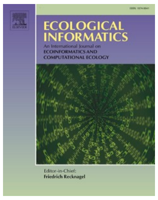
A forestry investigation: exploring factors behind improved tree species classification using bark images

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



A forestry investigation: Exploring factors behind improved tree species classification using bark images

Gokul Kottilapurath Surendran [a], Deekshitha [b,c,d], Martin Lukac [a,e], Martin Lukac [f], Jozef Vybostok [g], Martin Mokros [a,h,*]

[a] *Faculty of Forestry and Wood Sciences, Czech University of Life Sciences Prague, Prague CZ 165 000, Czechia*
[b] *Netherlands eScience Center, Science Park 402, Amsterdam, North Holland 1098 XH, the Netherlands*
[c] *Leiden Institute for Advanced Computer Science, University of Leiden, Leiden 9500 2300 RA, the Netherlands*
[d] *Information and Computing Sciences, Utrecht University, Utrecht 80125 3508 TC, the Netherlands*
[e] *School of Agriculture, Policy and Development, University of Reading, Reading RG6 6EU, UK*
[f] *Department of Computer Networks and Engineering, Hiroshima City University, Japan*
[g] *Faculty of Forestry, Technical University in Zvolen, Slovakia*
[h] *Department of Geography, University College London, Gower Street, London WC1E 6BT, UK*

ARTICLE INFO

Keywords:
Tree species classification
Machine learning
Close-range remote sensing
Grid search
CNN parameter tuning

ABSTRACT

Novel ground-based remote sensing approaches have demonstrated high potential for accurate and detailed mapping and monitoring of forest ecosystems. These methods enable the measurement of various tree parameters important for forest inventory or ecological research, such as diameter at breast height, tree height and volume, and crown parameters. One crucial piece of information is tree species, which is essential for various reasons and challenging to implement within ground-based technology workflows. This study investigates why researchers often focus on segment-specific bark images for tree species classification via deep neural networks rather than large or entire tree images. Additionally, the aim is to determine the most effective algorithmic approaches for efficient tree species classification from bark images and to make these methods more accessible to interdisciplinary researchers. The findings reveal that segment-specific datasets with more overlaps provide better accuracy across various algorithms. Additionally, pre-processing techniques such as scaling can enhance accuracy to a certain extent. Convolutional Neural Networks (CNNs) consistently deliver the highest accuracy, even with diverse datasets, but fine-tuning these algorithms poses significant challenges for interdisciplinary researchers. To address this, we developed Windows-based research software, CNN Parameter Tuner 1.0, which allows the import of various data formats (jpg and png) and efficiently conducts parameter tuning by selecting parameters and values from the menu options.

1. Introduction

Identifying tree species within diverse natural forests is essential for understanding forest functionality and predicting forest responses to natural phenomena and human-induced influences (Aszalós et al., 2022; Uriarte et al., 2009). Recent technological advancements, such as the use of Terrestrial Laser Scanners (TLSs) and photogrammetry devices, have considerably broadened the scope of possibilities for species identification and detection (Liang et al., 2018). Despite their enhanced efficiency and accuracy over traditional methods, these advanced technologies produce extensive datasets necessitating automated or semiautomated processing capabilities. Deep learning algorithms such as Convolutional Neural Networks (CNNs) have proven effective in analysing data from laser scanners and photogrammetry equipment (Wojtkowska et al., 2021). Moreover, machine learning techniques, including shallow and deep learning methods, have been applied in forest management. These methods Include analysing and surveilling forested areas, detecting forest fires (Nikolić et al., 2023; Salavati et al., 2022), and identifying endangered habitats (*Detecting forest threats with Artificial Intelligence – AZO – Space of Innovation*, 2023). Here, cameras and Unmanned Aerial Vehicles (UAVs), such as drones, play pivotal roles in accelerating and enhancing monitoring processes. Even compact computing devices can

