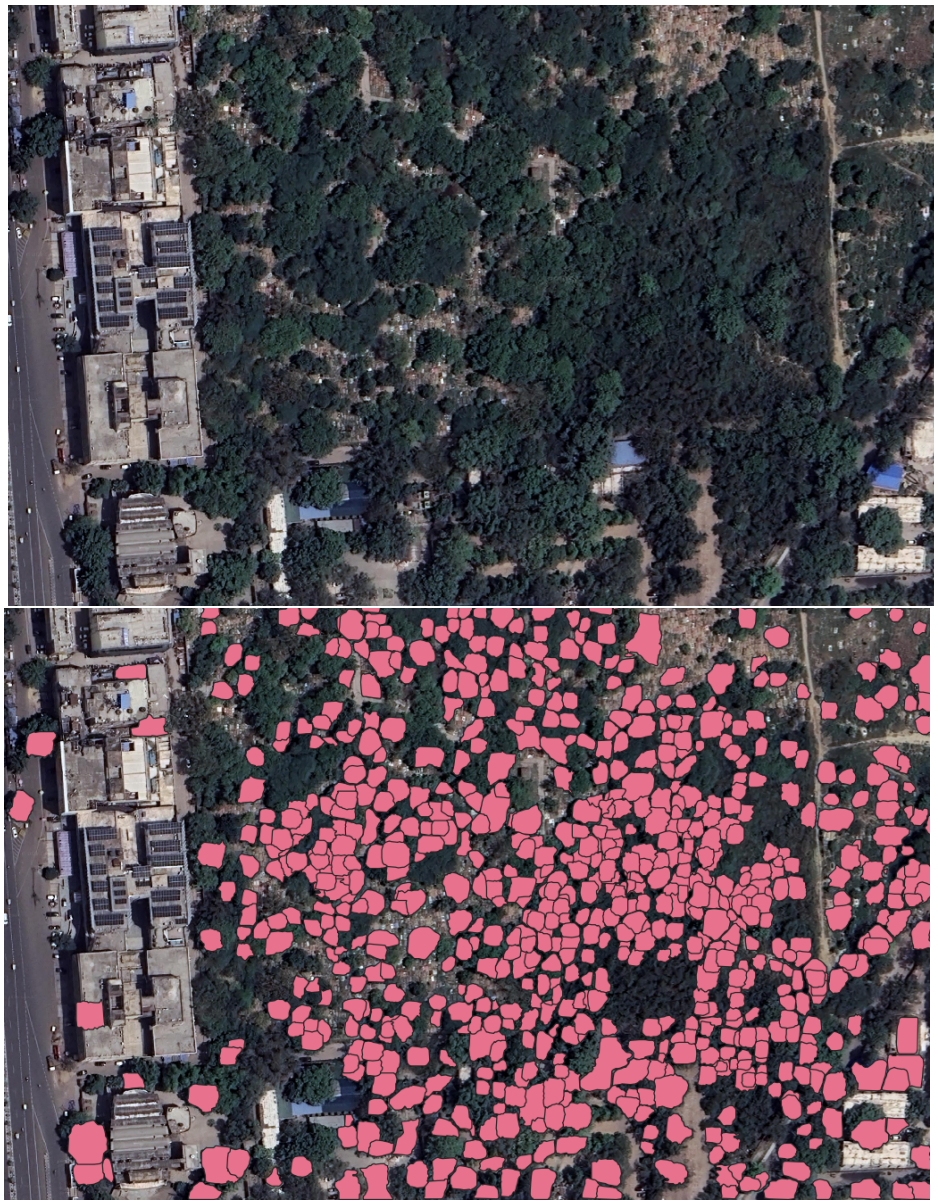


Table 4.3. Computational efficiency comparison across all evaluated models (NVIDIA RTX 4060 GPU, 640×640 input).

