IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 December 2024, accepted 5 January 2025, date of publication 10 January 2025, date of current version 17 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3528215

RESEARCH ARTICLE

Satellite-Based Forest Stand Detection Using Artificial Intelligence

PATRIK KOVAČOVIČ, RASTISLAV PIRNÍK, JÚLIA KAFKOVÁ, MÁRIO MICHÁLIK, ALŽBETA KANÁLIKOVÁ, AND PAVOL KUCHÁR

Department of Control and Information Systems, Faculty of Electrical Engineering and Information Technology, University of Žilina, 010 26 Žilina, Slovakia

Corresponding author: Patrik Kovačovič (patrik.kovacovic@feit.uniza.sk)

ABSTRACT The forest constitutes an essential and irreplaceable component of life for all organisms, with its primary significance lying in its role in creating a breathable atmosphere on Earth. Forests are vital for human health and well-being and hold significant ecological and economic value for humanity. This study aims to propose a method for identifying forest stands using artificial intelligence techniques. A custom dataset was developed, comprising high-quality satellite images that capture various structures such as forests, fields, roads, buildings, and lakes. This dataset was employed to train models from the category of convolutional neural networks that operate on the principle of instance segmentation. Several models, including YOLOv8, YOLOv5 and Mask R-CNN, were tested and compared. An optimal model was selected based on parameters such as detection accuracy, total training time, and the precision of labeling detected image elements. The selected model was then evaluated using images not included in the original training dataset to simulate real-world deployment scenarios. The final accuracy of best model achieved 91.67%. This model can detect the presence of forest stands in satellite images, as well as other features such as roads, buildings etc. The proposed method offers potential benefits for forest technicians, who can integrate it with other methods to monitor forest cover effectively.

INDEX TERMS Artificial intelligence, neural networks, forest, dataset, model.

I. INTRODUCTION

Forests are vital to modern society, necessitating their maintenance and regular inspection [1], [2]. As diverse ecosystems, they offer essential services [2], [3], [4], [5]. The health of forests is closely tied to human well-being, emphasizing the need to monitor their condition, expand forested areas, and regenerate trees. Forest inventories are crucial for sustainable resource management and future projections [6]. Disturbances such as logging, deforestation, wildfires, diseases, or pests significantly affect forests' ecological functions [7], with biotic disturbances having particularly wide-reaching impacts on forest ecosystems and the services they provide [7], [8].

Researchers focus on monitoring forests, particularly their health, development, wildfire impacts, and role in climate regulation. Traditionally, forest inventories rely on field measurements of attributes like tree diameter, height, and count, but this method is labor-intensive, time-consuming, and costly [9]. Drones, remote sensing, and image processing offer efficient alternatives for assessing forest stands. Drones have revolutionized forest operations by enabling real-time monitoring of landscape changes, improving threat detection for fires and pests [10]. Remote sensing is now a key technology in forestry, providing large-scale insights into ecosystems and detecting disturbances such as wildfires, logging, and pest outbreaks, which are difficult to capture through field surveys [11], [12], [13]. Image processing techniques enhance image quality and extract useful data [14], [15]. Recent advancements in image processing, particularly in the area of stereo image super-resolution, have focused on utilizing additional features from cross-view image pairs to enhance high-resolution image reconstruction. For example, methods such as CVCnet leverage both global contextual and local features for improved performance [16]. Additionally, lightweight networks like EMASRN have been developed

The associate editor coordinating the review of this manuscript and approving it for publication was Joewono Widjaja.

to address the computational challenges of super-resolution on resource-constrained devices, offering a balance between performance and efficiency [17]. Another approach utilizes interactive memory learning to implicitly capture semantic information between stereo image pairs, leading to enhanced image detail and quality [18].

In Eastern Europe, forest engineers primarily use field measurement methods alongside Geographic Information Systems (GIS) and electronic tools for mapping and dendrometric assessments. GIS are computer-based tools for visualizing, analyzing, and interpreting geographic data [19]. Although reliable, this approach is time-consuming, labor-intensive, and lacks automation. Engineers must visit forests to examine trees and the environment, then manually input data into GIS software, such as identifying high-density areas of trees or delineating boundaries. A significant advancement would be the automatic detection of forest stands using remote sensing technologies, such as satellite, aerial, or drone imagery, combined with artificial intelligence. These technologies, when integrated with GIS, enhance large-scale ecosystem health assessments and bridge gaps in environmental analysis [20].

Various image detection approaches exist, but convolutional neural networks (CNNs) are the most effective for feature extraction and task-specific applications. In this study, satellite images are used as the primary data source, and CNNs are employed to train the network to detect forest stands, as well as other elements like buildings, fields, and roads. CNNs are a key deep learning architecture, designed to learn spatial hierarchies of features via backpropagation, using convolutional, pooling, and fully connected layers for classification and image recognition [21], [22], [23], [24], [25]. Similar techniques have been used by other researchers to learn from satellite imagery and accurately identify image components.

The application of artificial intelligence, particularly neural networks and remote sensing, greatly enhances forest stand monitoring. This research aims to use a neural network to identify components of satellite images, focusing on forest stand recognition. Key objectives include accurately detecting non-forest areas, such as buildings and fields, and minimizing the time required to train the neural network. The study highlights the AI techniques used for satellite image analysis, with an emphasis on both forest stand recognition and the efficient identification of non-forest elements. A custom dataset of high-quality, dimensionally consistent, labeled satellite images has been developed, featuring forest cover as well as buildings, roads, and fields.

For effective training, the images in the dataset must be accurately labeled, uniform in size, and of high quality, allowing models to easily identify relevant components. The dataset will focus on images of forest cover, buildings, roads, and fields. Selected machine learning models will learn to recognize each labeled class from this dataset.

The models must be designed to effectively perform the required tasks. After implementing and evaluating all models, performance will be assessed based on detection accuracy, training time, and the accuracy of labeling detected components. The optimal model will then be selected to process unlabeled data, with its effectiveness evaluated by the correctness of its detection results.

This paper is structured as follows: Section II provides a literature overview. Section III details the methodology of data collection, including the creation of our custom dataset, which features examples of images obtained from satellite maps and describes the labeling process. Section IV presents the results of training the individual models on the smaller Forest_Full_V1 dataset, along with a summary of these results. Section V includes the training outcomes for the models on the optimized Forest_Full_V2 dataset. Section VI discusses the model selected for its superior parameters, which was tested on unlabeled images from a separate dataset that was not used for training, validation, or generalization. Finally, Section VII concludes the paper.

The key contributions of this research are outlined as follows:

- A custom dataset comprising 5,100 satellite images, each sized 438 × 369 pixels, has been created. The dataset is designed for use in forest stand detection research and is intended to support future studies. All images are precisely labeled, ensuring the dataset is ready for immediate application.
- An instance segmentation approach was applied to forest stand detection, representing a departure from previous methodologies.
- Several models, including YOLO and Mask-RCNN, were evaluated for their performance. The YOLOv8 model demonstrated the most favorable results among the tested approaches.

II. LITERATURE OVERVIEW

Artificial intelligence (AI) techniques, especially deep learning models, have made significant strides in forest detection by improving the analysis of large-scale remote sensing data. Traditional forest monitoring methods, which relied heavily on manual surveys and basic image analysis, were often limited in scope and efficiency. In contrast, AI models can automatically process and classify satellite images, offering enhanced accuracy and efficiency in detecting forest cover, types, and health.

Schürholz et al. [26] applied a convolutional neural network (CNN) for semantic segmentation of aerial images of mangrove forests, aiming to enumerate trees and delineate the coverage of the mangrove species Rhizophora mangle. Using the Detectron-2 library, they also identified surrounding land covers, such as mud and water. A canopy height model was created from the data, and allometric equations were applied to derive essential forest metrics. Their method achieved a precision of 97% and a recall of 87%.

Gonçalves et al. [9] took a different approach, utilizing an artificial neural network (ANN) to assess the correlation of various field-measured forest variables with

vegetation indices. Satellite imagery from the Sentinel-2 dataset was used to predict commercial tree volume. They trained 150 ANN networks, with the multilayer perceptron architecture achieving the most favorable statistical results: a root mean square error (RMSE) of less than 10% and a correlation coefficient (rr) greater than 0.98.

Li and Liu [27] focused on 3D point cloud data captured via laser measurement techniques. They encountered challenges in managing large volumes of unstructured data, which they addressed by developing the DMSdataset and the PointDMS framework. The framework uses 3D convolutional networks for semantic segmentation, achieving an overall identification accuracy of 93.5%, with an average identification accuracy of 88.7%.

Lim et al. [28] implemented semantic segmentation using the HRNet architecture, classifying pixels as either forest or non-forest. They validated their approach on 6,120 sliced images obtained from Google Earth, achieving a mean intersection over union (IoU) score of 85.58% and an accuracy of 92.24%.

Yao et al. [29] employed the YOLOv8 model, incorporating Dynamic Snake Convolution, the Multidimensional Collaborative Attention Mechanism, and Wise-IoU v3. They generated a dataset using UAV images and applied a sliding window method. Their model achieved a precision of 91.31% and a recall of 85.72%.

Bai et al. [30] combined the YOLO and SSD models with the Transformer algorithm to analyze satellite imagery in remote sensing. They outlined their algorithms in terms of supervision, attention mechanisms, and multi-scale approaches, while also highlighting future research directions in the field.

Haq et al. [31] provided a comprehensive review of the role of AI, the Internet of Things (IoT), and remote sensing technologies in monitoring forest diseases, fires, and other phenomena, laying the groundwork for future conservation efforts.

Mitra et al. [32] used CNN-based approaches to map the extent of planted forests on the Korean Peninsula, employing Sentinel-2 images with the UNet deep learning model and ResNet-34 as the backbone architecture. This model achieved a recall rate of 64% and precision of 76.8%, outperforming the ensemble-based Random Forest (RF) model, which had a recall rate of 55.2% and precision of 92%.

Bolfe et al. [33] developed a methodological framework combining the Normalized Difference Vegetation Index (NDVI), Normalized Difference Water Index (NDWI), and Soil-Adjusted Vegetation Index (SAVI), derived from Harmonized Landsat Sentinel-2 (HLS) data. The framework was integrated with machine learning algorithms like Random Forest (RF), Artificial Neural Networks (ANNs), and Extreme Gradient Boosting (XGBoost) to map agricultural intensification. The models achieved an overall accuracy between 85% and 99%.

Xiang et al. [34] utilized a Position and Orientation System (POS) for geometric correction on UAV imagery. They employed a CNN to extract forest boundaries and compared these with previous vector data, detecting forest reduction areas. The TernausNet model achieved an accuracy of 0.98 in identifying forest areas, while the Average Boundary Distance Algorithm (ABDA) filtered out misclassified patches, reaching an identification accuracy of 0.95 for reduced forest patches and a precision of 0.91.

Pei et al. [35] created a multiscale global graph convolutional neural network (MSG-GCN) and compared it with Random Forest, U-Net, and UNet++ models for classifying various forest types from aerial images. The MSG-GCN model outperformed the others, with misclassification rates between 10% and 15%.

Kalinaki et al. [36] tackled forest change detection using an attention-residual-based deep learning model with a U-Net architecture. They utilized Sentinel-2 satellite images, achieving a precision of 0.9330, which represented an improvement over other U-Net variants.

Molnár and Király [37] developed a combined approach using ESA Sentinel-2 imagery and Google Earth Engine cloud computing to detect forest disturbances. They applied vegetation indices such as NDVI and achieved an accuracy of 71% for forest damage detection and 81.9.

Ouchra et al. [38] performed land cover classification using remote sensing from Landsat 8 data. They applied six machine learning algorithms, with the Minimum Distance (MD) algorithm achieving the highest accuracy of 0.93.

Pérez-Carrasco et al. [39] developed a processing framework for generating predictions using the YOLO and Mask R-CNN models, based on UAV aerial imagery. The YOLO model achieved a precision of 0.72 and a recall of 0.68, while Mask R-CNN attained a precision of 0.82 and recall of 0.80.

Sun et al. [40] proposed a novel approach for detecting individual tree crowns (ITC) using a synergistic combination of YOLOv4 with generative adversarial networks (WGAN, CycleGAN, SinGAN). Their method achieved an overall recall of 83.6% and precision of 81.4% using airborne LiDAR data.

Xu et al. [41] applied YOLOv7 to classify tree species in transmission line corridors using aerial multispectral images. Their peak accuracy in identifying single tree species was 85.42%.

Miao et al. [42] introduced the Pseudo Tree Crown (PTC) data reformation method to enhance feature differentiation in the input layer. Using UAV-derived data and a ResNet50-based CNN in PyTorch, their model achieved a classification accuracy of 93%.

Syifa et al. [43] focused on pine wilt disease detection using drone imagery. They applied both ANN and Support Vector Machine (SVM) models, achieving accuracies of 94.13% (SVM) and 87.43% (ANN) in Anbi, and 86.59% (SVM) and 79.33% (ANN) in Wonchang.

The application of AI, particularly deep learning models, has proven to be highly effective in advancing forest detection through enhanced remote sensing data analysis. Traditional methods, limited by manual efforts and basic

image processing, have been supplanted by AI-driven techniques, which offer substantial improvements in accuracy and efficiency for detecting forest cover, types, and health. Overall, AI technologies are becoming indispensable tools for effective forest management and conservation.