* Corresponding author at: Department of Geography, University College London, Gower Street, London WC1E 6BT, UK.
E-mail addresses: ksg@fld.czu.cz (G.K. Surendran), d.kumbla@esciencecenter.nl (Deekshitha), lukacm@fld.czu.cz (M. Lukac), malu@hiroshima-cu.ac.jp (M. Lukac), jozef.vybostok@tuzvo.sk (J. Vybostok), m.mokros@ucl.ac.uk (M. Mokros).

https://doi.org/10.1016/j.ecoinf.2024.102932
Received 8 April 2024; Received in revised form 29 November 2024; Accepted 30 November 2024
Available online 5 December 2024

now execute fundamental image-processing tasks, streamlining these operations and reducing reliance on human resources.

In the study of da Silva et al., 2023, the authors utilised UAV imagery with classical machine learning algorithms to detect and model the presence of the invasive tree species *Hovenia dulcis* in a subtropical forest in Brazil. They employed two primary approaches for image analysis: Pixel-Based (PB) and Object-Based Image Analysis (OBIA), combined with the algorithms Random Forest (RF) and Support Vector Machine (SVM). The results demonstrated that the RF algorithm, mainly when applied in the PB approach, outperformed other combinations, achieving an overall accuracy of 91.5 % during training and 90.91 % in the validation phase, with a kappa index of 0.87. These outcomes suggest that integrating UAV-RGB data with machine learning techniques is highly effective in accurately identifying invasive species.

The UAV images were also utilised for diseased tree detection. Hu et al., 2022, focused on detecting and classifying diseased pine trees at various severity levels via UAV remote sensing images. To achieve this, they developed a method that integrates a modified YOLOv5 model, referred to as DDYOLOv5, with a ResNet50 network. The DDYOLOv5 model was enhanced by incorporating efficient channel attention (ECA) and hybrid dilated convolution (HDC) modules to improve the detection accuracy. Compared with traditional deep learning models such as Faster R-CNN and RetinaNet, the proposed method achieves a precision increase of 13.55 %, a recall improvement of 5.06 % and a 9.71 % increase in the F1-score compared with the original YOLOv5 model. Moreover, Veras et al., 2022 used multi-season UAV images to map tree species in the Amazonian forest. The authors used CNNs (ResNet-18 model and DeepLabv3+ architecture), and their goal was to explore whether the CNN could learn species-specific phenological characteristics and whether fusing multi-season images would improve classification accuracy. Their study reported an improvement in classification accuracy of up to 21.1 % when multi-season images were used, with the accuracy reaching 90.5 %. All these results show that the RGB images were adequate for species classification and detection tasks, which were focused primarily on machine learning algorithms.

The integration of photogrammetry has increased the accessibility and efficiency of species identification, mainly through leaf-based classification of tree species (Kanda et al., 2021; Minowa et al., 2022). Munisami (Munisami et al., 2015) and Zhou (Zhou et al., 2016) conducted experiments utilising classification methodologies such as k-Nearest Neighbors (k−NN) and Artificial Neural Networks (ANNs), which yielded favourable outcomes. Pushpa et al., 2024 focused on categorising the medical plant species. To achieve this, they developed a hierarchical classification framework. Their framework integrated convolutional features with geometric, texture, shape, and multispectral features for classification tasks. Moreover, they proposed a two-level hierarchical plant classification model to address the challenges of inter-class similarity and intra-class variations with an RF model. Studies have also focused on species classification on the basis of leaf characteristics. Barré et al., 2017 aimed to develop a deep learning system to learn discriminative features from leaf images and a classifier for species identification of plants. To achieve this, they developed LeafNet, a CNN-based plant identification system, and it provided better results when applied to the LeafSnap, Flavia and Foliage datasets.

