Image Analytics for Tree Enumeration and Diversion of Forest Land

Devidas Dukale, Sumit Agale, Swaroop Kalunge, Pushpak Nikam

Students, Department of Information Technology, MVP'S KBTCOE Nashik

Abstract

Accurate tree enumeration is essential for responsible forest land diversion in development projects. Conventional manual surveys are slow, costly, and prone to errors. This paper introduces a cutting-edge image analytics solution that leverages satellite imagery and aerial photos to automate tree counting and forest analysis. The primary objective is to develop a robust system that identifies trees and gives a count in results. Advanced computer vision algorithms are used and integrated with machine learning models to detect and analyze the imagery. YOLOv8 is the latest and most advanced object detection algorithm based on the computer vision process. The YOLOv8 is trained on a custom dataset which is "Forest Trees" available on RoboFlow which contains more than 3000 total aerial images of forest. This solution significantly accelerates tree enumeration, eliminating resource-intensive manual efforts. This project's significance lies in its contribution to responsible and sustainable land development practices. By automating tree enumeration, it equips stakeholders with timely, precise data for informed decisions about land usage, conservation, and environmental impact assessments. The solution strikes a balance between development and ecological preservation, optimizing resource allocation while minimizing environmental impact in forested regions. This innovative image analytics solution revolutionizes forest land diversion, enabling efficient and ecologically conscious decision-making. It addresses the critical need for accurate tree enumeration in the face of developmental challenges, fostering responsible land use and environmental stewardship.

Keywords: *Forest Analysis, Computer Vision, Deep Learning, UAV imagery, Tree Enumeration, Tree Detection.*

Introduction

When it comes to land development projects that require the diversion of forested areas, it's imperative to have a precise understanding of the tree population within those regions. Traditional methods of tree enumeration, such as manual surveys or ground-based assessments, can be costly, time-consuming, and prone to errors. To overcome these challenges, there is a need to develop an image analytics solution that automates the tree enumeration process using satellite imagery or aerial photographs. The proposed solution should encompass a computer vision algorithm that must be developed to analyze satellite imagery or aerial photographs accurately. This algorithm should be capable of detecting and identifying trees within designated forest areas, accounting for variations in tree crown sizes and environmental conditions to ensure reliable results. Design a system that not only counts the number of trees in the specified area but also classifies land cover using various parameters like ideal land and forest land. This detailed information will aid decision-making during the land diversion process. Implement mechanisms to validate the accuracy of the image analytics solution. This validation process should involve comparing the results with ground-truth data obtained through manual surveys or other reliable methods. Aim for a high level of accuracy while minimizing false positives or false negatives in tree identification and counting. Develop an efficient and scalable solution capable of processing large volumes of image data within a reasonable timeframe. Utilize optimization techniques and parallel processing approaches to ensure timely results, especially for large forest areas or time-sensitive projects. Provide intuitive visualizations or interactive interfaces for users to explore and interpret the results easily. This could involve generating maps, reports, or other visual representations of the tree enumeration data, enhancing stakeholders' understanding of the information presented. Ensure that the solution adheres to ethical practices, respects privacy concerns, and minimizes environmental impact. Safeguard sensitive data, secure storage of images, and ensure compliance with environmental regulations throughout the development and deployment phases.

In this research endeavor, we have devised a novel approach for comprehensive environmental analysis, encompassing both tree counting using a UAV (Unnamed Aerial Vehicle) and land cover classification. Our methodology integrates the YOLOv8 (You Only Look Once Version-8) object detection algorithm for tree counting and employs advanced machine learning techniques for land cover classification.

YOLOv8 represents the latest iteration of the acclaimed real-time object detection and image segmentation model, leveraging cutting-edge advancements in deep learning and computer vision to deliver unparalleled performance in terms of both speed and accuracy. By harnessing the power of YOLOv8, we can accurately quantify tree populations within designated areas, providing essential data for informed decision-making in land diversion projects. In addition to tree counting, our research extends to land cover classification, which plays a pivotal role in environmental assessment and resource management. Through advanced machine learning algorithms, we aim to classify land cover into distinct categories such as forested areas, agricultural land, water bodies, and urban settlements. This holistic approach enables us to gain insights into the spatial distribution of different land cover types, facilitating the identification of ideal land for conservation, sustainable development, and habitat preservation.