A. PERFORMANCE METRICS

The effectiveness of AI models in classifying various land cover types (e.g., forest, fields, roads, lakes, and buildings) is typically evaluated using several standard metrics, including Accuracy, Precision, Recall, and the F1-score. In this study, Accuracy and Precision were primarily utilized to assess model performance. These metrics are commonly used to measure the classification performance of models.

1) ACCURACY

The accuracy of the model is defined as the proportion of correctly classified pixels (true positives and true negatives) to the total number of pixels. It is calculated as:

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP = True Positive (correctly predicted area for a given class),
- TN = True Negative (correctly predicted non-area for a given class),
- FP = False Positive (incorrectly predicted area for a given class),
- FN = False Negative (incorrectly predicted non-area for a given class).

2) PRECISION

Precision measures the accuracy of the positive predictions for each specific class. It is calculated as:

$$P_{\text{class}} = \frac{TP_{\text{class}}}{TP_{\text{class}} + FP_{\text{class}}}$$

Where:

- TP_{class} = True Positive for the specific class (e.g., forest, field, etc.),
- FP_{class} = False Positive for the specific class.

3) RECALL

Recall measures the ability of the model to detect all relevant instances of a specific class. It is calculated as:

$$R_{\text{class}} = \frac{TP_{\text{class}}}{TP_{\text{class}} + FN_{\text{class}}}$$

Where:

- TP_{class} = True Positive for the specific class,
- FN_{class} = False Negative for the specific class.

4) F1-SCORE

The F1-score is the harmonic mean of Precision and Recall. It combines both metrics into a single value that balances the trade-off between Precision and Recall. It is calculated as:

$$F1_{\text{class}} = 2 \times \frac{P_{\text{class}} \times R_{\text{class}}}{P_{\text{class}} + R_{\text{class}}}$$

Where:

- P_{class} = Precision for the specific class,
- R_{class} = Recall for the specific class.

In these equations:

- True Positive (TP) refers to correctly classified pixels for a particular class (e.g., forest, fields, roads, lakes, or buildings),
- False Positive (FP) refers to incorrectly classified pixels as a particular class,
- False Negative (FN) refers to incorrectly unclassified pixels for a particular class.

These metrics provide a comprehensive way to assess the performance of models in classifying different types of land cover. While Accuracy gives an overall measure of performance, Precision and Recall give more specific insights into the model's ability to identify a particular class (e.g., forest or buildings). The F1-score combines both Precision and Recall and is particularly useful when there is an imbalance between the classes.

III. METHODOLOGY OF THE DATA COLLECTION PROCESS AND CREATING DATASET

Numerous datasets containing satellite images are available; however, the quality of these images is often inadequate. In this study, a Python script was used to create the dataset by capturing screenshots of a specified region of interest and automatically saving them to a designated folder. The satelites.pro map [44] was utilized at a zoom level of 20 meters. Subsequently, an additional Python script was employed to generate uniformly sized images from these screenshots by utilizing the coordinates to extract two equally sized images from each screenshot, which were then incorporated into the dataset. Each image measures 876 × 739 pixels. An example image is presented in (Fig. 1).

Given that the images were of sufficient resolution, Again a similar Python script was utilized to subdivide each image into four smaller images, each measuring 438 × 369 pixels. This approach not only increased the number of images but also preserved their quality. The images were captured from various regions in Slovakia, including the Tatras and the Podunajská nížina, which are characterized by extensive tree cover, significant rivers such as the Danube and the Váh, and numerous fertile areas containing fields. Additionally, satellite images were obtained from other regions that predominantly featured forests, as well as other elements necessary for training the models. The resulting images are illustrated in (Fig. 2).

TABLE 1. This table represents studies found in the literature for parameter accurary.

Author	Data	Methodology	Accuracy
Our stury	**Forest_Full_V2**	**YOLOv8**	**91.67%**
Li et al.	DMSDataset	Framework PointDMS (3D Convolutional deep networks)	93.5%
Lim et al.	Google Earth images	HRNet semantic segmentation	92.24%
Bolfe et al.	Harmonized Landsat Sentinel-2 (HLS)	Random Forest (RF), Artificial Neural Networks (ANNs), and Extreme Gradient Boosting (XGBoost)	85% - 99%
Xiang et al.	UAV imagery data to perform geometric correction with position and orientation system (POS)	TernausNet model	98%
Xiang et al.	UAV imagery data to perform geometric correction with position and orientation system (POS)	Average boundary distance algorithm (ABDA)	95%
Pei et al.	Aerial Images	Multiscale global graph convolutional neural network (MSG-GCN)	85% - 90%
Molnar et al.	ESA Sentinel-2 satellite imagery	Google Earth Engine cloud computing with field-based forest inventory data	81.9%
Ouchra et al.	Landsat 8 satellite data	Minimum distance (MD) algorithm	93%
Xu et al.	Aerial multispectral images obtained from drones	YOLOv7	85.42%
Miao et al.	UAV images	PyTorch Framework with ResNet-50	93%
Syifa et al.	Drone data from area Anbi	Artificial neural network (ANN)	87.43%
Syifa et al.	Drone data from area Anbi	Support vector machine (SVM)	94.13%
Syifa et al.	Drone data from area Wonchang	Artificial neural network (ANN)	86.59%
Syifa et al.	Drone data from area Wonchang	Support vector machine (SVM)	79.33%

TABLE 2. This table represents studies found in the literature for parameter precision.

Author	Data	Methodology	Precision
Our stury	**Forest_Full_V2**	**YOLOv8**	**93%**
Schürholz et al.	Aerial Images	CNN + Detectron 2 library	97%
Yao et al.	UAV Images	YOLOv8 with Dynamic Snake Convolution, the Multidimensional Collaborative Attention Mechanism, and Wise-IoU v3	91.31%
Mitra et al.	Sentinel-2 Dataset	Unet + backbone ResNet-34	76.8%
Mitra et al.	Sentinel-2 Dataset	Random Forest (RF)	92%
Xiang et al.	UAV imagery data to perform geometric correction with position and orientation system (POS)	Average boundary distance algorithm (ABDA)	91%
Kalinaki et al.	Sentinel-2 satellite images	Attention-residual-based deep learning model with U-Net architecture	93%
Perez et al.	UAV Aerial Images	You Only Look Once (YOLO)	72%
Perez et al.	UAV Aerial Images	Mask R-CNN	82%
Sun et al.	Airborne light detection and ranging (LiDAR)	YOLOv4 with synergetic use	81.4%

A. ANALYSIS OF IDENTIFICATION LIMITATIONS

Forest stand identification employs various techniques and technologies that utilize aerial and satellite imagery. In Slovakia, the responsibility for forest monitoring typically falls to an assigned forest technician or taxidermist. This technician often conducts physical inspections of the monitoring area, collects various samples, and subsequently processes these samples using selected software.

An alternative method for monitoring forest stand involves the analysis of aerial and satellite images. This analysis is performed prior to the technicians physical inspection of the forest stand. However, the identification of tree age through satellite and aerial imagery presents certain limitations, particularly the inability to determine the age of specific trees with certainty. Despite this, the boundaries of the forest stand and other features, such as meadows or fields, can generally be delineated with a high degree of accuracy.

When comparing satellite and aerial imagery in terms of quality, availability, and zoom capabilities, particularly using the cadastral map of Slovakia as a reference, it becomes evident that aerial imagery offers superior zoom capabilities, resulting in more detailed images. Conversely, satellite imagery is more accessible. In terms of image quality, satellite images typically exhibit richer colors and sharper edges. Therefore, satellite maps are often more practical for certain applications, and with the utilization of higher quality satellite imagery, it is possible to achieve approximations that are comparable to those obtained from aerial imagery.

The annotations for the dataset were conducted using the platform roboflow.com [45], employing instance segmentation techniques. A total of five annotation classes were established: forest, field, lake, road, and building. The split of the instances of each class is shown in (Fig. 3).

The forest class encompasses instances of various tree species, including deciduous and coniferous trees, as well as tree arrangements that form forest stands, in addition to individual trees present in urban environments.

The road class encompasses all vehicular roads, excluding field roads. The lake class includes areas of water such as lakes, rivers, and streams.

FIGURE 1. Example of image, size 876 × 739 pixels.

FIGURE 2. Example of image, size 876 × 739 pixels.

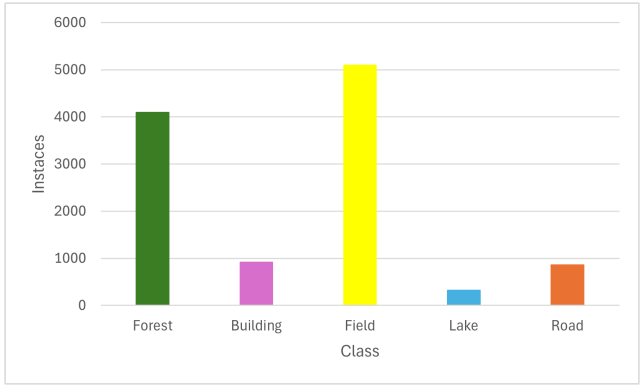


FIGURE 3. Class division.

The building class includes houses, apartment complexes, industrial structures, and other types of buildings. Lastly, the field class primarily consists of areas such as fields, meadows, pastures, vineyards, lawns, and other land types that do not fall under the forest, building, road, or lake categories.

FIGURE 4. Annotated picture with forest and field classes.

Upon the completion of the annotations for all images, the training process for each of the selected models was initiated. The models chosen for this study were YOLOv8, YOLOv5 and Mask R-CNN, all of which utilize instance segmentation for object detection. Both datasets are available on [46].

IV. IMPLEMENTATION OF MODELS FOR DATASET FOREST_FULL_V1

The training of our models was initially commenced using the Forest_Full_V1 dataset, which consists of approximately 4,000 authentic and labeled images. To enhance the dataset, data augmentation techniques were applied, including horizontal and vertical rotations of the images. Additionally, the saturation levelswas modified by ±30% and adjusted the brightness by ±25%. These augmentations significantly increased the total number of images, which were subsequently divided into training, validation, and test sets.

The training set consisted of approximately 9,800 images, while the validation set contained 600 images, and the test set included 200 images. Collectively, this dataset was designated as Forest_Full_V1. Furthermore, a second version of the dataset, named Forest_Full_V2, was developed, which was generated from 5,100 authentic images. In this version, the training set contained approximately 11,900 images, the validation set comprised around 800 images, and the test set included approximately 300 images.

For the training process, three models were selected: YOLOv8, YOLOv5, and Mask R-CNN. All of these models are categorized as convolutional neural networks, and instance segmentation was chosen as the detection approach, which is supported by each model. Object detection alone was deemed insufficient in this context, as labeling portions of images using bounding boxes is often imprecise, resulting

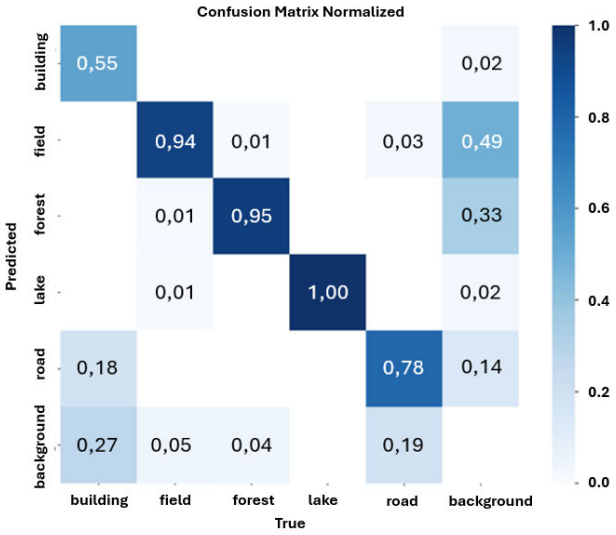


FIGURE 5. Confusion Matrix for YOLOv8 Nano with batch-size 16, after generalization for Forest_Full_V1.

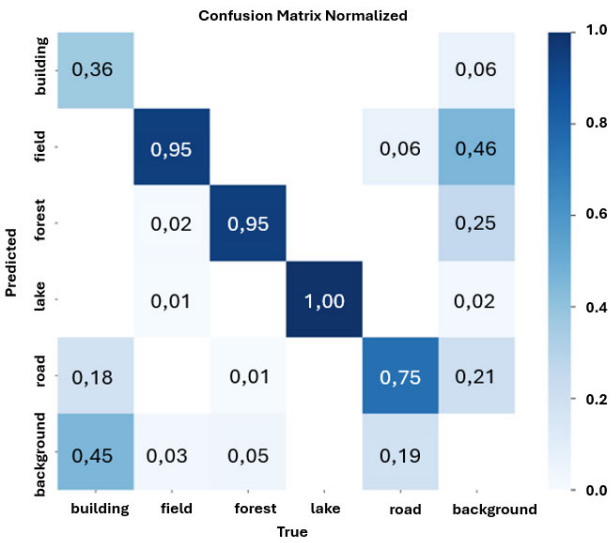


FIGURE 6. Confusion Matrix for YOLOv8 Small with batch-size 16, after generalization for Forest_Full_V1.

in inadequate and chaotic visualizations of the outcomes. Although semantic segmentation could have been employed, providing pixel-level masks for each identified class of objects in the image, instance segmentation was ultimately chosen. This approach generates a pixel-level mask for each identified object instance, facilitating a precise definition of image regions and their corresponding edges. Training of the models took place on a computer equipped with an NVIDIA GeForce RTX 4060 Ti, 64 GB RAM, and an AMD Ryzen 9 7950X processor. The time required to train each model ranged from 2-4 hours.