However, the colour, structure, and patterns of leaves undergo seasonal variations. They can be influenced by environmental and biological factors (Chaki et al., 2019), thus limiting the applicability of seasonal organ-based identification for forest management objectives. In contrast, tree bark is relatively stable throughout the season as a permanent feature. Boudra (Boudra et al., 2022) and Remeš & Haindl (Remeš and Haindl, 2019) employed diverse machine learning algorithms for tree species classification on the basis of bark attributes. However, bark structures are often small, display species-specific significance, and may be masked by external elements such as lichens or mosses (Fekri-Ershad, 2020), leading to a heightened risk of misclassification. In response, researchers have explored the integration of bark and leaf characteristics for classification, resulting in superior outcomes compared with single-organ approaches (Fiel and Sablatnig, 2011; Zhao et al., 2020). Jendoubi et al. (Jendoubi et al., 2020) proposed a two-step methodology involving an RF classifier for identifying new leaf characteristics on the basis of pre-trained data and then selecting bark features and clustering employing a k-NN classifier.

Researchers have reported that combining bark images and leaves can improve classification task accuracy. Additionally, few of them focused on a single permanent feature, 'bark', for classification (Boudra et al., 2021; Bressane et al., 2015; Carpentier et al., 2018; Fekri-Ershad, 2020; Kim et al., 2022; Remeš and Haindl, 2019; Robert et al., 2020), and they achieved better accuracy on this classification task. Considering bark as the main feature for species identification, researchers have also developed an automatic image recognition model for urban tree species (Sun and Shi, 2023). This approach included 21 tree species employed within a combined Channel Attention Module (CAM) framework with algorithms such as Spatial Pyramid Pooling (SPP) and Mixed Depthwise Dilated Convolutional Kernels. Moreover, in this proposed framework, the authors used a Mixed Convolutional Kernel (MK) and a CAMP-MKNet Convolutional Neural Network as core algorithms for bark classification. Here, the core model achieved an accuracy of 84.25 %.

Moreover, the researchers have focused on multiple cameras and ultrasonic sensors (Chen et al., 2018). In this approach, the devices are integrated into a single organic mechanical structure that can rotate to detect the surrounding environment, which helps reduce the non-detection zone. In this framework, the authors used a multi-feature fusion technique. They used Histogram Oriented Gradient (HOG) and SVM algorithms in the initial training phase. After this, a cross-edge detector extracts the trunk's gradient histogram features. Additionally, ultrasonic sensors are used to obtain the location data of the trunks, and a moving average filter is used to reduce the error of mobile robot localisation. The employed trunk recognition framework provided a recall of 92.14 % and an accuracy of 95.49 %. The automatic robotics methods also face challenges because of harsh environmental conditions such as terrain irregularities and steep slopes. A new framework was employed to address this issue and extract reliable features (Aguiar et al., 2020). Their approach used a single camera and an Edge Tensor Processing Unit (TPU) with object detection via various deep learning models. They also utilised transfer learning on several pre-trained MobileNet V1 and MobileNet V2 versions.

Tree trunk detection was not only performed for species classification. Researchers have focused on detection to conduct biomass-related research. D. Q. da Silva et al., 2021 detected the ground level of forest tree trunks in visible and thermal images via deep learning based methods. The authors used SSD MobileNet V2, SSD Inception-v2, SSD ResNet50, SSDLite MobileNet and YOLOv4 Tiny. The YOLOv4 Tiny was the best model for this trunk detection task. It provided an accuracy of 90 %. The authors also used various algorithms on different datasets and achieved an accuracy of 90 % in trunk detection.

Open-access databases for tree species classification are limited, yet datasets serve as pivotal assets in research. For example, the Austrian Federal Forest (AFF) dataset represents a private dataset comprising 1082 images spanning 11 tree species (Fiel and Sablatnig, 2011). The publicly available Trunk12 (TRUNK12, 2022) dataset comprises only 360 images encompassing 12 species. However, the sample collection methodology employed in collecting the Trunk12 dataset was unclear, diminishing its suitability for research purposes. Another dataset, Bark101 (Boudra et al., 2022), derived from the PlantCLEF 2017 initiative, presents challenges due to an imbalanced distribution of images across classes. The scarcity of accessible datasets restricts the breadth of research pursuits and contributions within this domain. Most of the research has focused on these datasets, and it is also important to note that the dataset size is smaller for most traditional machine learning and neural network algorithms.