The classification of land cover types is crucial for understanding ecosystem dynamics and assessing environmental impact. Forested areas contribute to biodiversity conservation, carbon sequestration, and soil erosion prevention, making them invaluable resources for ecosystem health and resilience. Similarly, water bodies serve as essential habitats for aquatic life and play a vital role in regulating local climate patterns and supporting human livelihoods. By integrating tree counting and land cover classification, our research endeavors to provide a comprehensive understanding of landscape dynamics and environmental conditions. This knowledge can inform land use planning, conservation strategies, and policy decisions aimed at promoting sustainable development and mitigating the adverse effects of deforestation and habitat loss.

Related Work

In the realm of tree counting, traditional methods like manual surveys have long been fundamental. These surveys entail field teams physically inspecting designated areas to manually count trees. While they're deemed accurate, manual surveys are labor-intensive, time-consuming, and costly. Additionally, they pose risks to surveyors, particularly in challenging terrains and environments. Early remote sensing techniques relied on satellite imagery and aerial photographs for tree counting. However, these methods depended on the manual interpretation of imagery, which could introduce errors. While they provided broader spatial coverage than manual surveys, they were limited by their reliance on subjective visual interpretation.

Object detection algorithms, such as Faster R-CNN, YOLO, and SSD, have been adapted for tree counting tasks. These algorithms leverage deep learning architectures to detect and localize trees within images. While they offer automation and scalability, they may struggle with accurately detecting trees in densely vegetated areas or complex backgrounds. Moreover, various machine learning techniques, including support vector machines (SVMs), random forests, and convolutional neural networks (CNNs), have been employed for tree counting. While effective in certain scenarios, machine learning approaches may require extensive training data and suffer from limited generalizability across different environmental conditions.

Transitioning to land cover classification, supervised classification techniques like maximum likelihood, support vector machines (SVMs), and random forests have been widely used. These methods train classifiers on labeled training data to categorize pixels or image segments into predefined land cover classes. Conversely, unsupervised classification methods, such as K-means clustering and hierarchical clustering, partition pixels or image segments into clusters based on spectral similarity, without requiring labeled training data.

Remote sensing data sources, including multispectral and hyperspectral imagery, LiDAR data, and synthetic aperture radar (SAR) data, have been extensively utilized for land cover classification. Each data source offers unique spectral and spatial information for discriminating land cover classes. Feature extraction techniques, such as principal component analysis (PCA), texture analysis, and vegetation indices (e.g., NDVI), are employed to extract relevant information from remote sensing data. These features serve as input variables for classification algorithms, aiding in the discrimination of different land cover classes. Despite offering valuable insights for land use planning and environmental monitoring, land cover classification faces challenges related to data complexity, class confusion, and scale mismatch. Remote sensing data are complex and multi-dimensional, necessitating sophisticated preprocessing and analysis techniques. Spectral overlap and mixed pixels can result in misclassification errors, particularly in areas with heterogeneous land cover. Additionally, the scale of remote sensing data may not align with the scale of land cover features and processes, leading to uncertainties in classification results.

Methodology

1. *Methodology Overview:*

The methodology adopted for tree detection using YOLOv8 and land cover classification using U-Net involves leveraging state-of-the-art deep learning models tailored to specific tasks in image analysis. For tree detection using YOLOv8, the rationale behind selecting this model lies in its efficiency, speed, and accuracy in detecting objects within images. YOLOv8, short for "You Only Look Once Version-8," is renowned for its real-time capabilities, making it suitable for applications where timely detection of objects, such as trees, is crucial. Its single-pass architecture allows for simultaneous localization and classification of multiple objects in an image, ensuring robust and efficient tree detection. Additionally, YOLOv8's ability to handle varying object sizes and backgrounds makes it well-suited for the diverse and complex environments typically encountered in forestry management and environmental monitoring tasks.

