Tree Classification and Detection Using Open Computer Vision

Dr.C.P. Divate[1], Ms.N.R. Bhokre[2], Mr. S. M. Patil[3], Mayuri M. Sutar[4], Riya S. Mohite[5], Renuka K. Satnaik[6], Husna M. Naikwadi[7], Gayatri A. Chavan[8], Rutja C. Jadhav[9]

[1,2] *Professor, Department of Computer Engineering, Shri Ambabai Talim Sanstha's Sanjay Bhokare Group of Institute, Miraj, Maharashtra, India*

[4,5,6,7,8,9] *Students, Department of Computer Engineering, Shri Ambabai Talim Sanstha's Sanjay Bhokare Group of Institute, Miraj, Maharashtra, India*

Abstract—**Tree detection and classification play an important role in environmental monitoring, forest management, urban planning, and biodiversity conservation. With the advancement of computer vision techniques, automated tree analysis has become more accurate and efficient. This paper presents a method for tree classification and detection using Open Computer Vision (OpenCV). The proposed approach uses image processing techniques such as preprocessing, feature extraction, and classification to identify and classify different types of trees from images. The system aims to reduce manual effort, improve accuracy, and provide a cost-effective solution for large-scale tree monitoring.**

I. INTRODUCTION

Trees are a vital part of the ecosystem and play a crucial role in maintaining environmental balance. Traditional methods of tree identification and classification are manual, time-consuming, and prone to human error.

With the rapid growth of computer vision and image processing technologies, automated tree detection has become possible.

OpenCV is an open-source computer vision Library that provides various tools for image processing, object detection, and machine Learning. By using OpenCV, trees can be detected and classified based on features such as shape, color, texture, and size. This project focuses on using OpenCV techniques to automatically detect and classify trees from digital images. Zero-shot capabilities: Researchers have used VLMs for tree species classification by feeding the model an image snippet of a detected tree crown along with a prompt like "Is this a pine or a spruce?" or "Identify the species in the image".

Domain adaptation: To improve performance on fine-grained and hierarchical classification tasks common in forestry, models can be continually pre-trained on domain-specific datasets (e.g., GlobalGeoTree, TreeSatAI) to capture complex visual and semantic relationships unique to tree species.

Structured reasoning: Some research explores using "tree-based" or hierarchical reasoning prompts with VLMs, guiding the model through a decision-tree-like structure to improve interpretability and accuracy, although this approach can have limitations compared to standard zero-shot prompting.

II. LITERATURE REVIEW

Several researchers have worked on tree detection and classification using image processing and machine Learning techniques: Earlier studies used manual feature extraction techniques Like edge detection and color histograms for tree identification.

Some researchers applied machine learning algorithms such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) for tree classification.

Recent studies focus on deep Learning approaches, especially Convolutional Neural Networks (CNNs), for better accuracy.

OpenCV has been widely used for preprocessing tasks such as noise removal, segmentation, and contour detection.

From the Literature, it is observed that combining OpenCV with machine Learning improves detection accuracy and reduces computational cost.

environmental applications. Accuracy: Deep learning models generally achieve high performance in tree species classification, with F1-scores often exceeding 0.9 (90%) for specific datasets. Point cloud methods have reported accuracies as high as 96% for specific species pairs.

Seasonal Variability: The performance of models can be affected by the season when images are captured; advanced models (ViT, EfficientNetB0) prove more effective with fall images than simpler ones.

Data Choice**:** The selection of data source (e.g., LiDAR vs. UAV imagery) depends on factors like cost, desired accuracy, and terrain complexity. Fusing data from multiple sources (e.g., WorldView-3 and Google Earth images) can improve classification accuracy.

ss

III. RESEARCH OBJECTIVES

The main objectives of this research are:

1. To develop an automated system for tree detection using OpenCV.
2. To classify different types of trees based on visual features.
3. To reduce manual effort in tree identification.
4. To improve accuracy and efficiency in tree monitoring
5. To provide a cost-effective and scalable solution for

IV. METHODOLOGY

Module1–Image Acquisition and Dataset Preparation

This module is responsible for collecting and organizing the image data required for training and testing the system. Tree images are acquired using digital cameras, mobile devices, or publicly available datasets. The dataset includes images captured under varying environmental conditions such as different lighting, backgrounds, and angles to improve model robustness.

All collected images are labeled manually based on predefined tree categories (e.g., deciduous, evergreen, fruit-bearing). The dataset is then split into training, validation, and testing sets to ensure unbiased performance evaluation. Proper dataset preparation is crucial as the accuracy of the classification model largely depends on data quality and diversity.

Module 2 – Image Preprocessing Module

The preprocessing module enhances raw images to make them suitable for analysis. Images are resized to a uniform resolution to maintain consistency across the dataset. Noise removal techniques such as Gaussian Blur and Median Filtering are applied to reduce distortions caused by camera sensors or environmental factors.

Color space conversion (RGB to HSV or Grayscale) is performed to improve segmentation accuracy. Histogram equalization is applied to enhance contrast, especially I n images captured under poor lighting conditions. These preprocessing steps significantly improve feature visibility and detection accuracy in later stages.

Module3–Tree Detection and Segmentation Module

This module focuses on identifying tree regions within the image. OpenCV-based segmentation techniques such as color thresholding, edge detection (Canny), and contour detection are used to separate trees from the background.

Morphological operations like dilation and erosion refine detected regions by removing small unwanted objects and filling gaps. Bounding boxes are drawn around detected tree contours to localize tree positions. This detection step ensures that only relevant image regions are forwarded to the classification stage, improving efficiency and reducing false predictions.

Module 4 – Feature Extraction Module

Once tree regions are detected, meaningful features are extracted to differentiate between tree types. Features include:

Color features: Mean and histogram values from HSV or RGB channels

Texture features: Gray Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP)

Shape features: Contour area, perimeter, aspect ratio, and compactness

These features provide a numerical representation of visual tree characteristics, enabling accurate classification by machine learning models.

Module5–Classification and Model Training Module
The extracted features are used to train machine learning models such as Support Vector Machines (SVM), Random Forest, or lightweight Convolutional Neural Networks (CNN). The training process involves feeding labeled data into the model and adjusting parameters to minimize classification error.
Model performance is evaluated using accuracy, precision, recall, and confusion matrices. Hyperparameter tuning and cross-validation are applied to improve generalization and avoid overfitting.

Module6– Real-Time Detection and Validation Module
In the final module, the trained model is integrated with a real-time camera feed using OpenCV. Each video frame undergoes preprocessing, detection, feature extraction, and classification sequentially. The detected tree type is displayed on the screen with bounding boxes and labels.
System performance is validated under real-world conditions to assess robustness, processing speed, and classification accuracy. This module confirms the practical applicability of the system for environmental monitoring and field deployment.

V. CONCLUSION

In this paper, a tree classification and detection system using Open Computer Vision (OpenCV) has been presented. The proposed system effectively detects trees and classifies them based on visual features extracted from images. By using image processing techniques, the system reduces manual effort and achieves reliable accuracy with low computational cost.
The results show that this approach is efficient and practical for real-world applications such as forest management, urban planning, and environmental monitoring. The simplicity and scalability of the system make it suitable for deployment in resource-constrained environments.

Future enhancements may include the integration of deep learning models to improve classification accuracy and real-time performance under complex environmental conditions.

REFERENCES

[1] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.
[2] Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th Edition). Pearson Education.
[3] Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).
[4] Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.
[5] Redmon, J., et al. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[6] A comprehensive review on tree detection methods using point cloud and aerial imagery from unmanned aerial vehicles" (2024).