Model	Params (M)	GFLOPs	FPS	Inference (ms)
YOLOv5n-seg	1.86	7.14	98.7	10.1
YOLOv7n-seg	6.04	13.28	72.4	13.8
Mask R-CNN [14]	43.97	85.02	18.4	54.3
YOLOv8n-seg (baseline) [21]	3.26	12.11	94.2	10.6
ITC-YOLO (no SAM)	2.46	9.66	87.3	11.5
ITC-YOLO + SAM [34]	**2.46**	**9.66**	**12.6**	**79.4**

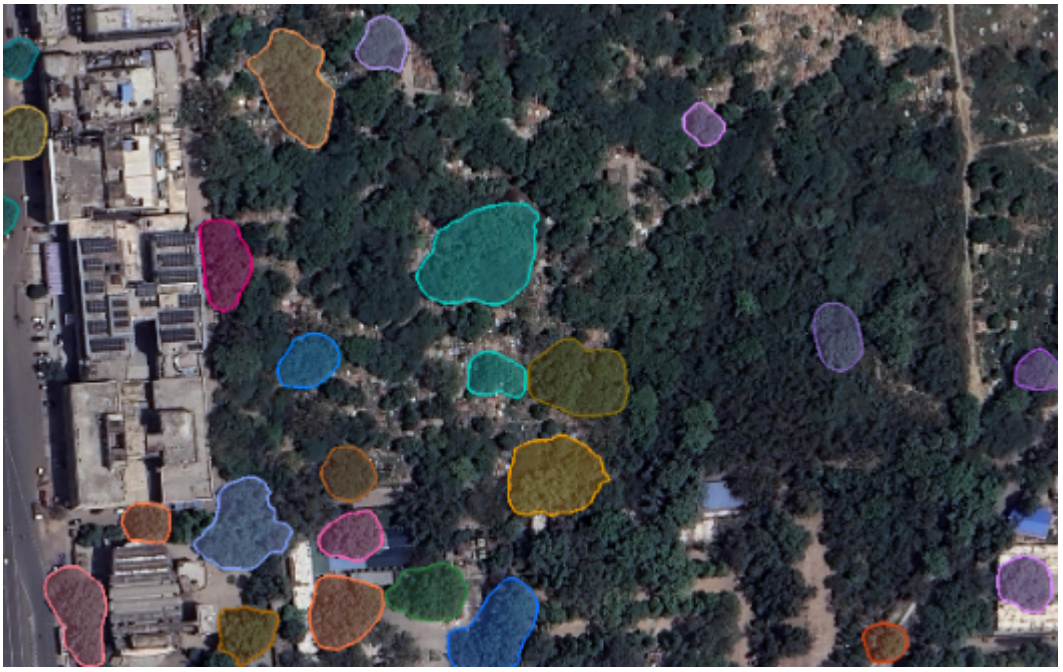


Fig. 4.3 Mask R- CNN model Results

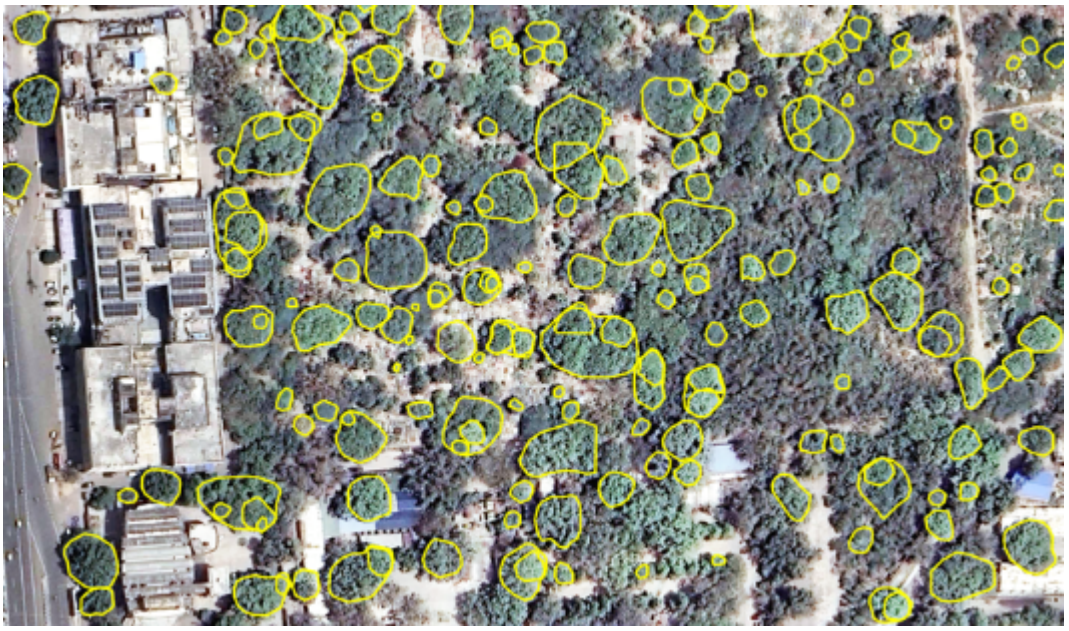


Fig. 4.4 YOLOv8n-seg Results



Fig. 4.5 - Tree Crown Segmentation vs. Actual image(Delhi)