A. YOLOv8 WITH DATASET FOREST_FULL_V1

The YOLOv8 model was trained on the training dataset using two model sizes: Nano and Small. Subsequently, both models were validated against the validation dataset, and their performance was then generalized on the test dataset. The selected batch sizes for training were 16 and 4, respectively. These varying batch sizes were chosen primarily to enhance the precision of each class in YOLO model, as the batch size influences the number of images from which the model learns simultaneously, thereby affecting training precision. However, it is important to note that this is not a universal rule. Training with a smaller batch size does not consistently yield improved results and may, in some cases, produce outcomes comparable to or even worse than those achieved with a larger batch size. The effectiveness of the batch size is contingent upon the specific application area and the nature of the model employed.

A comparison of the training results, illustrated in (Fig. 5) and (Fig. 6), indicates that the YOLOv8 Nano model outperformed the Small model. Both models exhibited identical precision for the Forest and Lake classes, with a respective precision of 95% for the Forest class and 100% for the Lake class. The Small model, however, demonstrated a 1% higher precision for the Field class, achieving 95%. In contrast, the Nano model outperformed the Small model for the Road class, attaining an precision of 78%, which is 3% higher than that of the Small model. The most pronounced difference was observed in the Building class, where the Nano model achieved an precision of 55%, while the Small model only reached 36%. This disparity of 19% is significant. These findings highlight a critical issue regarding the imbalance in the Forest_Full_V1 dataset, which likely contributed to the lower precision observed for the Building and Road classes.

Additionally, the YOLOv8 model was trained using a smaller batch size to enhance model accuracy. With a batch size of 4, the model processed 4 images at a time. This reduction in batch size resulted in nearly double the training duration compared to training with a batch size of 16.

The training results for batch-size = 4 are presented in (Fig. 7) and (Fig. 8). Contrary to expectations, the smaller batch size did not lead to an improvement in the accuracy of the individual models. For the Building class, the Nano model achieved an precision of 45%, while the Small model reached 55%, demonstrating a decline rather than the anticipated increase in precision. Similarly, there was a reduction in precision for other classes. For instance, the Forest class precision for the Nano model dropped to 90%, which is 5% lower than the precision achieved with a batch size of 16, while the Small model achieved 92%, a decrease of 3%. In the Field class, the Nano model precision was 92%, which is 3% less than at batch-size = 16, and the Small model reached 94%, a 1% reduction compared to batch-size = 16.

In summary, after comparing all training results across model types and batch sizes, the Nano model yielded the best performance with a batch-size of 16.

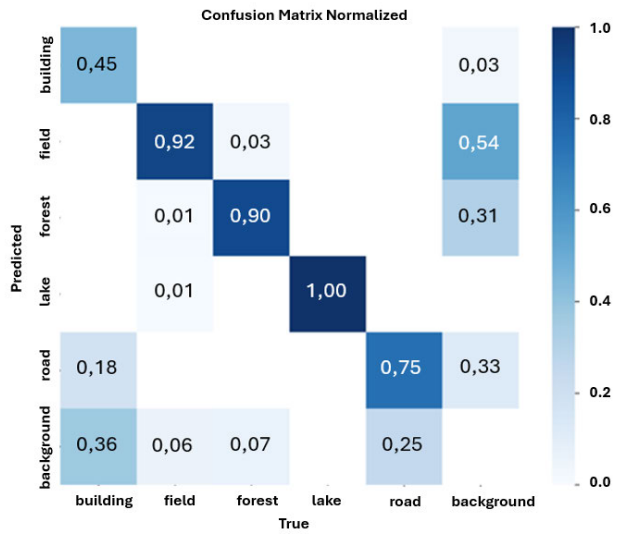


FIGURE 7. Confusion Matrix for YOLOv8 Nano with batch-size 4, after generalization for Forest_Full_V1.

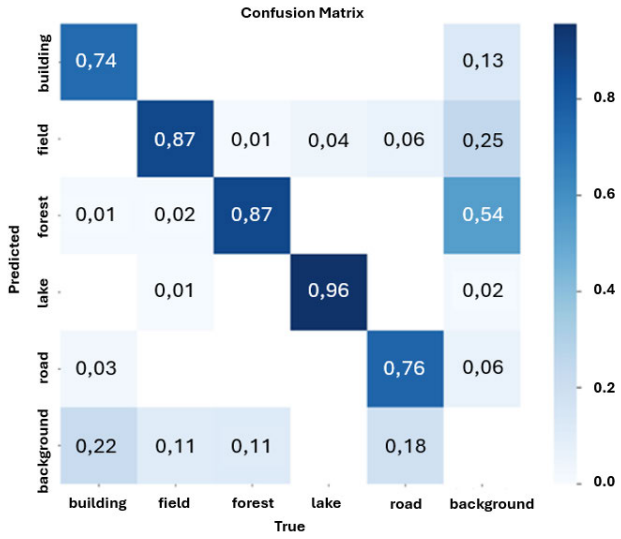


FIGURE 9. Confusion Matrix for YOLOv5 Nano with batch-size 16, after generalization for Forest_Full_V1.

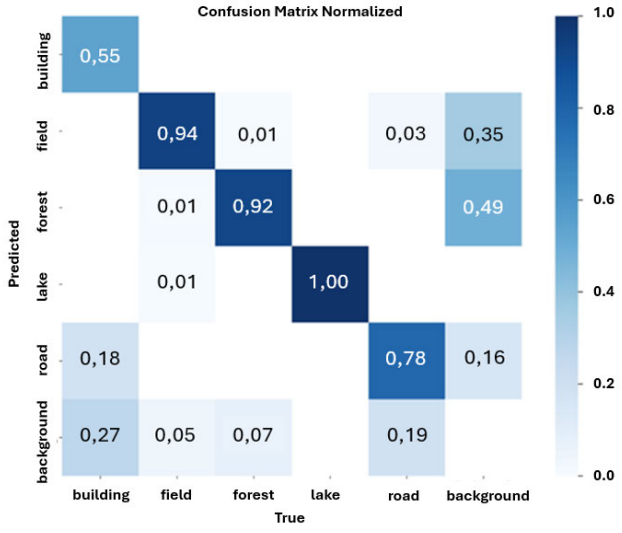


FIGURE 8. Confusion Matrix for YOLOv8 Small with batch-size 4, after generalization for Forest_Full_V1.

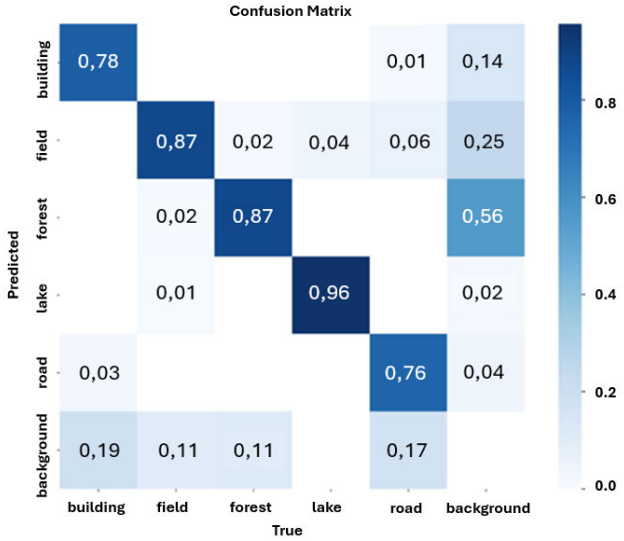


FIGURE 10. Confusion Matrix for YOLOv5 Small with batch-size 16, after generalization for Forest_Full_V1.

B. YOLOv5 WITH DATASET FOREST_FULL_V1

Similar to the approach used for YOLOv8, the YOLOv5 model was trained on the Forest_Full_V1 dataset for both Nano and Small model sizes. The models were subsequently validated on the validation dataset and generalized on the test data. YOLOv5, published in 2020, was selected to compare its performance with YOLOv8, which was released in 2023, in order to evaluate the different results produced by these two models.

The training results for YOLOv5 are depicted in (Fig. 9) and (Fig. 10). The figures illustrate the performance alignment between the two models. Although they performed nearly identically across most classes, the Small model demonstrated a 4% higher precision in the Building class. For all other classes, the precision remained the same between the two model sizes. The batch-size used in these experiments was set to 16.

Additional training of the YOLOv5 models was conducted using a reduced batch size of 4, as shown in (Fig. 11) and (Fig. 12). Similar to the YOLOv8 results, reducing the batch-size did not lead to any improvements in performance; rather, the results showed a decline. For the Nano model, the Building class precision dropped to 63%, representing an 11% decrease compared to the batch-size = 16 training. Additionally, the Road class precision declined by 4%, reaching 72%. The remaining classes saw no change. The Small model also showed no improvements, with the Building class precision decreasing by 6%, to 72%.

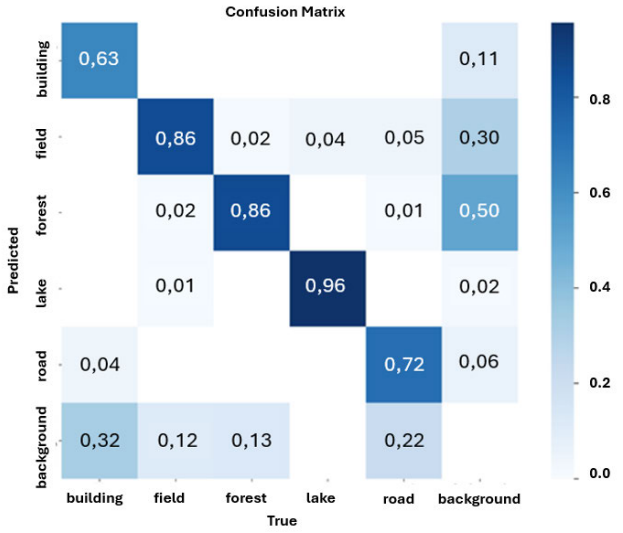


FIGURE 11. Confusion Matrix for YOLOv5 Nano with batch-size 4, after generalization for Forest_Full_V1.

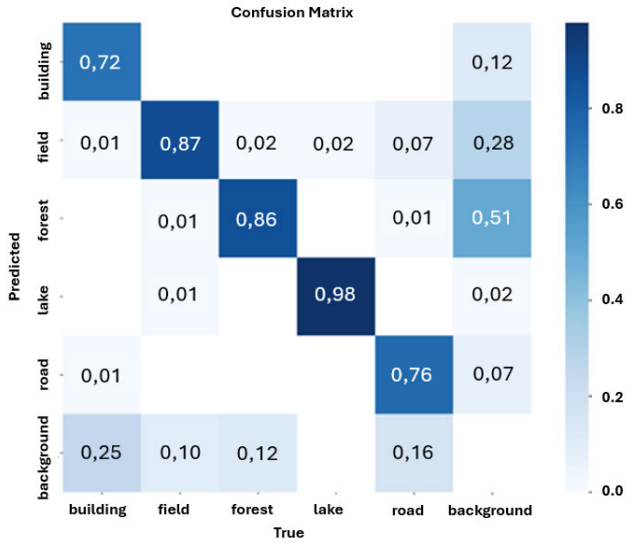


FIGURE 12. Confusion Matrix for YOLOv5 Small with batch-size 4, after generalization for Forest_Full_V1.

In conclusion, based on the training results for the YOLOv5 model on the Forest_Full_V1 dataset, the best results was achieved by the Small model with a batch-size of 16.

C. MASK-RCNN WITH DATASET FOREST_FULL_V1

The Mask R-CNN model was trained using the Detectron-2 library, with ResNet-101 and ResNet-50 backbones to enhance the performance of instance segmentation tasks. A total of 5000 iterations were conducted, with validation performed after every 200 iterations. The training was executed using batch sizes of 16 and 4. To visualize the

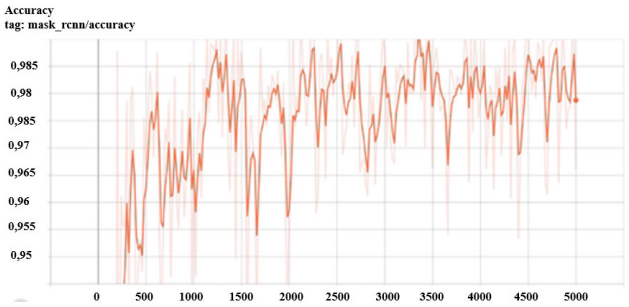


FIGURE 13. Accuracy of the Mask-RCNN model with backbone Resnet-101 and batch-size 16 for Forest_Full_V1.