In response to this constraint, Carpentier et al., 2018 introduced a novel open dataset named BarkNet 1.0 to enrich research contributions and enhance dataset quality. The BarkNet dataset encompasses 23,000 images spanning 23 distinct tree species, rendering it conducive for utilisation in deep learning algorithms. However, the BarkNet dataset represents tree species native to Quebec city, Canada. Upon evaluation utilising neural network algorithms such as VGG-16 and EfficientNet, as conducted by Kim (Kim et al., 2022) on the BarkNet dataset, species identification proved relatively straightforward, even when relying solely on single-organ features. They encountered challenges stemming from intra-class similarity, leading to occasional misclassification of the same species into different genera.

When tree species classification is focused on bark images, dataset quality is essential for research. During data collection, factors such as light glare, the presence of mosses, and foreign elements such as different tree branches (Carpentier et al., 2018) can alter pixel intensity and image appearance, potentially misleading machine learning algorithms, particularly for deep learning models. Classical machine learning methods may utilise diverse feature extraction techniques, yet anomalies and misleading elements can still influence outcomes. Although highly advanced, deep learning requires substantial computational resources and expertise to execute complex tasks efficiently. Despite the potential effectiveness of deep learning with small datasets, there is a greater risk of overfitting, and deploying these methods in real-time scenarios may yield less precise outcomes.

In the literature, research on tree species classification using bark datasets is limited. Moreover, even the RGB images from UAVs could provide better results on classification tasks. To the best of our knowledge, no one has investigated why there is a need to shift to deep neural networks instead of classical machine learning. Can these neural network algorithms address complex datasets and ensure better classification accuracy? To address these research questions, we need to investigate the bark classification (tree species) approach differently, such as what contributes better results, such as the combination of datasets or which part from the dataset contributes better results, or whether algorithms with different parameters can increase the classification accuracy. It is necessary to examine whether the best algorithm can deal with all kinds of datasets, even if they are not high-quality or complex.

This study also aims to assist forestry researchers in efficiently utilising classical machine learning and neural networks for bark image classification. Therefore, the contributions of this study are as follows:

- Exploring effective strategies: This study investigates effective strategies for utilising data to improve tree species classification accuracy while addressing common machine learning challenges such as overfitting and underfitting.
- Comparative analysis: This study compares classical machine learning techniques with neural networks to evaluate their effectiveness in tree species classification.
- Guidance for forestry researchers: The findings provide valuable insights and practical guidance for optimising the use of machine learning algorithms in their research.
- Development of research software: New research software has been developed for tuning CNN parameters, further enhancing the utility and effectiveness of machine learning methodologies in forestry research.

By addressing challenges systematically and comparing various techniques, this research significantly advances the understanding of applying machine learning in forestry.

This work employed classical machine learning and neural network algorithms on various datasets and dataset formats to address the above-mentioned questions. Classical machine learning algorithms with pre-defined parameters were employed in the initial phase on various datasets. Moreover, in this phase, we investigated the influence of feature scaling, parameter tuning (grid search), etc. Then, we examined neural network algorithms on all the datasets. The findings reveal that segment-specific datasets with more overlaps provide better accuracy across various algorithms. Moreover, the CNN consistently delivered the highest accuracy, even with diverse datasets. However, fine-tuning these algorithms poses significant challenges for interdisciplinary researchers. To address this, we developed a user-friendly Windows-based research software, the CNN parameter Tuner 1.0.

The subsequent sections of this paper are organised as follows. Section 2 (Methodology) focuses on data collection and pre-processing, followed by the proposed architecture. Section 3 presents the results of all the methods described in Section 2. Section 4 discusses the challenges and limitations and finalises the results in the conclusion section.