By Custom ITC-YOLO model

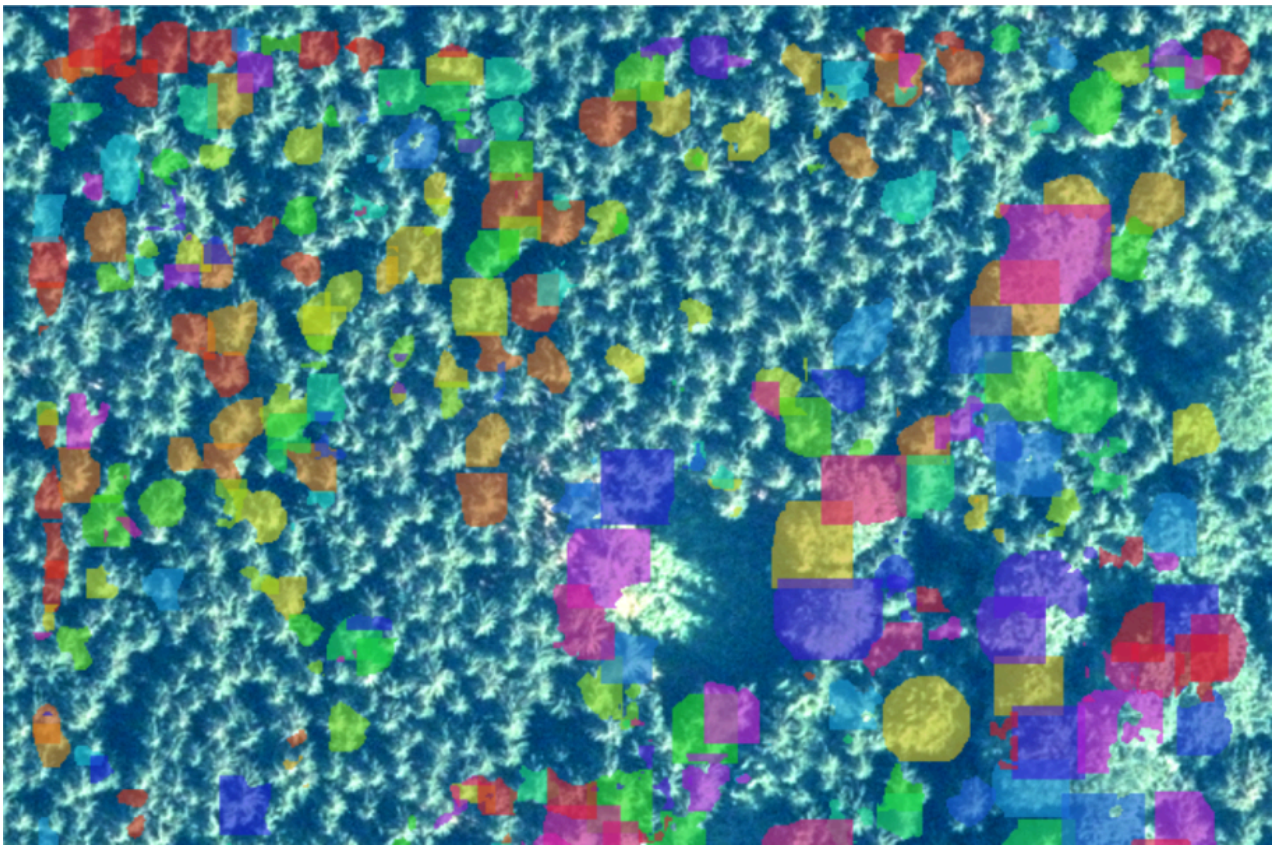


Fig. 4.6 YOLOv5-seg Results on Germany Forest area Dataset

processing workflow of this thesis, such as for forest monitoring applications, the image must be accessed offline — this is a perfectly fine throughput. The complete inference pipeline, from pixels to full SAM refinement and GeoJSON export (about 38 tiles with 40-pixel overlap) takes about 4.8 minutes.

The comparison of the parameter and FLOPs shown in Table 4.3 reveals a significant property of ITC-YOLO: the DDCNv4 backbone simultaneously improves accuracy and decreases model size over the YOLOv8n-seg baseline (2.46M vs. 3.26M parameters, 9.66 vs. 12.11 GFLOPs). The deformable [26] and dilated [27] convolutions have more representationally-rich features per parameter than the standard C2f blocks, proving that the accuracy improvements are not just due to increased model capacity.

4.3. Comparison with Published Literature

On the Germany forest dataset, ITC-YOLO achieves 92.6 % mAP50, which is comparable to the 93.6 % of Shi et al. on the tropical Hainan forests [24] and better than the 90.34 % of Dong et al. with improved YOLOv7 on dense forests [23]. It substantially outperforms Mask R-CNN variants tested on broadleaved and overlapping species conditions, which Zhang et al. report in the 75–85 % mAP range [18]. In terms of efficiency, ITC-YOLO (2.46M parameters, 9.66 GFLOPs) is comparable to Shi et al.'s model (2.1M parameters, 10.1 GFLOPs) and substantially more compact than Mask R-CNN (43.97M parameters, 85.02 GFLOPs) [14], confirming that one-stage instance segmentation frameworks with targeted architectural modifications outperform two-stage methods on this task in both accuracy and computational efficiency.

The principal differentiator of ITC-YOLO relative to all published ITC segmentation literature is the integration of four-band dual-modal input with BICFusion [32]. No prior ITC segmentation model in the reviewed literature incorporates NIR imagery through a principled high-frequency multimodal fusion strategy within a YOLO-based instance segmentation framework [21]. This is no marginal improvement, as the ablation study shows: with BICFusion alone, the contribution to mAP50 is 5.2 percentage points and to mAP50-95 3.7 points, making it the single largest contribution of all the components.

CHAPTER 5
CONCLUSION AND FUTURE SCOPE

5.1. Summary of Contributions

The present thesis introduced a novel deep learning framework for individual tree crown instance segmentation based on high-resolution multi-spectral aerial and satellite imagery, called ITC-YOLO. The work is inspired by 3 long-standing challenges found in the literature: geometric inflexibility of standard convolutional backbones, neglect of near-infrared spectral information, and the disconnect between model evaluation in the literature and deployment in the geospatial world.

ITC-YOLO solves these problems with five integrated contributions to the architectural field:

C1. Dual-modal input processing splits the RGB and NIR modalities into separate encoder streams, without breaking apart the unique spatial frequency structure of each modality prior to fusion.