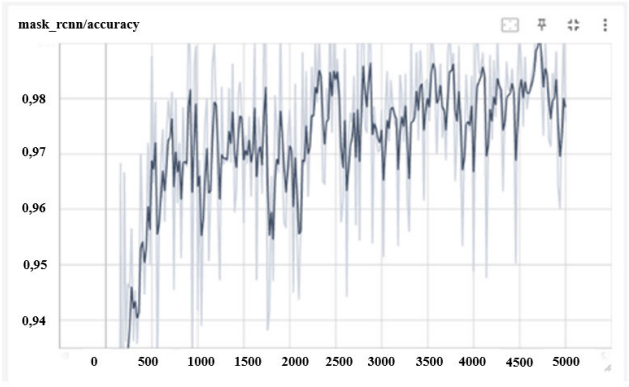


FIGURE 14. Accuracy of the Mask-RCNN model with backbone Resnet-50 and batch-size 16 for Forest_Full_V1.

training progress and results, TensorBoard was employed, which provides specialized features for monitoring machine learning models and assessing their performance using various metrics [47].

Based on the TensorBoard results, displayed in (Fig. 13) and (Fig. 14), the training precision for both ResNet-101 and ResNet-50 backbones ranged between 95% and 98%.Although the model showed good accuracy but was not able to work well on the test images. Examples of the models performance on test images can be seen in (Fig. 15), (Fig. 16), (Fig. 17).

The Mask-RCNN model was also trained with a batch size of 4. The resulting accuracy ranged between 95% and 99% for both models, as shown in (Fig. 18) and (Fig. 19). However, similar to the YOLO models, there was no significant improvement in accuracy. In fact, the accuracy deteriorated slightly, as the models exhibited larger fluctuations in performance compared to those trained with a batch size of 16. Additionally, the models fit on the test data was not as robust. The class predictions were less accurate, and the instances of class overlap or incomplete labeling of image segments were more frequent. After evaluating all models, the best results were achieved by the Mask-RCNN model with the ResNet-101 backbone at a batch size

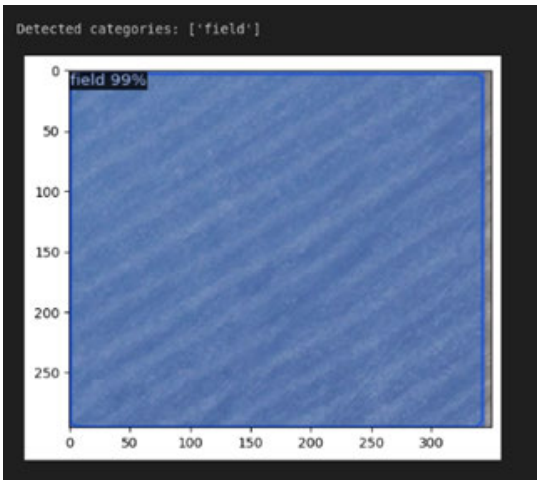


FIGURE 15. Defined class Field using Mask-RCNN with backbone Resnet-50 and batch-size 16 with omitted part of the image for Forest_Full_V1.

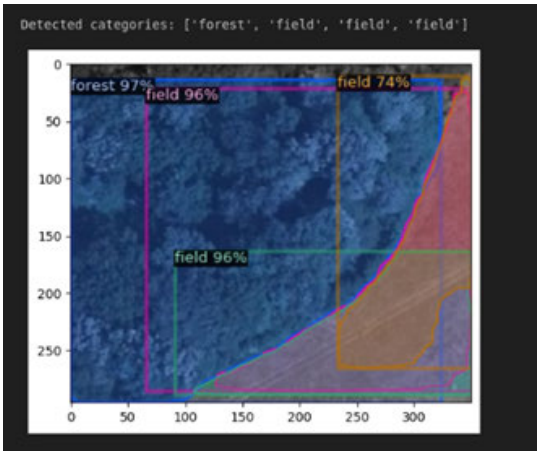


FIGURE 16. Defined Field and Forest classes using Mask-RCNN with backbone Resnet-101 and batch-size 16 with overlay classes for Forest_Full_V1.

of 16. The deployment results of this model are illustrated in (Fig. 20) and (Fig. 21).

V. IMPLEMENTATION OF MODELS FOR DATASET FOREST_FULL_V2

Based on the training results from the initial dataset, the non-uniformity of the dataset was assessed. It was observed that the dataset contained a limited number of images representing instances of buildings and roads. To address this deficiency, the dataset was enriched with an additional 1,100 images featuring primary buildings and roads. Training on the Forest_Full_V2 dataset was conducted with batch sizes of 16 and 4. Especially, the previous training of the models on the Forest_Full_V1 dataset using a smaller batch size did not yield improvements in model accuracy.

Following the modification of the dataset and the expansion of the image count, the dataset is now more balanced in terms of the number of instances for each class. Consequently,

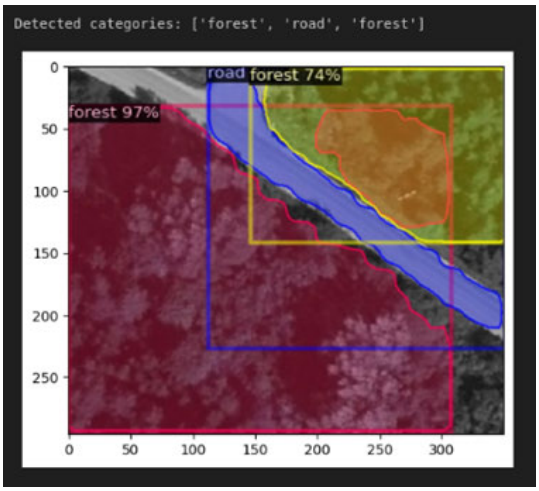


FIGURE 17. Defined Forest and Road classes using Mask-RCNN with backbone Resnet-50 and batch-size 16 with omitted part of the image and overlaid classes for Forest_Full_V1.

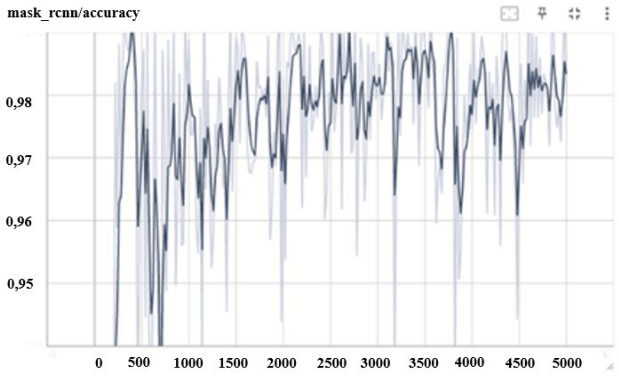


FIGURE 18. Accuracy of the Mask-RCNN model with Resnet-101 backbone and batch-size 4 for Forest_Full_V1.

training with a batch size of 4 was also executed, as it was anticipated that this might lead to enhancements in training outcomes. Again for this training, three models were selected: YOLOv8, YOLOv5, and Mask-RCNN.

A. YOLOv8 WITH DATASET FOREST_FULL_V2

Further training of the YOLOv8 model was conducted on the updated training dataset, utilizing both the Nano and Small model sizes. Following the training phase, validation was performed on the validation dataset, and the models were subsequently generalized on the test dataset. As in previous experiments, the selected batch sizes were 16 and 4.

In (Fig. 22) and (Fig. 23), the performance of the Nano and Small YOLOv8 models, trained with a batch size of 16, is depicted. A comparison of the models reveals very similar precision results. For the Building class, the Nano model achieved an precision of 84%, while the Small model exhibited a slightly lower precision of 80%, a difference of 4%. Conversely, for the Field class, the Small model demonstrated better performance with 92% precision,

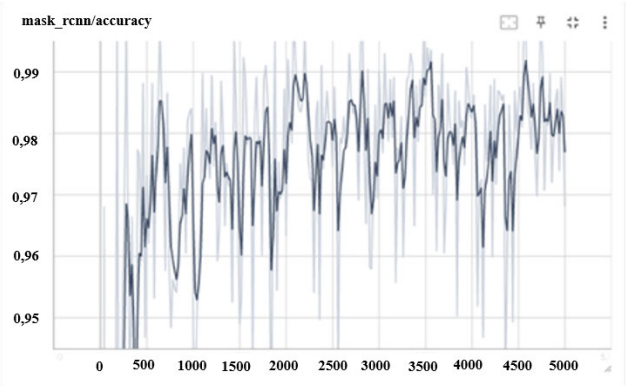


FIGURE 19. Accuracy of the Mask-RCNN model with backbone Resnet-50 and batch-size 4 for Forest_Full_V1.

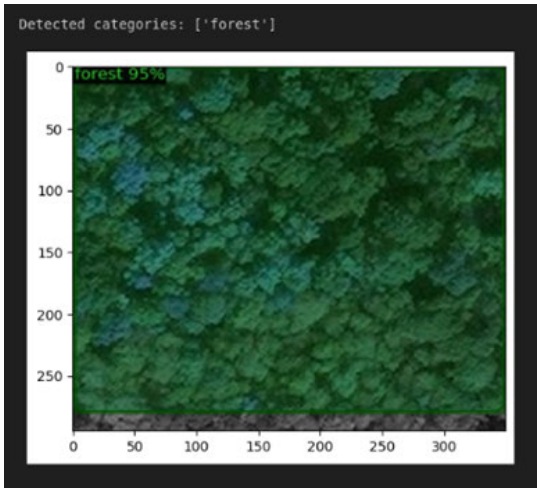


FIGURE 20. Specified Forest class, using Mask-RCNN with backbone Resnet-101 with the omitted part of the image for Forest_Full_V1.

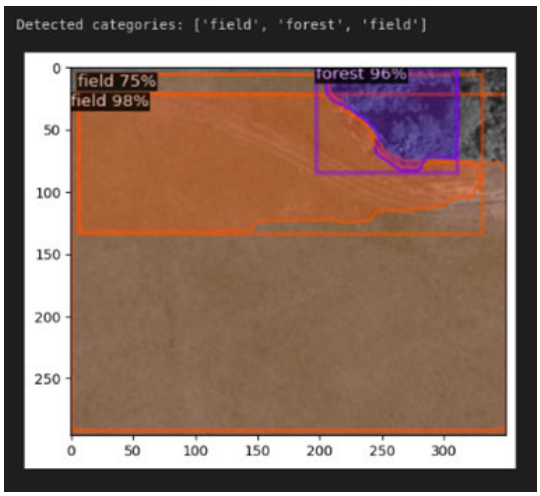


FIGURE 21. Specified Forest and Field class using Mask-RCNN with backbone Resnet-101 with omitted part of the image and overlaid classes for Forest_Full_V1.

outperforming the Nano model, which achieved 89%, a 3% difference.

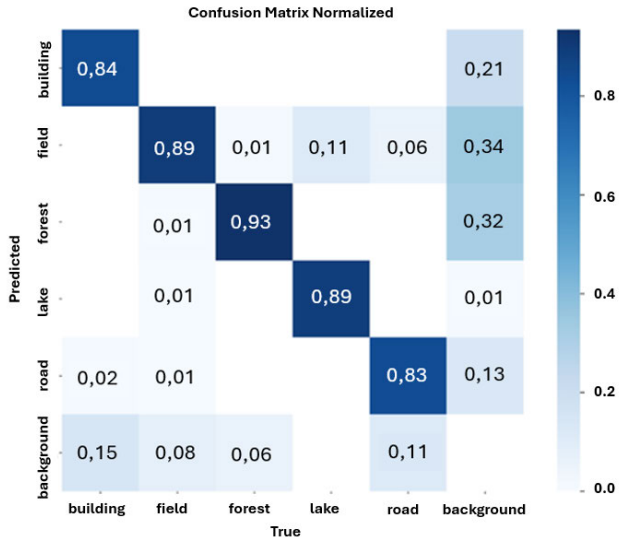


FIGURE 22. Confusion Matrix for YOLOv8 Nano with batch-size 16, after generalization for Forest_Full_V2.

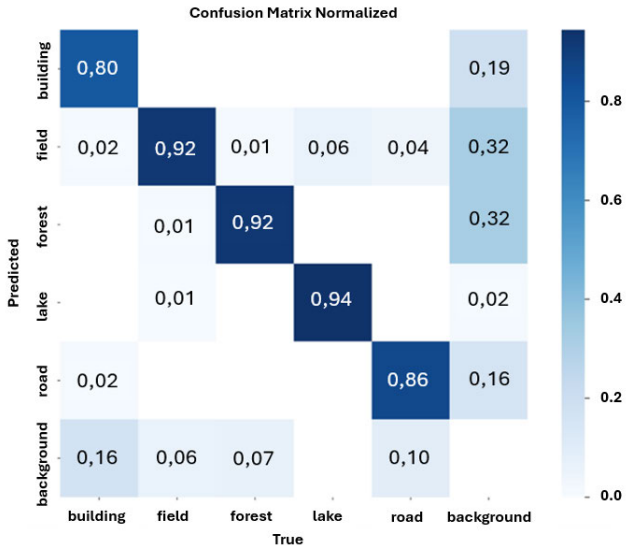


FIGURE 23. Confusion Matrix for YOLOv8 Small with batch-size 16, after generalization for Forest_Full_V2.

The Forest class, which is of primary interest, showed nearly identical accuracy for both models, with the Nano model achieving 93% and the Small model at 92%. Both models performed equally well in the Lake class, achieving 94% precision. For the Road class, however, the Small model outperformed the Nano model, achieving 86% precision compared to the Nano model 83%, representing a 3% difference.