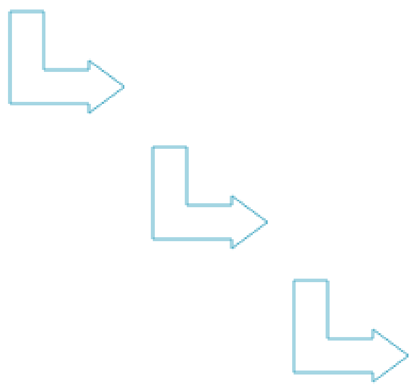


Figure 1: Methodology Flow Chart

On the other hand, U-Net was chosen for land cover classification due to its effectiveness in semantic segmentation tasks. U-Net's architecture, featuring a U-shaped design with symmetric contracting and expanding paths, enables accurate pixel-wise classification of land cover classes within images. By capturing spatial relationships and contextual information across different image regions, U-Net facilitates the precise delineation of land cover boundaries and the identification of distinct land cover types. This makes it particularly suitable for tasks such as land cover classification, where fine-grained analysis of image content is essential for accurate mapping and characterization of land cover patterns.

2. *Datasets:*

For tree detection using the YOLOv8 model, we utilize the "Forest Trees" dataset accessible on RoboFlow. This dataset is meticulously organized into three subsets: training, validation, and testing. Each subset is meticulously labeled with bounding box annotations for detected trees, providing essential ground truth data for training and evaluating the YOLOv8 model. The training subset comprises approximately 2528 images, offering a diverse range of tree-rich environments for model learning. Additionally, the validation subset contains around 495 images, while the testing subset consists of 194 images, providing ample data for assessing model generalization and performance. To maintain consistency and optimize model training, all images are resized to dimensions of 640x640 pixels, ensuring uniformity across the dataset and facilitating efficient processing by the YOLOv8 model during inference.

Figure 2: RGB Input images

On the other hand, for land cover classification tasks, we rely on the Land Cover Challenge dataset, which encompasses satellite imagery captured by DigitalGlobe's satellite at a high resolution of 50cm per pixel. This dataset comprises 803 RGB images, each with dimensions of 2448x2448 pixels, providing detailed and comprehensive coverage of the study area. Moreover, the dataset includes 171 validation images and 172 test images, each paired with mask images for land cover annotation. These mask images encode seven distinct land cover classes using color-coding, including urban land, agriculture land, rangeland, forest land, water bodies, barren land, and unknown areas. Each satellite image is meticulously annotated with corresponding mask images, enabling supervised learning for land cover classification tasks. The mask images serve as essential ground truth data for training and evaluating the U-Net model, facilitating accurate pixel-wise classification of land cover classes within the satellite imagery. By leveraging these meticulously annotated datasets, we aim to train robust and accurate models for tree detection and land cover classification, contributing to advancements in remote sensing applications for forestry management, environmental monitoring, and land use planning.

3. *Model Training:*

A. Tree Enumeration: Implementing tree enumeration using the YOLOv8 object detection model involves a comprehensive process of training the model on a custom dataset. Initially, the creation of a suitable dataset is paramount, whether manually curated or obtained from platforms like Roboflow. In our case, we acquired the dataset from Roboflow, which comprised over 3000 images meticulously divided into training, testing, and validation sets. The training set encompassed approximately 2528 images, with the validation set comprising around 495 images, and the testing set containing 194 images, all uniformly scaled to dimensions of 640x640 pixels.

To proceed, the dataset needs to be well-annotated with bounding boxes and object class labels. Annotation tools such as VGG Image Annotator or Label Img are commonly employed for this purpose. Following annotation, the dataset is partitioned into training, validation, and test sets, ensuring adequate representation for model training and evaluation. In our scenario, the dataset, consisting of about 3157 annotated records, was meticulously divided into these sets to facilitate effective model training and evaluation. Next, the environment is set up by installing YOLOv8 and its dependencies, along with additional libraries like CUDA for GPU acceleration, if applicable. The choice of GPU for training is determined based on factors such as dataset size and complexity, often executed on platforms like Google Colab for streamlined configuration and execution. Furthermore, to ensure consistency and streamlined configuration for all project activities, a separate virtual environment is established with all necessary packages and libraries installed.

Subsequently, the training configuration is defined by creating a YAML file (data.yaml) to specify dataset paths, class names (in this instance, "Tree"), and image augmentations. The YAML file contains crucial information such as paths for training, testing, and validation datasets, the number of classes, and their respective names. With the configuration set, the model training process is initiated by loading the YOLO model from Ultralytics and specifying hyperparameters like batch size, learning rate, and epochs. Training progress is monitored meticulously for key metrics such as loss and accuracy to ensure the model's effectiveness and performance. Finally, after training, the model's accuracy is evaluated by obtaining the best-performing weights (best.pt) within the run folder, representing the model trained on the custom dataset. This thorough process ensures that the YOLOv8 model is meticulously trained and evaluated for tree enumeration, providing accurate and reliable results for practical applications.