2. Methodology

The extensive use of machine learning (shallow and deep learning) algorithms is evident in forestry, although they often employ predefined architectures and scenarios from various fields. Our study is dedicated to identifying optimal scenarios and architectures for classifying tree species on the basis of bark images. Using grid search methodology, we determine the best parameter values for each algorithm, encompassing both classical machine learning methods such as k-NN, Gaussian Naïve Bayes (GNB), RF, Decision Tree (DT), Gradient Boosting (GB), and Support Vector Machine (SVM), as well as neural networks models such as Multilayer perceptron (MLP) and CNNs. Fig. 1 provides an overview of the proposed research architecture.

The methodology section is divided into various subsections according to the research architecture diagram. The first subsection addresses the data acquisition (2.1) and describes the datasets considered for the study. The following subsections provide more information about data pre-processing, such as segmentation (2.2) and nonlinear transformation (2.3). After this, we cover the post-processing phases, such as the feature extractor considered for the study (Gray-Level Co-Occurrence Matrix), and the following subsection briefly describes the considered algorithms and the grid search approach.

2.1. Data acquisition

For this study, we generated two primary datasets, Slovak and Czech University of Life Sciences (CZU), which include images of seven tree species. These images were captured via Sony Alpha 7 and Canon EOS 4000D digital cameras. The dataset was captured from nearly half a meter distance from the trees. Each tree, in turn, is represented by a collection of 15–22 images that comprehensively capture the tree bark from different ground angles, ensuring a minimum 60 % overlap between consecutive images. The first dataset, the Slovak dataset, consists of 1369 cropped and 527 exact cropped images of European beech (*Fagus sylvatica* L.), Sessile oak (*Quercus petraea* (Matt.) Liebl.), Norway spruce (*Picea abies* (L.) H. Karst.), and European silver fir (*Abies alba* Mill.). These were taken in Včelien, Banskobystrický, Slovakia, in June 2022. Owing to anomalies on the tree stems, the original images were unsuitable for use. Thus, these images are segmented into two categories: normal cropped and exact cropped. Subsection 2.2 describes how we segmented these tree images into two categories.

The second dataset, which was collected from the CZU campus in August 2022, includes 386 images of European beech (*Fagus sylvatica* L.), Large-leaved linden (*Tilia platyphyllos*), Norway Maple (*Acer platanoides*), and Scots pine (Pinus sylvaltica L.). The dataset includes original and segmented images. Moreover, each dataset comprises multiple angles of tree stems, with at least ten photographs taken per tree to capture the bark structure and patterns. We divided each dataset into training and testing sets, with 75 % of the images used for training and the remaining for testing the machine learning model.

Additionally, we generated a third Nonlinear dataset to determine the efficiency of the machine learning algorithm in performing complex nonlinear patterns. For this purpose, the Slovak exact cropped dataset