Training with a lower batch size did not result in a significant improvement in overall accuracy, as shown in (Fig. 24) and (Fig. 25). For the Nano model with a batch size of 4, the precision for the Building class was 85%, while the Small model achieved 82%. In the Field class, both models performed identically, with a precision of 91%. The precision

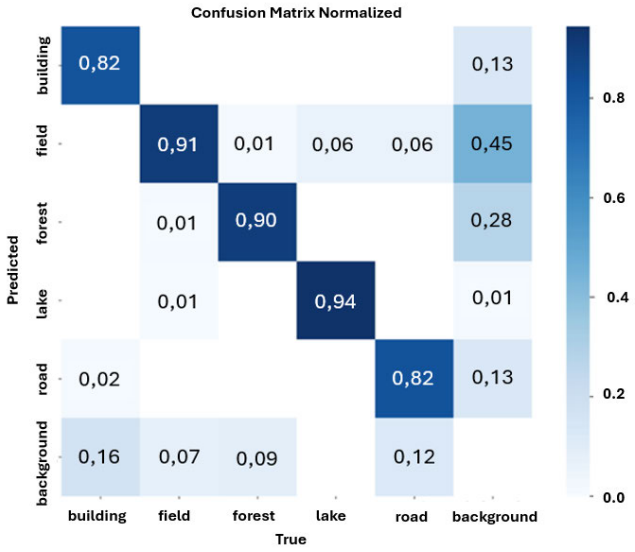


FIGURE 24. Confusion Matrix for YOLOv8 Nano with batch-size 4, after generalization for Forest_Full_V2.

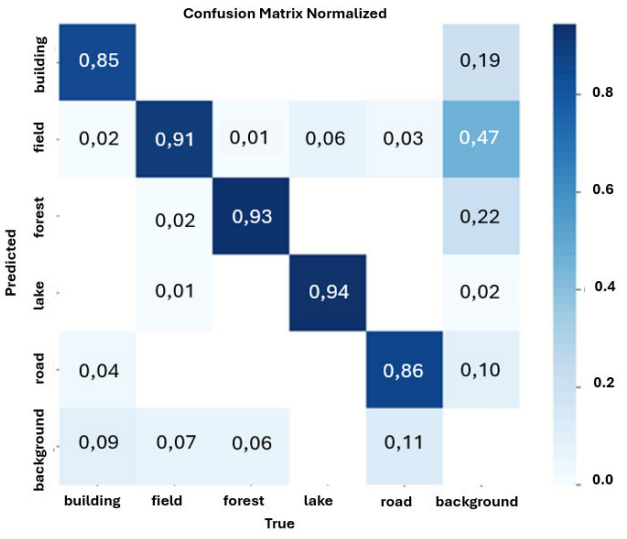


FIGURE 25. Confusion Matrix for YOLOv8 Small with batch-size 4, after generalization for Forest_Full_V2.

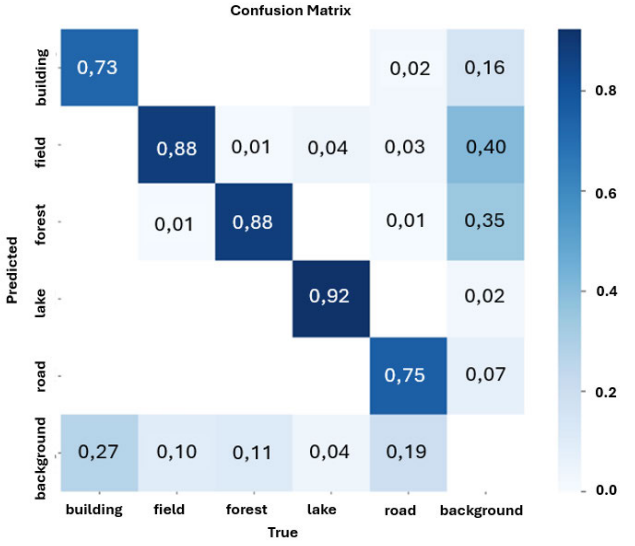


FIGURE 26. Confusion Matrix for YOLOv5 Nano with batch-size 16, after generalization for Forest_Full_V2.

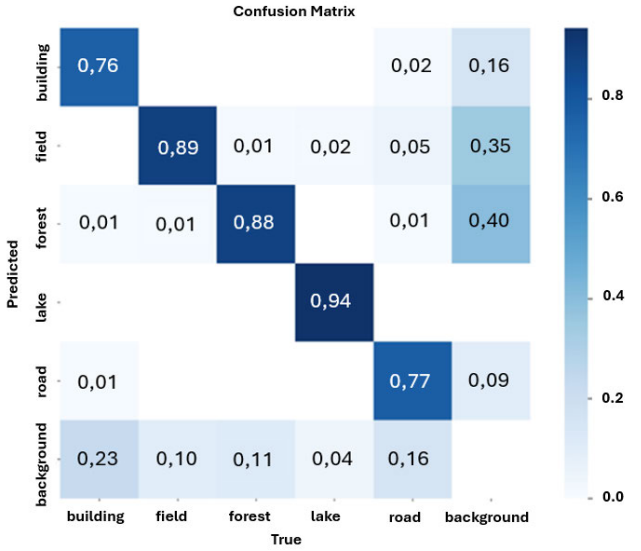


FIGURE 27. Confusion Matrix for YOLOv5 Small with batch-size 16, after generalization for Forest_Full_V2.

for the Forest class was 90% for the Nano model, whereas the Small model achieved 93%. For the Lake class, both models again produced identical results, each achieving 94% precision. In the Road class, the Small model outperformed the Nano model, achieving a precision of 86% compared to the Nano model 82%.

Overall, the best performance was observed with the YOLOv8 Small model trained with a batch size of 4.

B. YOLOv5 WITH DATASET FOREST_FULL_V2

The YOLOv5 model was retrained in a manner consistent with previous protocols, utilizing the training data for both the Nano and Small model sizes. Subsequently, validation was conducted on the respective validation datasets, followed by generalization on the test data. Finally, a comparative analysis of the results from all training runs of the YOLOv5 model was performed.

When comparing the Nano and Small models at a batch size of 16, as illustrated in (Fig. 26) and (Fig. 27), it is evident that the performance results are quite similar. Specifically, the Nano model achieved a precision of 73% for the Building class, while the Small model reached an precision of 76%. For the Field class, the Nano model recorded an precision of 88%, compared to 89% for the Small model, resulting in a negligible difference of 1%. Both models exhibited identical precision of 88% for the Forest class. In terms of the Lake class, the Nano model achieved an precision of 92%, whereas

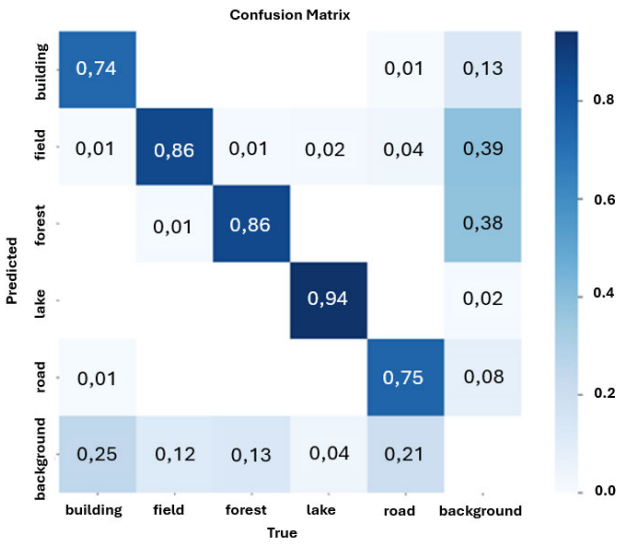


FIGURE 28. Confusion Matrix for YOLOv5 Nano with batch-size 4, after generalization for Forest_Full_V2.

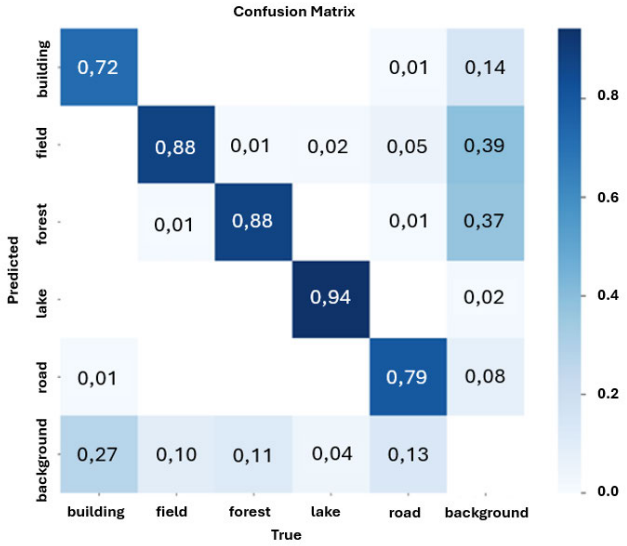


FIGURE 29. Confusion Matrix for YOLOv5 Small with batch-size 4, after generalization for Forest_Full_V2.

the Small model attained 94%. Similarly, for the Road class, the Small model scored 77%, while the Nano model achieved 75%.

In the context of a batch size of 4, the training duration was approximately doubled, as illustrated in (Fig. 28) and (Fig. 29). For the Building class, the Nano model attained a precision of 74%, whereas the Small model achieved a precision of 72%. The accuracies for the Field and Forest classes were identical in the Nano model, both recorded at 86%. In contrast, the Small model exhibited a slight improvement, achieving a precision of 88% for these classes. The accuracy for the Lake class remained consistent across both models at 94%. Additionally, the Small model outperformed the Nano model in the Road class, achieving a precision of 79% compared to the Nano model 75%. In summary, the optimal results were observed for the Small model when utilizing a batch size of 16.

C. MASK-RCNN WITH DATASET FOREST_FULL_V2

The Mask R-CNN model was trained once again utilizing the Detectron-2 library, incorporating Resnet-101 and Resnet-50 backbones. The model underwent a total of 5000 iterations, with validation conducted after every 200 iterations. The selected batch sizes were 16 and 4, and TensorBoard was employed to visualize the training results.

As indicated by the results obtained from TensorBoard, presented in (Fig. 30) and (Fig. 31), the accuracy of both models ranged from 94% to 98% for Resnet-101 and Resnet-50. Following the training process, the model was applied to the test data.Similar to the Forest_Full_V1 dataset, the model responds to test images accordingly. The outcomes of class determination on the test data are illustrated in (Fig. 32) and (Fig. 33).

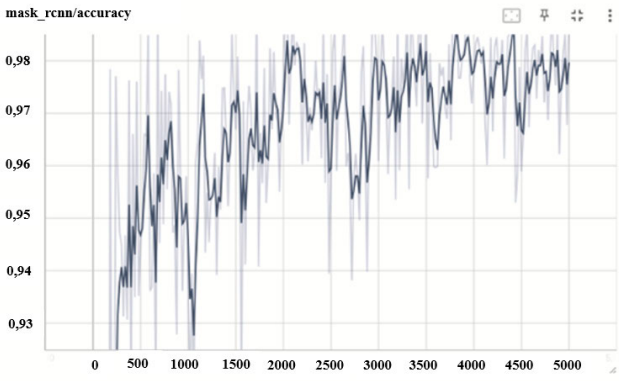


FIGURE 30. Accuracy of the Mask-RCNN model with backbone Resnet-101 and batch-size 16 for Forest_Full_V2.

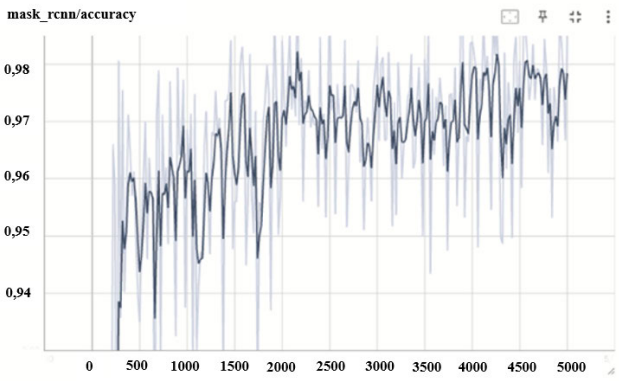


FIGURE 31. Accuracy of the Mask-RCNN model with backbone Resnet-50 and batch-size 16 for Forest_Full_V2.

The Mask R-CNN model was trained with a batch size of 4 in a manner consistent with the previous training protocol. The accuracy results for this training iteration

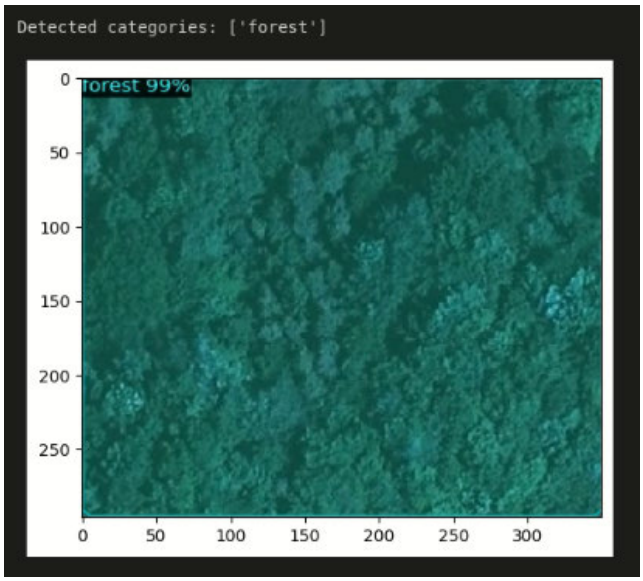


FIGURE 32. Correctly determined Forest class using Mask-RCNN with backbone Resnet-101 for Forest_Full_V2.