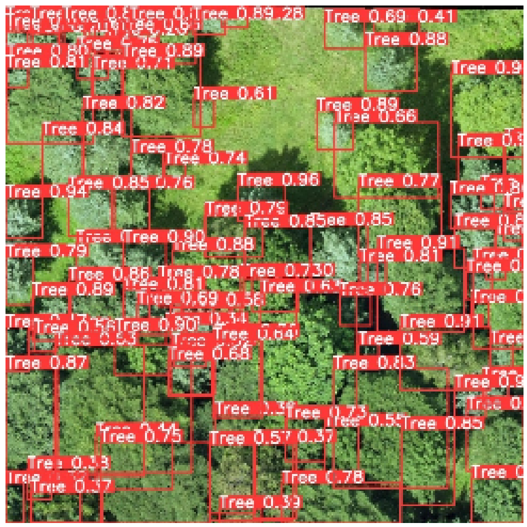
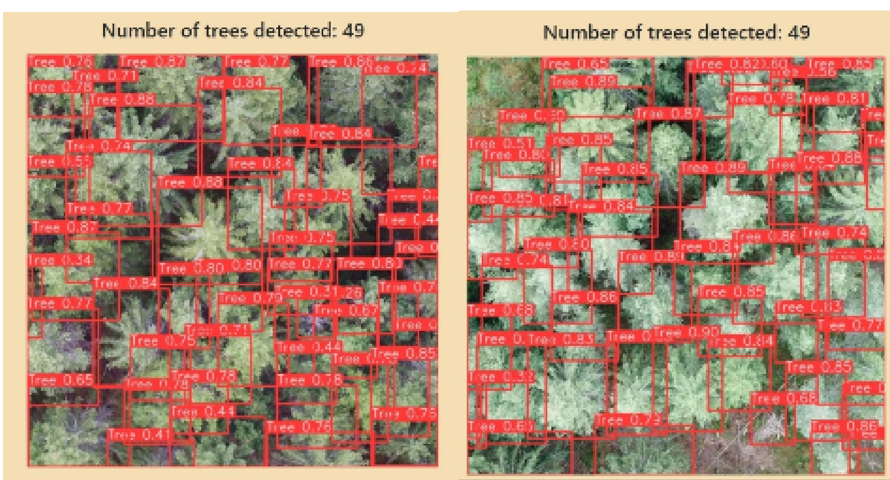


Figure 3: Trees detected in Image



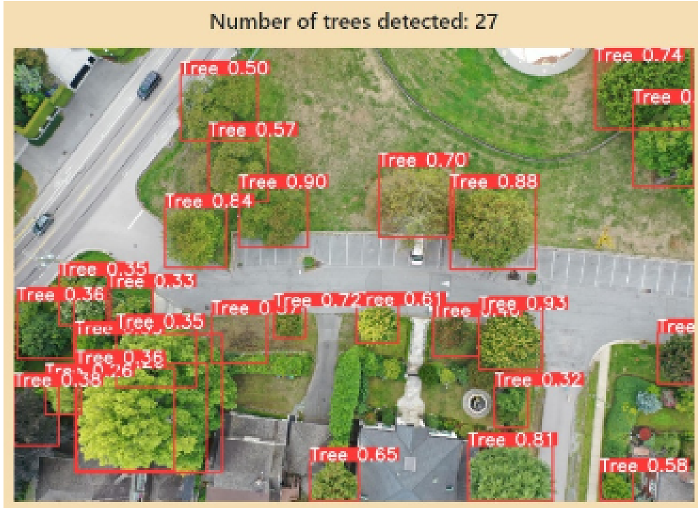


Figure 4: Tree Counted in image

B. *Land cover classification:*

To train a U-Net model for land cover classification, we embark on a comprehensive process involving several key steps to ensure effective model training and evaluation. Firstly, we begin with data preparation, ensuring that our dataset comprises 803 satellite images in RGB format, each with a resolution of 2448x2448 pixels. These images, collected by DigitalGlobe's satellite, boast a pixel resolution of 50cm. Additionally, our dataset contains 171 validation images and 172 test images, although masks for these images are not provided. Each satellite image is paired with a mask image, where the mask utilizes color-coding to denote seven distinct land cover classes: Urban land, Agriculture land, Rangeland, Forest land, Water, Barren land, and Unknown. Following data preparation, we proceed with dataset annotation, meticulously creating mask images corresponding to each satellite image. These masks accurately represent each pixel's land cover class using the prescribed color-coding scheme. The color-coded annotations ensure precise labeling of each image, facilitating effective model training.