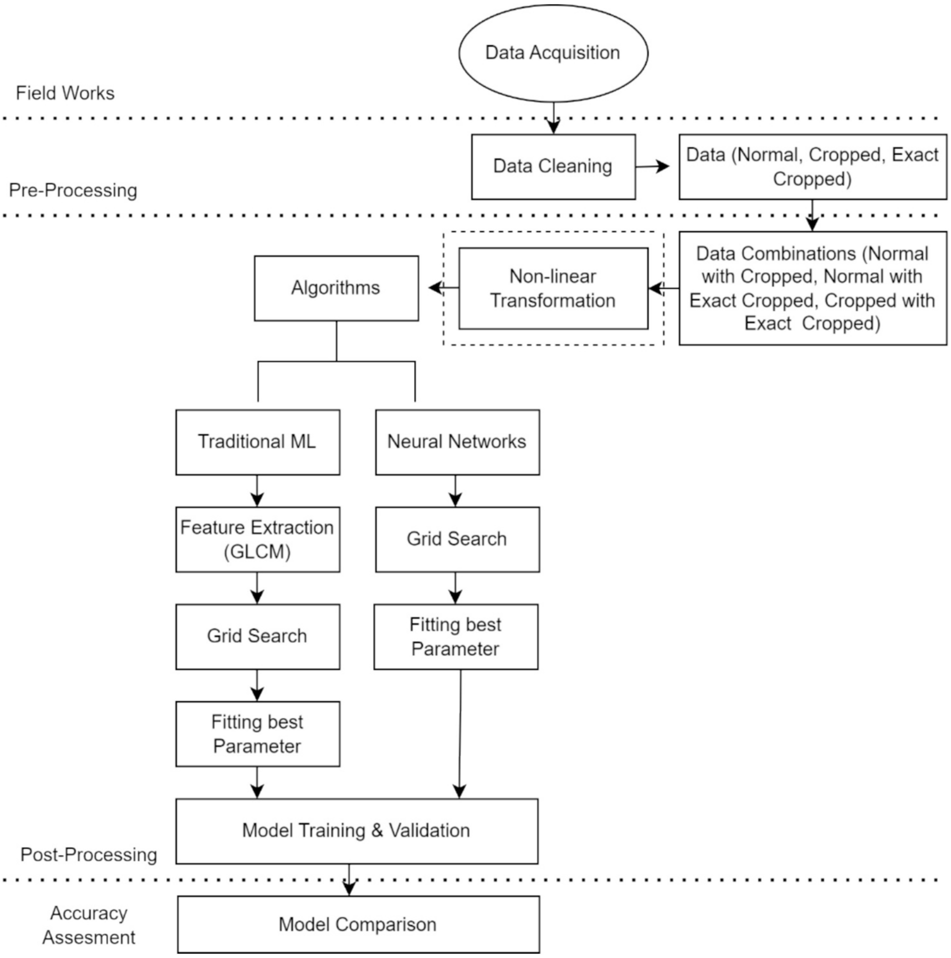

Data Acquisition
Field Works
Data Cleaning
Data (Normal, Cropped, Exact Cropped)
Pre-Processing
Algorithms
Non-linear Transformation
Data Combinations (Normal with Cropped, Normal with Exact Cropped, Cropped with Exact Cropped)
Traditional ML
Neural Networks
Feature Extraction (GLCM)
Grid Search
Grid Search
Fitting best Parameter
Fitting best Parameter
Model Training & Validation
Post-Processing
Accuracy Assesment
Model Comparison

Fig. 1. Research architecture workflow diagram.

has been regenerated into a new nonlinear dataset. This involves applying nonlinear deformation/transformation and swirl (as outlined in Subsection 2.3) to generate the dataset. The primary objective of this procedure is to assess the effectiveness and capacity of classical machine learning algorithms and neural networks for bark image classification.

2.2. Segmentation

The first step in the segmentation process includes carefully examining the datasets to eliminate anomalies, such as numbers, mosses, tree leaves, or branches present on the bark surface in the images. The identified anomalies from the image datasets are shown in Fig. 2.

The next step is to crop the edges of the bark areas from the original data. Owing to the inherent challenges associated with automatic segmentation algorithms, particularly in accurately delineating the bark edge and accommodating various markings and anomalies, these methods were not employed in this study. Additionally, the image acquisition process significantly influences the segmentation task; images captured at a distance from the stem or with a high aperture value (e.g., those captured with entry-level cameras) pose considerable difficulties due to occlusion, as depicted in Fig. 3.