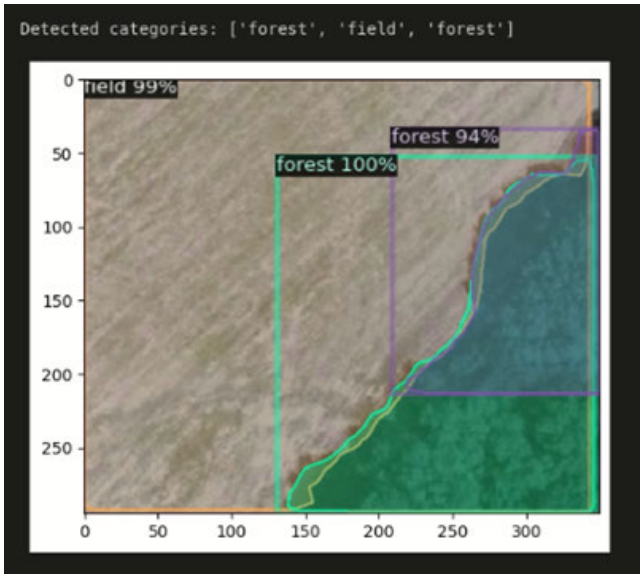


FIGURE 33. Defined class Forest and Field using Mask-RCNN with backbone Resnet-50 with overlaid classes for Forest_Full_V2.

are presented in (Fig. 34) and (Fig. 35). Unfortunately, this training did not yield the expected results, as there was no improvement observed compared to prior iterations. Additionally, the duration of training for this model was equivalent to that of the batch size of 16. The models fitting and performance metrics were also nearly identical to those achieved with the larger batch size. Issues such as mask underfitting and class overlapping persisted. The results of the model deployment on the test images are illustrated in (Fig. 36) and (Fig. 37). Upon evaluating the training outcomes on the Forest_Full_V2 dataset, it was determined that the

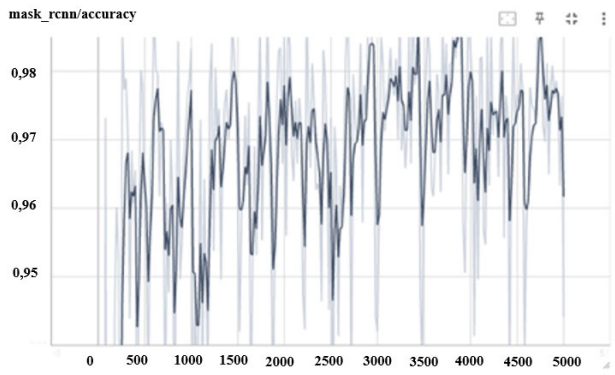


FIGURE 34. Accuracy of the Mask-RCNN model with Resnet-101 backbone and batch-size 4 for Forest_Full_V2.

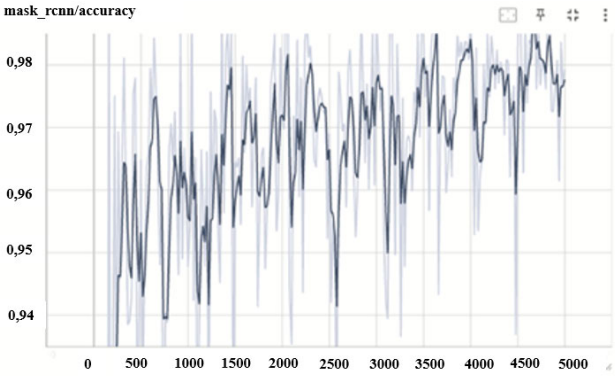


FIGURE 35. Accuracy of the Mask-RCNN model with backbone Resnet-50 and batch-size 4 for Forest_Full_V2.

Mask R-CNN model with the Resnet-50 backbone at a batch size of 16 produced the best results.

VI. COMPARING MODELS TRAINED ON FOREST_FULL_V1 AND FOREST_FULL_V2 DATASETS

When comparing the performance of individually trained models on the Forest_Full_V1 and Forest_Full_V2 datasets for the selected YOLOv8, YOLOv5, and Mask-RCNN architectures, notable differences in the precision of class predictions are observed.

A. COMPARING YOLOv8 MODELS

YOLOv8 models trained on the Forest_Full_V1 dataset achieved a precision of 36%-55% for the Building class, 92%-94% for the Field class, the main Forest class achieved an accuracy of 90%-95%, the Lake class achieved 100% for all models, and the Road class achieved 75%-78%.

YOLOv8 models trained on the Forest_Full_V2 dataset achieved higher accuracy values compared to the Forest_Full_V1 dataset. Specifically, the Building class achieved 80%-85%, the Field class achieved 89%-92%, the main Forest class achieved an accuracy of 90%-93%, the Lake class achieved 89%-94%, and the Road class achieved 82%-86%.

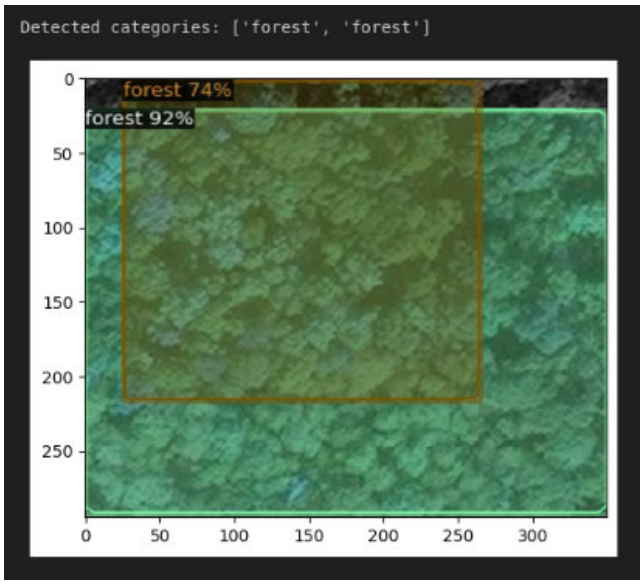


FIGURE 36. Identified Forest class using Mask-RCNN with backbone Resnet-50 with omitted part of the image and overlaid classes for Forest_Full_V2.

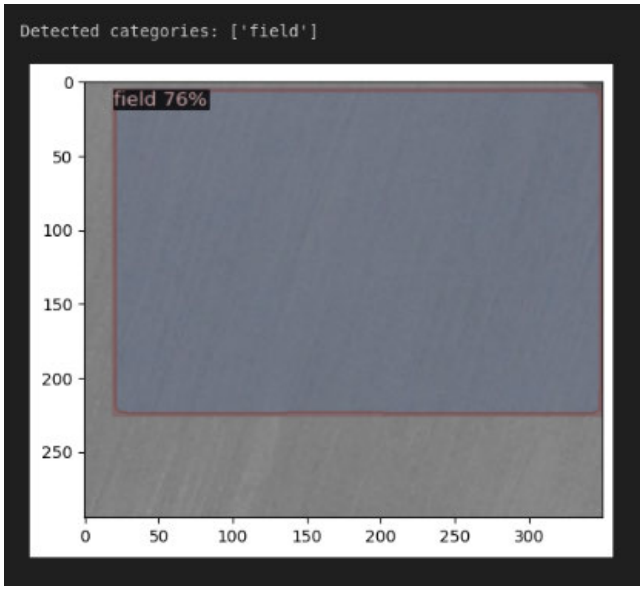


FIGURE 37. Identified Field class using Mask-RCNN with backbone Resnet-101 with omitted part of the image for Forest_Full_V2.

A comparison of the performance of YOLOv8 models trained on the Forest_Full_V1 and Forest_Full_V2 datasets reveals notable improvements in accuracy for several classes in the Forest_Full_V2 dataset.

For the Building class, the models trained on the Forest_Full_V1 dataset achieved a precision range of 36%-55%, whereas models trained on the Forest_Full_V2 dataset showed a significant improvement, achieving 80%-85%.

In the Field class, the Forest_Full_V1 models performed with a precision of 92%-94%, while the Forest_Full_V2 models achieved slightly lower but still strong performance, ranging from 89%-92%.

For the main Forest class, the accuracy remained relatively consistent, with the Forest_Full_V1 models achieving 90%-95% and the Forest_Full_V2 models performing within a slightly narrower range of 90%-93%.

The Lake class achieved perfect accuracy (100%) for all models trained on the Forest_Full_V1 dataset, but models trained on the Forest_Full_V2 dataset showed a slight decrease, with accuracy ranging from 89%-94%.

Finally, for the Road class, the Forest_Full_V1 models achieved an accuracy of 75%-78%, while the models trained on the Forest_Full_V2 dataset performed better, achieving 82%-86%.

Overall, YOLOv8 models trained on the Forest_Full_V2 dataset demonstrated higher accuracy for most classes, with notable improvements in the Building and Road classes, although the accuracy for the Lake class saw a slight decrease.

B. COMPARING YOLOv5 MODELS

YOLOv5 models trained on the Forest_Full_V1 dataset achieved accuracy values of 63%-78% for the Building class, 86%-87% for both the Field and Forest classes, 96%-98% for the Lake class, and 72%-76% for the Road class.

YOLOv5 models trained on the Forest_Full_V2 dataset achieved accuracy values of 72%-76% for the Building class, 86%-89% for the Field class, 86%-88% for the main Forest class, 92%-94% for the Lake class, and 75%-79% for the Road class.

When comparing the performance of YOLOv5 models trained on the Forest_Full_V1 and Forest_Full_V2 datasets, several differences in accuracy across the classes are evident.

For the Building class, YOLOv5 models trained on the Forest_Full_V1 dataset achieved accuracy values ranging from 63%-78%, while models trained on the Forest_Full_V2 dataset showed an improvement, with accuracy ranging from 72%-76%.

In the Field class, the accuracy remained consistent, with models trained on both datasets achieving values between 86%-87% for the Forest_Full_V1 dataset and 86%-89% for the Forest_Full_V2 dataset.

For the main Forest class, the models trained on the Forest_Full_V1 dataset achieved accuracy values of 86%-87%, while models trained on the Forest_Full_V2 dataset performed slightly better, with accuracy ranging from 86%-88%.

In the Lake class, models trained on the Forest_Full_V1 dataset achieved high accuracy levels of 96%-98%, while models trained on the Forest_Full_V2 dataset showed a slight decrease, with accuracy ranging from 92%-94%.

Finally, for the Road class, the accuracy improved in the Forest_Full_V2 dataset, with models achieving values of 75%-79%, compared to 72%-76% for the Forest_Full_V1 dataset.

Overall, YOLOv5 models trained on the Forest_Full_V2 dataset showed modest improvements in accuracy for the Building, Field, and Road classes, while the Lake class

experienced a slight drop in accuracy. The Forest class remained relatively consistent across both datasets.

C. COMPARING MASK-RCNN MODELS

Mask-RCNN models trained on the Forest_Full_V1 dataset with the ResNet-50 and ResNet-101 backbones achieved precision values ranging from 95% to 98%.

In comparison, Mask-RCNN models trained on the Forest_Full_V2 dataset with the same backbones (ResNet-50 and ResNet-101) achieved precision values ranging from 94% to 98%.

Overall, the performance of Mask-RCNN models on both datasets remained similar, with a slight decrease in precision observed for the Forest_Full_V2 dataset, particularly in the lower range of precision.

As previously mentioned issues such as mask underfitting and class overlapping persisted in both datasets. Despite the improvements in precision across the different models, these challenges continued to affect the overall performance of the Mask-RCNN models. Mask underfitting, where the predicted masks fail to accurately cover the target objects, and class overlapping, where different classes share similar visual features and are incorrectly detected as the same, remained persistent problems that influenced the accuracy of the models.

VII. IMPLEMENTATION OF MODEL ON UNLABELLED DATA

From the trained models, the YOLOv8 model with a Small size and a batch size of 4 was selected, which was trained on the Forest_Full_V2 dataset. This model was chosen due to its optimal precision across all classes, with less significant differences in precision compared to the other trained models.

The YOLOv8 models outperformed both YOLOv5 and Mask-RCNN in terms of accuracy. Among the selected YOLOv8 models, the one with a Small size and a batch size of 4 achieved the highest accuracy for the main Forest class, reaching 93%, surpassing the other YOLOv8 models. While the YOLOv8 model with a Nano size and a batch size of 16 also achieved the same accuracy for the main Forest class, its performance for other classes was noticeably lower. Specifically, the YOLOv8 with a Nano size and a batch size of 16 achieved an accuracy of 89% for the Field class, 84% for the Building class, 89% for the Lake class, and 83% for the Road class. In contrast, the YOLOv8 model with a Small size and a batch size of 4 achieved higher accuracy values. 91% for the Field class, 85% for the Building class, 94% for the Lake class, and 86% for the Road class. Although the differences in accuracy are not highly significant, the YOLOv8 model with a Small size and a batch size of 4 still achieved the best overall performance among the YOLOv8 models.