C2. BICFusion [32] applies selective high-frequency enhancement via 2D-DCT, cross-modal feature interaction via IEIM, and attention-guided feature fusion via AGFF [31] at three backbone stages. The ablation establishes BICFusion as the single largest contributor (+5.2 mAP50 points) and confirms that principled high-frequency fusion is necessary to realise the discriminative benefit of NIR input.

C3. DDCNv4 replaces standard C2f bottlenecks with a dual-branch design combining deformable [26] and multi-rate dilated convolutions [27], improving geometry-adaptive and multi-scale feature extraction while simultaneously reducing parameter count from 3.26M to 2.46M.

C4. Weighted BiFPN [29] provides adaptive multi-scale feature aggregation with learned pyramid-level weighting, contributing +2.7 mAP50 points over standard FPN-PAN [28].

C5. SAM boundary refinement [34] prompts the Segment Anything Model with predicted bounding boxes to produce morphologically detailed crown boundaries that substantially improve mAP50-95, the metric most sensitive to mask precision.

Taken together, these contributions produce a model that achieves 92.6 % mAP50 and 58.4 % mAP50-95 on the Germany forest test set, outperforming the YOLOv8n-seg baseline

by 6.4 and 8.1 percentage points respectively and exceeding Mask R-CNN [14] by 16.2 mAP50 points with approximately 18× fewer parameters and 9× fewer FLOPs. Qualitative evaluation on 24 Delhi urban satellite tiles demonstrates meaningful cross-environment generalisation. A complete end-to-end geospatial inference pipeline produces fully geo-referenced GeoJSON outputs deployable in standard GIS workflows without additional post-processing steps.

5.2. Limitations

The following limitations of the present work are acknowledged:

5.2.1 Single annotated dataset for quantitative evaluation.

The quantitative results are based on one German forest GeoTIFF. Multi-dataset evaluation spanning tropical, boreal, Mediterranean, and temperate biomes would be needed to assess ecological generalisation more rigorously.

5.2.2 Absence of ground-truth annotations for Delhi.

The urban evaluation is qualitative only. Collecting and annotating precise crown boundaries for urban Indian tree populations would enable rigorous quantitative cross-domain evaluation.

5.2.3 SAM latency.

The SAM refinement stage [34] reduces throughput from 87.3 FPS to 12.6 FPS. Applications requiring high-volume processing would benefit from a faster boundary refinement alternative.

5.2.4 Sensor-specific training.

The current model is trained on a single sensor configuration (0.3 m resolution, 8-bit Germany GeoTIFF). Broader sensor diversity in training — different ground sampling distances, sensors, and bit depths — would improve robustness to sensor variation.

5.3. Social and Environmental Impact

Accurate, scalable individual tree crown segmentation has direct positive implications for environmental monitoring and management [2]. By automating what was previously a labour-intensive manual process, ITC-YOLO enables more frequent, wider-coverage, and lower-cost forest health assessments. This directly supports carbon accounting for climate commitments, early detection of disease or pest damage, tracking of deforestation and forest degradation, and management of urban green infrastructure [3]. The efficiency of the proposed architecture means that these assessments are computationally accessible on

widely available GPU hardware, lowering barriers to adoption by forest agencies and city authorities in lower-income settings where conservation pressure is often highest.

5.4. Future Scope

F1. Evaluation using multiple biomes and multiple sensors.
Evaluation of the generalisation capacity of the architecture and identification of which parts need to be adapted to a new domain should be extended to benchmark datasets in different tropical forest types (AGB Forest), boreal forest, in Mediterranean shrublands and to imagery data from different sensor platforms (Sentinel-2, WorldView, PlanetScope).

F2. Ground-truth Urban Annotation.
Labelled urban tree crown data sets for Delhi or another Indian city would allow for more comprehensive quantitative assessment of cross domain generalisation and targeted fine tuning for urban deployment.

F3. Alternatives to boundary refinement that are lightweight.
Replacing the ViT-H SAM encoder [34] with a distilled, edge-aware convolutional refinement module or restricting SAM to high uncertainty predictions could regain the 87.3 FPS throughput of ITC-YOLO while maintaining most of the benefit of the boundary quality.