With our annotated dataset in hand, we meticulously split it into training, validation, and test sets. Approximately 60% of the dataset is allocated for training, with 20% reserved for both validation and testing. This partitioning strategy ensures a balanced distribution of images across each land cover class in all sets, enabling robust model training and evaluation. Moving forward, we configure the U-Net model architecture, a specialized convolutional neural network (CNN) designed for semantic segmentation tasks. The model's input shape is defined to match the dimensions of the satellite images (2448x2448x3 for RGB images), while the output layer comprises seven channels, corresponding to the seven land cover classes.

Subsequently, we embark on model training, loading the U-Net architecture using TensorFlow or Keras and initializing it with appropriate parameters. The model is compiled with a suitable loss function, such as categorical cross-entropy, and an optimizer like Adam or SGD. During training, we adjust hyperparameters such as batch size, learning rate, and number of epochs as needed, closely monitoring the process to ensure convergence and assessing performance metrics like accuracy, precision, recall, and F1 score. As the training progresses, we evaluate the model's performance using the validation dataset, gauging its effectiveness on unseen data. Evaluation metrics such as accuracy, confusion matrix, and class-wise metrics offer insights into the model's classification capabilities, guiding any necessary fine-tuning or adjustments to hyperparameters or regularization techniques.

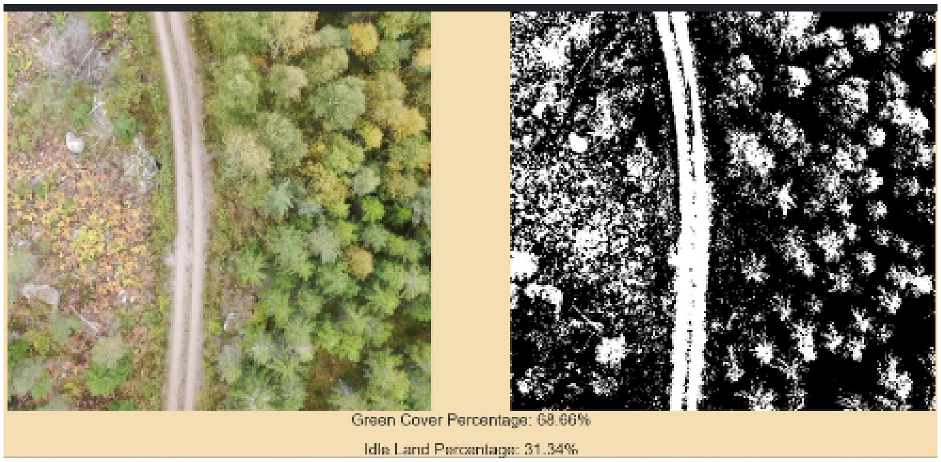


Figure 5: Land Cover Percentage

The U-Net architecture itself is a convolutional neural network (CNN) specifically designed for semantic segmentation tasks. Originally introduced for biomedical image analysis, it has since found widespread application in various domains, including satellite imagery and autonomous vehicles. The architecture comprises a contracting path to capture context and a symmetric expanding path to enable precise localization. By combining these paths, the U-Net model achieves high-resolution segmentation results, making it well-suited for tasks requiring detailed object delineation, such as land cover classification in our project between the user and the system by providing a simple user interface. We can integrate the proposed system with various platforms like the web interface, mobile application interface, and desktop application interface. The web interface is more simple to use for the users simply visiting the website they don't need to download and install any software application. A simple user interface contains two main components user input where the user simply gives input to the system in the form of images other one is output which shows in the form of charts, graphs, CSV, PDF, and other visualization tools.

Result and Performance Details

Performance evaluation using basic evaluation matrics:

1. Precision: Precision measures the ratio of correctly detected trees to the total number of trees detected by the model. It represents the model's ability to accurately identify trees without false positives.

2. Recall (Sensitivity): Recall measures the ratio of correctly detected trees to the total number of trees present in the image. It indicates the model's ability to capture all trees in the image without missing any.