The YOLOv8 model with a Small size and a batch size of 4 was applied to a dataset comprising 48 unlabeled images. Upon examination, the model correctly identified 44 images while misclassifying 4 images, resulting in an overall detection accuracy of 91.67% across the 48 images. Given the lack of labels, it was not possible to extract a confusion matrix from the detection results. Consequently, it was essential to analyze and evaluate the results obtained from this model.

FIGURE 38. Unlabelled images on which the model made a prediction.

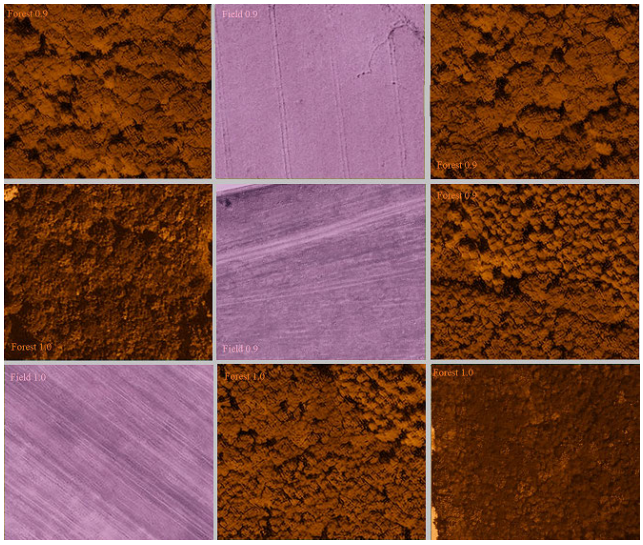


FIGURE 39. Final image prediction.

VIII. HYPERPARAMETER TUNING

In machine learning, a hyperparameter is a configuration variable that determines certain aspects of a model's learning process and overall behavior. These are values set before the training begins, and they are not learned from the data like model parameters (e.g., weights and biases in neural networks). Instead, hyperparameters influence how the training algorithm works and how the model structure is defined.

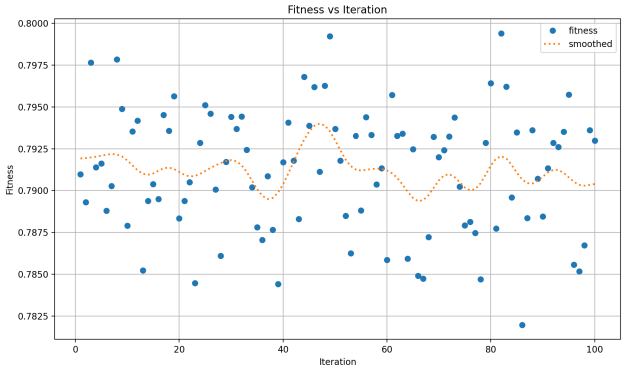


FIGURE 40. Hyperparameter - fitness vs. iteration.

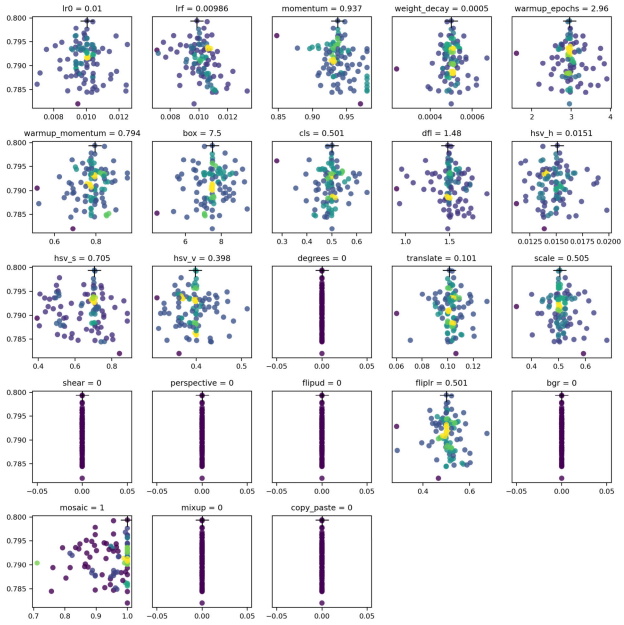


FIGURE 41. Hyperparameter - Scatter plots.

Hyperparameters can broadly be classified into two categories: model hyperparameters and algorithm hyperparameters. Model hyperparameters define the architecture or structure of the model. Examples include the number of layers in a neural network, the number of neurons in each layer, and the type of activation functions used. These hyperparameters directly influence the capacity of the model to capture patterns in the data.

Algorithm hyperparameters, on the other hand, control the training process itself. Examples include the learning rate, which determines how quickly the model updates its parameters during training; the batch size, which dictates how many samples are processed at a time; and the number of epochs, which specifies how many times the model sees the entire training dataset.

In addition to the standard training process, we performed hyperparameter tuning to further optimize the YOLO model's performance. Hyperparameter tuning plays a critical role in enhancing model accuracy and convergence speed, as it allows the selection of the most suitable configuration for the dataset. For this purpose, we utilized the Forest_Full_V2 dataset. The number of iterations was set to 100, and the tuning process spanned one week. The optimal hyperparameter values were obtained at the 82nd iteration. The tuning results are presented in Fig. 40 and Fig. 41 with hyperparameters in Table 3. It is important to note that hyperparameters initialized to 0, such as degrees or shear, are excluded from the tuning process and remain unchanged.

TABLE 3. Hyperparameters and their values used in the model.

Parameter	Value
lr0	0.01004
lrf	0.00986
momentum	0.937
weight_decay	0.0005
warmup_epochs	2.95642
warmup_momentum	0.7941
box	7.5
cls	0.50077
dfl	1.48011
hsv_h	0.0151
hsv_s	0.70477
hsv_v	0.39803
degrees	0.0
translate	0.10131
scale	0.50505
shear	0.0
perspective	0.0
flipud	0.0
fliplr	0.50062
bgr	0.0
mosaic	1.0
mixup	0.0
copy_paste	0.0

IX. DISCUSSION

The Forest_Full_V2 dataset was developed to address the limited representation of the Building and Road classes in the Forest_Full_V1 dataset. The insufficient number of instances for these classes in the Forest_Full_V1 dataset adversely impacted the training outcomes for the models. For instance, YOLOv8 achieved an accuracy range of 36% to 55% for the Building class following training on the Forest_Full_V1 dataset.

The YOLO models achieved reasonable results across both datasets. Also YOLO models were also able to detect parts of the test images decently. On the Forest_Full_V1 dataset, both models underperformed in the Building and Road classes. However, on the Forest_Full_V2 dataset, where additional instances of these classes were included, YOLOv8 demonstrated significant improvement, particularly in these challenging classes. In contrast, YOLOv5 showed only a minor improvement on the Forest_Full_V2 dataset.

When the Mask-RCNN models were applied to the test data, a significant drop in performance was observed. The models struggled with accurate class identification and mask generation, often misclassifying objects or omitting portions

of the image, which negatively impacted overall accuracy. Additionally, overlapping predictions of the same class further reduced the clarity and effectiveness of the detections.

Compared to the YOLO models, the Mask-RCNN models demonstrated inferior class determination and accuracy. While they successfully labeled certain images, they frequently failed to fully identify others. Inconsistent application of labeling masks resulted in omissions and overlaps among multiple classes, leading to cluttered outputs and a reduction in overall performance.

An overall accuracy of 91.67% was achieved, with the precision of the Forest class reaching 93%.

Our final accuracy has better value than [35], [37], [41], and [43] in 3 of 4 models. We have achieved worse results than [27], [28], [33], [34], [38], [42], and [43] in 1 of 4 models. These comparisons are made in tabular form in Table 1 and Table 2. The worse results could be due to the size and quality of the dataset, mislabeling of the individual classes in the images, or the disproportion of the individual instances that contained the images in the dataset. Another possibility for why the model did not achieve better results is a setting of the model itself, such as the batch-size value. Adding images to balance the number of instances in the dataset would optimize the dataset, and adjusting the batch-size to a different value such as 8 or 32 may also improve the overall accuracy of the model.

The resulting precision achieved was better than [29], [32], [34], [39], and [40]. Same results of precision reached Kalinaki et al. and better precision reached only Schürholz et al. Precision has reached good values, but value can be improved by adding images with instances that contain forests.

This study contributions are underscored by the development of a robust, custom dataset and the application of an instance segmentation approach, distinguishing it from prior research that predominantly utilized semantic segmentation techniques. Furthermore, the proposed methodology demonstrates the potential for broad applications in forest monitoring, enabling forest technicians to integrate AI-based solutions into their workflows effectively.

While the methodology and results are promising, they also highlight areas for improvement. These include optimizing dataset composition, enhancing model configurations such as batch size, and exploring advanced augmentation techniques to improve both accuracy and precision. Such refinements could significantly enhance the utility and reliability of the approach for large-scale forest cover detection.

X. CONCLUSION

A custom dataset comprising 4,000 authentic images was developed, designated as Forest_Full_V1. Augmentation techniques were applied to expand the dataset, thereby facilitating the training of the YOLOv8, YOLOv5, and Mask-RCNN models. The results obtained from training highlighted the non-uniformity of the dataset, with a notable lack of instances for the Building and Road classes, which contributed to lower precision rates. Conversely, there were sufficient instances of the Forest, Field, and Lake classes, all of which achieved precisions exceeding 90%.

This research contributes to the field of AI-based land-use classification by developing a custom satellite image dataset tailored to forest monitoring, addressing the need for a more diverse and balanced representation of forest-related features, such as buildings and roads.

For the YOLOv8 model, optimal results were obtained with the Nano variant when the batch size was set to 16. Similarly, the YOLOv5 model yielded the best results with the Small variant at the same batch size of 16. The Mask-RCNN model exhibited the highest performance with the Resnet-101 backbone, also using a batch size of 16.

In response to the observed deficiency of instances in the Building and Road classes within the Forest_Full_V1 dataset, an additional 1,100 authentic images were incorporated, resulting in a second version of the dataset containing 5,100 images. This updated dataset was designated as Forest_Full_V2, and augmentation techniques were once again applied to enhance the training set. Training results on the Forest_Full_V2 dataset demonstrated noticeable improvements, with overall accuracy for all classes surpassing that of the previous dataset, achieving precisions greater than 80%.

The findings of this study have significant practical implications, as the developed models can be employed for real-time forest monitoring and land-use analysis, benefiting forest managers and environmental researchers who require accurate and efficient automated methods.

In this training cycle, the YOLOv8 model achieved its best performance with the Small variant at a batch size of 4. For the YOLOv5 model, the Small variant again provided optimal results at a batch size of 16. The Mask-RCNN model exhibited the best performance with the Resnet-50 backbone at a batch size of 16.

When comparing the training outcomes of all models, the YOLOv8 Small model trained on the Forest_Full_V2 dataset with a batch size of 4 demonstrated the highest precision. This model was subsequently deployed on 48 unlabeled images to assess its performance in detecting various elements within the images. Due to the absence of labels, a confusion matrix could not be generated. So, the detection results were analyzed qualitatively.

Examples of the models detections are illustrated in (Fig. 38) and (Fig. 39). The YOLOv8 and YOLOv5 models are both suitable for similar types of problems, with YOLOv8 being the more effective choice. Although YOLOv5 produced adequate results, they were inferior to those achieved by YOLOv8. In contrast, the Mask-RCNN model encountered significant challenges in accurately identifying and labeling image segments.

By integrating advanced techniques such as YOLOv8, YOLOv5, and Mask-RCNN, this research introduces an innovative approach to forest detection, offering a versatile solution for automated detection in satellite imagery.

This research contributes valuable insights into the optimization of forest detection models through dataset composition and model configuration. The improvements seen with the Forest_Full_V2 dataset demonstrate the importance of class balance, especially for underrepresented classes like Building and Road. Moreover, the choice of YOLOv8 Small variant with a batch size of 4 proved to be the most effective for high precision, positioning it as a strong candidate for real-world applications, such as forest monitoring and land-use classification.

Future work may involve further refinement of the dataset by addressing potential label inconsistencies and exploring additional model architectures or configurations to enhance detection accuracy. Furthermore, the model's performance could be validated on larger and more diverse datasets to assess its robustness and generalization ability. Such improvements would strengthen the model's applicability in diverse environmental contexts.