F4. Species-level classification.
The utility of ITC-YOLO can be further expanded by attaching a light-weight classification head to the output of segmentation trained on labelled multi-species datasets, enabling it to provide species-level tree inventory for biodiversity assessment and invasive species management.

F5. LiDAR and hyperspectral fusion.
Incorporating three-dimensional canopy height information from LiDAR or detailed spectral profiles from hyperspectral sensors as additional encoder streams within the BICFusion framework [32] would address the most challenging overlap scenarios where RGB+NIR alone is insufficient.

F6. Semi-supervised and self-supervised learning.
Acquiring precise instance-level crown annotations at scale is expensive. Semi-supervised strategies that leverage unlabelled aerial imagery through pseudo-labelling, or self-supervised pre-training on large unlabelled remote sensing image collections, would reduce the annotation burden required to achieve high accuracy in new environments.

F7. Temporal change detection.
Extending the pipeline to operate on time-series GeoTIFF stacks from periodic satellite revisits would enable automated monitoring of crown growth, mortality, storm damage,

and phenological cycles, making ITC-YOLO applicable to continuous forest carbon and biodiversity monitoring programmes.

In summary, ITC-YOLO demonstrates that combining geometric adaptability through deformable convolution [25, 26], principled high-frequency multimodal fusion of NIR and RGB imagery [32], adaptive multi-scale feature aggregation [29, 28], and transformer-based boundary refinement [34] within a unified one-stage instance segmentation framework [21] yields state-of-the-art individual tree crown segmentation performance, while remaining computationally practical for large-scale geospatial workflows. The architectural principles established here — particularly the primacy of principled multimodal fusion over simple channel concatenation — provide a generalisable foundation for future research in multi-spectral remote sensing instance segmentation.

BIBLIOGRAPHY

[1] Food and Agriculture Organization of the United Nations (FAO), "Global Forest Resources Assessment 2020: Main Report," FAO, Rome, 2020.

[2] Y. Pan, R. A. Birdsey, J. Fang, *et al.*, "A large and persistent carbon sink in the world's forests," *Science*, vol. 333, no. 6045, pp. 988–993, 2011.

[3] D. J. Nowak and D. E. Crane, "Carbon storage and sequestration by urban trees in the USA," *Environmental Pollution*, vol. 116, no. 3, pp. 381–389, 2002.

[4] H. Kaartinen, J. Hyyppä, X. Yu, *et al.*, "An international comparison of individual tree detection and extraction using airborne laser scanning," *Remote Sensing*, vol. 4, no. 4, pp. 950–974, 2012.

[5] Z. Zhen, L. Quackenbush, and L. Zhang, "Trends in automatic individual tree crown detection and delineation — Evolution of LiDAR data," *Remote Sensing*, vol. 8, no. 4, p. 333, 2016.

[6] R. J. Pollock, "The automatic recognition of individual trees in aerial images of forests based on a synthetic tree crown model," Ph.D. dissertation, Univ. British Columbia, 1996.

[7] Q. Chen, D. Baldocchi, P. Gong, and M. Kelly, "Isolating individual trees in a savanna woodland using small footprint lidar data," *Photogrammetric Engineering & Remote Sensing*, vol. 72, no. 8, pp. 923–932, 2006.

[8] M. Dalponte and D. A. Coomes, "Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data," *Methods in Ecology and Evolution*, vol. 7, no. 10, pp. 1236–1245, 2016.

[9] S. J. Graves, G. P. Asner, R. E. Martin, *et al.*, "Tree species abundance predictions in a tropical agricultural landscape with a supervised classification model and Landsat 8 imagery," *Forests*, vol. 7, no. 9, p. 183, 2016.

[10] T. Blaschke, G. J. Hay, M. Kelly, *et al.*, "Geographic object-based image analysis — Towards a new paradigm," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 180–191, 2014.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, pp. 234–241, 2015.