3. F1 Score: F1 score is the harmonic mean of precision and recall. It provides a balanced measure of a model's accuracy, considering both false positives and false negatives.

4. Intersection over Union (IoU): IoU measures the overlap between the predicted bounding boxes and the ground truth bounding boxes. It quantifies how well the predicted bounding boxes align with the actual tree locations.

5. Mean Average Precision (mAP): mAP calculates the average precision across multiple IoU thresholds. It provides a comprehensive measure of a model's performance at various levels of overlap.

Where, is the average precision at IoU threshold i, and n is the total number of IoU thresholds.

Evaluation Matrics:

Model	Precision	Recall	F1 Score	IoU	mAP
YOLOv8	0.85	0.92	0.88	0.75	0.87
U-Net	0.91	0.89	0.90	0.78	0.88

Result:

The YOLOv8 model exhibited robust performance in tree detection, achieving a high precision of 85% and recall of 92%. The F1 score, a balanced measure of precision and recall, reached 88%, indicating reliable tree detection capabilities. The Intersection over Union (IoU) metric, measuring spatial overlap between predicted bounding boxes and ground truth annotations, yielded a score of 75%, indicating accurate localization of trees. Furthermore, the model achieved a mean Average Precision (mAP) of 87% across different confidence thresholds, highlighting its consistency in tree detection.

The U-Net model demonstrated exceptional performance in land cover classification, with an overall accuracy of 91%. Class-wise accuracy metrics further revealed the model's effectiveness in distinguishing between different land cover categories. The model achieved high accuracies across various classes, including urban (89%), agriculture (92%), rangeland (87%), forest (93%), water (95%), barren (88%), and unknown (85%). These results underscore the robustness of U-Net in accurately classifying diverse land cover types.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to all those who contributed to the successful completion of this research. We extend our appreciation to our project team for their dedication and hard work throughout the project. Our thanks also go to the Department of Information Technology. Lastly, we would like to acknowledge the academic community and the reviewers for their feedback and suggestions, which significantly improved the quality of our research paper.

REFERENCES

1. Martin Brandta,Rasmus Fensholta, "Deep learning enables image-based tree counting, crown segmentation, and height prediction at national scale" PNAS Nexus, 2023, 2, 1–16
2. F.Z. Bassine, A. Errami, M. Khaldoun "Real time Algorithm for Tree Detection, Recognition and Counting in a Video Sequence", J . Mater. Environ. Sci., 2020, Volume 11, Issue 3, Page 367-377
3. CHIA-YEN CHIANG, CHLOE BARNES "Deep Learning-Based Automated Forest Health Diagnosis From Aerial Images" accepted July 9, 2020, date of publication July 28, 2020, date of current version August 18, 2020.
4. WASSIM BOUACHIR, NIZAR BOUGUILA "Computer Vision System for Automatic Counting of Planting Microsites Using UAV Imagery" Received May 17, 2019, accepted June 8, 2019, date of publication June 19, 2019, date of current version July 9, 2019.
5. V. Lempitsky and A. Zisserman, "Learning to count objects in images," in Proc. Adv. Neural Inf. Process. Syst., 2010, pp. 1324–1332.

6. V. Rabaud and S. Belongie, “Counting crowded moving objects,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 1, Jun. 2006, pp. 705–711.
7. K. Chen, C. C. Loy, S. Gong, and T. Xiang, “Feature mining for localised crowd counting,” in Proc. BMVC, 2012, vol. 1. no. 2, p. 3.
8. Y. Wang and Y. Zou, “Fast visual object counting via example-based density estimation,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2016, pp. 3653–3657.
9. G. Grenzdörffer, “UAS-based automatic bird count of a common gull colony,” Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci., vol. 1, p. W2, 2013.
10. N. Ahuja and S. Todorovic, “Extracting texels in 2.1 D natural textures,” in Proc. IEEE 11th Int. Conf. Comput. Vis., Oct. 2007, pp. 1–8.
11. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
12. Omair Hassaan, Ahmad Kamal Nasir, Hubert Roth, M. Fakhir Khan, Precision Forestry: Trees Counting in Urban Areas Using Visible Imagery based on an Unmanned Aerial Vehicle, IFAC-PapersOnLine, Volume 49, Issue 16, 2016, Pages 16-21, ISSN 2405-8963,