REFERENCES

[1] P. Baldrian, R. López-Mondéjar, and P. Kohout, "Forest microbiome and global change," *Nature Rev. Microbiol.*, vol. 21, no. 8, pp. 487–501, Aug. 2023.
[2] K. Guo, B. Wang, and X. Niu, "A review of research on forest ecosystem quality assessment and prediction methods," *Forests*, vol. 14, no. 2, p. 317, Feb. 2023.
[3] A. Raihan, "A review on the integrative approach for economic valuation of forest ecosystem services," *J. Environ. Sci. Econ.*, vol. 2, no. 3, pp. 1–18, Jul. 2023.
[4] E. G. Brockerhoff, L. Barbaro, B. Castagneyrol, D. I. Forrester, B. Gardiner, J. R. González-Olabarria, P. O. Lyver, N. Meurisse, A. Oxbrough, H. Taki, I. D. Thompson, F. van der Plas, and H. Jactel, "Forest biodiversity, ecosystem functioning and the provision of ecosystem services," *Biodiversity Conservation*, vol. 26, no. 13, pp. 3005–3035, Dec. 2017.
[5] S. Chen, J. Chen, C. Jiang, R. T. Yao, J. Xue, Y. Bai, H. Wang, C. Jiang, S. Wang, Y. Zhong, E. Liu, L. Guo, S. Lv, and S. Wang, "Trends in research on forest ecosystem services in the most recent 20 years: A bibliometric analysis," *Forests*, vol. 13, no. 7, p. 1087, Jul. 2022.
[6] P. Tompalski, N. C. Coops, J. C. White, T. R. H. Goodbody, C. R. Hennigar, M. A. Wulder, J. Socha, and M. E. Woods, "Estimating changes in forest attributes and enhancing growth projections: A review of existing approaches and future directions using airborne 3D point cloud data," *Current Forestry Rep.*, vol. 7, no. 1, pp. 1–24, Mar. 2021.
[7] Y. Zhang, S. Liu, Y. Wang, H. Gao, Y. Jiang, and D. Wei, "Forest management practices and policies exert strong impacts on the spatio-temporal variations of forest disturbance in Hunan province, China over the last three decades," *Forest Ecol. Manage.*, vol. 544, Sep. 2023, Art. no. 121167.
[8] A. S. Weed, M. P. Ayres, and J. A. Hicke, "Consequences of climate change for biotic disturbances in North American forests," *Ecol. Monographs*, vol. 83, no. 4, pp. 441–470, Nov. 2013.
[9] F. C. Gonçalves, E. P. Miguel, E. A. T. Matricardi, F. Emmert, and C. C. Santana, "Artificial intelligence associated with Sentinel-2 data in predicting commercial volume in Brazilian Amazon forest," *J. Appl. Remote Sens.*, vol. 15, no. 4, Nov. 2021, Art. no. 044511.
[10] A. Buchelt, A. Adrowitzer, P. Kieseberg, C. Gollob, A. Nothdurft, S. Eresheim, S. Tschiatschek, K. Stampfer, and A. Holzinger, "Exploring artificial intelligence for applications of drones in forest ecology and management," *Forest Ecol. Manage.*, vol. 551, Jan. 2024, Art. no. 121530.
[11] F. E. Fassnacht, J. C. White, M. A. Wulder, and E. Næsset, "Remote sensing in forestry: Current challenges, considerations and directions," *Forestry, Int. J. Forest Res.*, vol. 97, no. 1, pp. 11–37, Jan. 2024.
[12] S. Kanga, "Advancements in remote sensing tools for forestry analysis," *Sustain. Forestry*, vol. 6, no. 1, pp. 1–15, Oct. 2023.
[13] A. T. Stahl, R. Andrus, J. A. Hicke, A. T. Hudak, B. C. Bright, and A. J. H. Meddens, "Automated attribution of forest disturbance types from remote sensing data: A synthesis," *Remote Sens. Environ.*, vol. 285, Feb. 2023, Art. no. 113416.
[14] A. Kanáliková and E. Bubeníková, "Parking system with image processing," in *Proc. IEEE 17th World Symp. Appl. Mach. Intell. Informat. (SAMI)*, Jan. 2019, pp. 281–286.
[15] V. Wiley and T. W. Lucas, "Computer vision and image processing: A paper review," *Int. J. Artif. Intell. Res.*, vol. 2, no. 1, pp. 29–36, Feb. 2018.
[16] X. Zhu, K. Guo, H. Fang, L. Chen, S. Ren, and B. Hu, "Cross view capture for stereo image super-resolution," *IEEE Trans. Multimedia*, vol. 24, pp. 3074–3086, 2022.
[17] X. Zhu, K. Guo, S. Ren, B. Hu, M. Hu, and H. Fang, "Lightweight image super-resolution with expectation-maximization attention mechanism," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1273–1284, Mar. 2022.
[18] X. Zhu, K. Guo, T. Qiu, H. Fang, Z. Wu, X. Tan, and C. Liu, "Stereoscopic image super-resolution with interactive memory learning," *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120143.
[19] S. Yasmin, A. K. Murth, K. Shilpashree, and M. Rashid, "Geographic information system (GIS) and healthcare: An overview," *Int. J. Res. Rev.*, vol. 10, no. 2, pp. 286–294, Feb. 2023.
[20] I. Soubry, T. Doan, T. Chu, and X. Guo, "A systematic review on the integration of remote sensing and GIS to forest and grassland ecosystem health attributes, indicators, and measures," *Remote Sens.*, vol. 13, no. 16, p. 3262, Aug. 2021.
[21] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.
[22] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020.
[23] S. Albelwi and A. Mahmood, "A framework for designing the architectures of deep convolutional neural networks," *Entropy*, vol. 19, no. 6, p. 242, May 2017.
[24] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018.
[25] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, Mar. 2023.
[26] D. Schürholz, G. Castellanos-Galindo, E. Casella, J. Mejía-Rentería, and A. Chennu, "Seeing the forest for the trees: Mapping cover and counting trees from aerial images of a mangrove forest using artificial intelligence," *Remote Sens.*, vol. 15, no. 13, p. 3334, Jun. 2023.
[27] J. Li and J. Liu, "PointDMS: An improved deep learning neural network via multi-feature aggregation for large-scale point cloud segmentation in smart applications of urban forestry management," *Forests*, vol. 14, no. 11, p. 2169, Oct. 2023.
[28] S. V. Lim, M. A. Zulkifley, A. Saleh, A. H. Saputro, and S. R. Abdani, "Attention-based semantic segmentation networks for forest applications," *Forests*, vol. 14, no. 12, p. 2437, Dec. 2023.
[29] J. Yao, B. Song, X. Chen, M. Zhang, X. Dong, H. Liu, F. Liu, L. Zhang, Y. Lu, C. Xu, and R. Kang, "Pine-YOLO: A method for detecting pine wilt disease in unmanned aerial vehicle remote sensing images," *Forests*, vol. 15, no. 5, p. 737, Apr. 2024.
[30] C. Bai, X. Bai, and K. Wu, "A review: Remote sensing image object detection algorithm based on deep learning," *Electronics*, vol. 12, no. 24, p. 4902, Dec. 2023.
[31] B. Haq, M. A. Jamshed, K. Ali, B. Kasi, S. Arshad, M. K. Kasi, I. Ali, A. Shabbir, Q. H. Abbasi, and M. Ur-Rehman, "Tech-driven forest conservation: Combating deforestation with Internet of Things, artificial intelligence, and remote sensing," *IEEE Internet Things J.*, vol. 11, no. 14, pp. 24551–24568, Jul. 2024.
[32] A. Mitra, C. I. Alvarez, A. O. Abbasi, N. L. Harris, G. Shao, B. C. Pijanowski, M. R. Jahanshahi, J. G. P. Gamarra, H.-S. Kim, T.-K. Kim, D. Ryu, and J. Liang, "Mapping planted forests in the Korean Peninsula using artificial intelligence," *Forests*, vol. 15, no. 7, p. 1216, Jul. 2024.
[33] É. L. Bolfe, T. C. Parreiras, L. A. P. D. Silva, E. E. Sano, G. M. Bettiol, D. D. C. Victoria, I. D. Sanches, and L. E. Vicente, "Mapping agricultural intensification in the Brazilian savanna: A machine learning approach using harmonized data from Landsat Sentinel-2," *ISPRS Int. J. Geo-Inf.*, vol. 12, no. 7, p. 263, Jul. 2023.
[34] J. Xiang, Z. Zang, X. Tang, M. Zhang, P. Cao, S. Tang, and X. Wang, "Rapid forest change detection using unmanned aerial vehicles and artificial intelligence," *Forests*, vol. 15, no. 9, p. 1676, Sep. 2024.

[35] H. Pei, T. Owari, S. Tsuyuki, and Y. Zhong, "Application of a novel multiscale global graph convolutional neural network to improve the accuracy of forest type classification using aerial photographs," *Remote Sens.*, vol. 15, no. 4, p. 1001, Feb. 2023.
[36] K. Kalinaki, O. A. Malik, and D. T. C. Lai, "FCD-AttResU-Net: An improved forest change detection in Sentinel-2 satellite images using attention residual U-Net," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 122, Aug. 2023, Art. no. 103453.
[37] T. Molnár and G. Király, "Forest disturbance monitoring using cloud-based Sentinel-2 satellite imagery and machine learning," *J. Imag.*, vol. 10, no. 1, p. 14, Jan. 2024.
[38] H. Ouchra, A. Belangour, and A. Erraissi, "Machine learning algorithms for satellite image classification using Google Earth engine and Landsat satellite data: Morocco case study," *IEEE Access*, vol. 11, pp. 71127–71142, 2023.
[39] M. Pérez-Carrasco, B. Karelovic, R. Molina, R. Saavedra, P. Cerulo, and G. Cabrera-Vives, "Precision silviculture: Use of UAVs and comparison of deep learning models for the identification and segmentation of tree crowns in pine crops," *Int. J. Digit. Earth*, vol. 15, no. 1, pp. 2223–2238, Dec. 2022.
[40] C. Sun, C. Huang, H. Zhang, B. Chen, F. An, L. Wang, and T. Yun, "Individual tree crown segmentation and crown width extraction from a heightmap derived from aerial laser scanning data using a deep learning framework," *Frontiers Plant Sci.*, vol. 13, Jun. 2022, Art. no. 914974.
[41] S. Xu, R. Wang, W. Shi, and X. Wang, "Classification of tree species in transmission line corridors based on YOLO v7," *Forests*, vol. 15, no. 1, p. 61, Dec. 2023.
[42] S. Miao, K. F. Zhang, H. Zeng, and J. Liu, "Improving artificial-intelligence-based individual tree species classification using pseudo tree crown derived from unmanned aerial vehicle imagery," *Remote Sens.*, vol. 16, no. 11, p. 1849, May 2024.
[43] M. Syifa, S.-J. Park, and C.-W. Lee, "Detection of the pine wilt disease tree candidates for drone remote sensing using artificial intelligence techniques," *Engineering*, vol. 6, no. 8, pp. 919–926, Aug. 2020.
[44] (2024). *Satelites Pro Maps*. Accessed: Mar. 9, 2024. [Online]. Available: https://satellites.pro/
[45] (2024). *Roboflow*. Accessed: Mar. 16, 2024. [Online]. Available: https://roboflow.com/
[46] P. Kovacovic. (2024). *Forest Datasets*. [Online]. Available: https://app.roboflow.com/zilinska-univerzita-v-ziline/forest-full/2
[47] R. Bujack, K. Rink, D. Zeckzer, and S. Jänicke, "Predict saturated thickness using tensorboard visualization," in *Workshop on Visualisation in Environmental Sciences (EnvirVis)*. Brno, Czech Republic: The Eurographics Association, Jun. 2018.

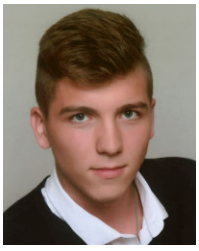
PATRIK KOVAČOVIČ received the B.S. degree in automation and the M.S. degree in process control from the University of Žilina, in 2022 and 2024, respectively, where he is currently pursuing the Ph.D. degree with the Department of Control and Information Systems.

RASTISLAV PIRNÍK was born in Humenné, Slovakia, in 1978. He received the M.S. and Ph.D. degrees in telecommunication engineering from the Faculty of Electrical Engineering, University of Žilina, in 2001 and 2010, respectively. He is currently an Associate Professor with the Department of Control and Information Systems, University of Žilina. He has been involved in numerous research projects, particularly those supported by the APVV and VEGA agencies. In addition, he has served as a Project Manager for two projects funded by the ASFEU agency and has contributed to the National Traffic Information System project. His research interests include data transmission in transport telematics systems and legislative matters.

JÚLIA KAFKOVÁ received the B.S. and M.S. degrees in biomedical engineering from the University of Žilina, Slovakia, in 2021 and 2023, respectively, where she is currently pursuing the Ph.D. degree with the Department of Control and Information Systems. Her research interests include machine learning, signal processing, and electronics, with the main focus on ECG.

MÁRIO MICHÁLIK received the B.S. degree in automation and the M.S. degree in process control from the University of Žilina, Slovakia, in 2022 and 2024, respectively, where he is currently pursuing the Ph.D. degree with the Department of Control and Information Systems. His research interests include machine learning, real-time visual detection, and computer vision.

ALŽBETA KANÁLIKOVÁ was born in Žilina, Slovakia, in 1969. She received the Ing. degree in communication technology from the PEDAS Faculty, University of Transport in Žilina, in 1991, and the Ph.D. degree in communication engineering from the Faculty of Management and Informatics, University of Žilina, in 2009. Currently, she is an Assistant Professor with the Department of Control and Information Systems, University of Žilina. She participated in many research projects, especially projects supported by the KEGA and VEGA agencies. Her research interests include computer science, information technology, cyber security, and data analysis.



PAVOL KUCHÁR received the B.S. degree in automation and the M.S. degree in process control from the University of Žilina, Slovakia, in 2019 and 2021, respectively, and the Ph.D. degree from the Department of Control and Information Systems, in 2024. His research interests include machine learning, telematics, and transportation systems.