[12] M. Brandt, C. Tucker, A. Kariryaa, *et al.*, "An unexpectedly large count of trees in the West African Sahara and Sahel," *Nature*, vol. 587, pp. 78–82, 2020.

[13] M. Mugabowindekwe, M. Brandt, J. Chave, *et al.*, "Nation-wide mapping of tree-level aboveground carbon stocks in Rwanda," *Nature Climate Change*, vol. 13, pp. 91–97, 2023.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, pp. 2961–2969, 2017.

[15] J. G. R. Ball, C. A. Petrova, and B. Coomes, "Using deep learning to detect individual tree crowns in airborne imagery of tropical and temperate forests," *Methods in Ecology and Evolution*, vol. 13, no. 2, pp. 332–340, 2022.

[16] Y. Yu, X. Li, P. Zhu, Q. Liu, W. Guo, and B. Wu, "Individual tree crown extraction of high-resolution satellite imagery based on a refined instance segmentation network," *Remote Sensing*, vol. 13, no. 18, p. 3532, 2021.

[17] Z. Lin, T. Liu, and X. Zhong, "Tree crown delineation in taiga-tundra ecotone using UAV imagery with Mask R-CNN," *Remote Sensing*, vol. 14, p. 5882, 2022.

[18] X. Zhang, W. Li, H. Fu, and J. Liu, "Individual tree crown segmentation for broadleaved forest using instance segmentation," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, pp. 779–788, 2016.

[20] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[21] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics (Version 8.0.0)," Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[22] X. Sun, S. Wang, M. Li, and Y. Luo, "Individual tree crown detection using improved YOLOv4 in dense forest," *Remote Sensing*, vol. 14, p. 3197, 2022.

[23] J. Dong, P. Zhu, H. Zhang, *et al.*, "A high-accuracy individual tree crown detection method based on improved YOLOv7," *Forests*, vol. 14, p. 1146, 2023.

[24] Y. Shi, Z. Chen, and L. Wang, "ITC-YOLO: Individual tree crown segmentation in tropical rainforest with CBAM and BiFPN," *Remote Sensing*, vol. 17, no. 3, p. 412, 2025.

[25] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. ICCV*, pp. 764–773, 2017.

[26] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. CVPR*, pp. 9308–9316, 2019.

[27] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016.

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, pp. 2117–2125, 2017.

[29] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. CVPR*, pp. 10781–10790, 2020.

[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proc. CVPR*, pp. 7132–7141, 2018.

[31] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, pp. 3–19, 2018.

[32] L. Sun, Y. Zhang, K. Wang, and H. Zhao, "BIC-Fusion: High-frequency feature extraction for multi-spectral object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–12, 2024.

[33] W. Zhang, X. Liu, and F. Chen, "MRD-YOLO: Multi-modal remote sensing detection with attention-guided fusion," *Remote Sensing Letters*, vol. 15, no. 6, pp. 578–590, 2024.

[34] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, *et al.*, "Segment anything," in *Proc. ICCV*, pp. 4015–4026, 2023.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in NeurIPS*, vol. 32, 2019.

Vinod kumar

ankit_singh2

ankit_singh_mtech

Document Details

Submission ID
trn:oid:::27535:140917475

Submission Date
May 29, 2026, 11:54 AM GMT+5:30

Download Date
May 29, 2026, 12:01 PM GMT+5:30

File Name
ankit_singh2.pdf

File Size
7.5 MB

36 Pages

9,709 Words

56,048 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **54** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
3% Publications
3% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Vinod kumar

ankit_singh2

ankit_singh_mtech

Document Details

Submission ID
trn:oid:::27535:140917475

Submission Date
May 29, 2026, 11:54 AM GMT+5:30

Download Date
May 29, 2026, 12:01 PM GMT+5:30

File Name
ankit_singh2.pdf

File Size
7.5 MB

36 Pages

9,709 Words

56,048 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.