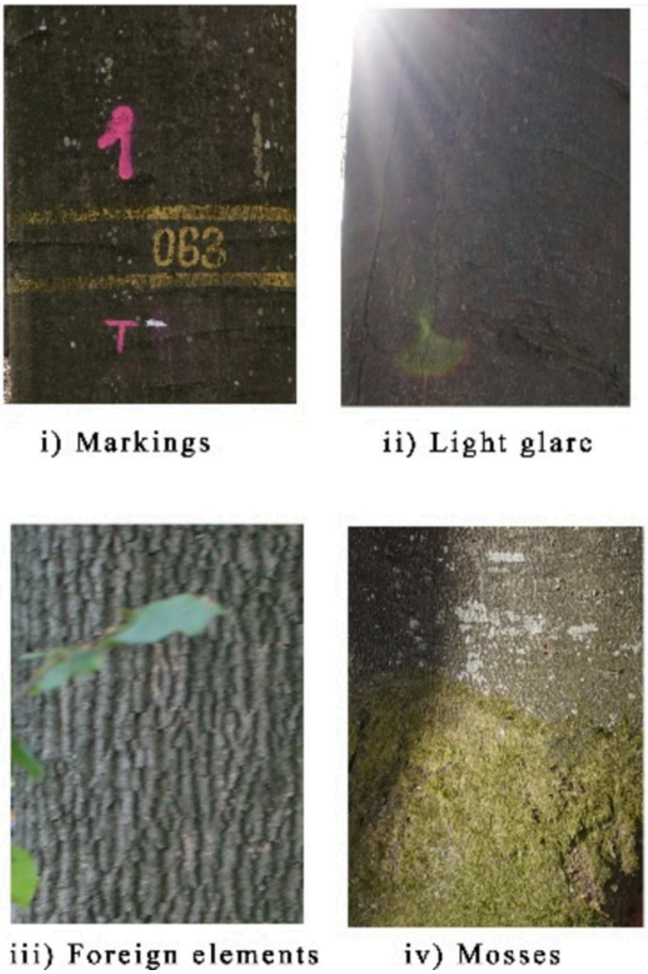
Thus, manual segmentation was executed via the open-source software Krita (Krita | Digital Painting, 2024). The extraneous elements (anomalies) were removed from the bark images during cropping. Moreover, images of more than 60 % extraneous elements in the bark area were excluded from the analysis.

The CZU and Slovak dataset samples are shown in Fig. 4. The first column shows the original image, and the second column represents the first set of generated datasets after removing the anomalies. In this approach, we removed the anomaly region from the images, kept the rest of the image in the original form, and named these datasets normal cropped. The third column shows the exact cropped dataset. Here, we only took a small portion of the dataset, which is a tiny significant portion (segment-specific) of the images as per the quality level, and it is named an exact cropped dataset.

The details of the generated dataset are shown in Fig. 5. The first set of bars (orange) represents Nonlinear data (321 images), and the next bar represents combination data (772 images). The red bars represent the two datasets: the CZU exact cropped dataset and the CZU normal cropped dataset (386 images per dataset). Green represents the Slovak exact cropped dataset (527 images), and the last shows the Slovak normal cropped dataset (1367 images). From the datasets, 75 % of the data are allocated for the training process, with the remaining portion utilised for testing purposes.

2.3. Nonlinear transformation/deformation

The application of nonlinear transformation alters the linear correlation between variables, leading to a deviation from direct proportionality between the input and output, consequently affecting the correlation between them. In this study, we used swirl nonlinear image deformation and image warping (Fig. 6), as documented in (*Scikit-image: Image Processing in Python — Scikit-image*, 2024), to manipulate the input images. Swirl induces a whirlpool effect by initially computing the relative center to (x_0,y_0) and transforming the images into polar coordinates (Eq. (1) and Eq. (2)).



i) Markings

ii) Light glare

iii) Foreign elements

iv) Mosses

Fig. 2. Anomalies in the dataset: The dataset consists of anomalies such as markings performed by other studies, light glares (sunlight), foreign elements such as other tree leaves and branches, and finally, the presence of mosses.

$$\Theta = arctan\left(\left(y - y_0\right) / \left(x - x_0\right)\right) \tag{1}$$

$$\rho = \sqrt{\left(x - x_0\right)^2 + \left(y - y_0\right)^2} \tag{2}$$

The swirl whirlpool effect is calculated as:

$$\Theta' = \Phi + s\, e^{-\rho/r} + \Theta \tag{3}$$

Here, the adjusted angle Θ′ is calculated by summing the rotation angle Φ with the scaled strength s adjusted by the exponential of the ratio ρ to radius r and adding to the original angle Θ.

2.4. Gray level co-occurrence matrix

Gray-Level Co-occurrence Matrix (GLCM), coined by Haralick, Shanmugam, and Dinstein in 1973 (Haralick, 1979; Haralick et al., 1973), is a widely utilised method for extracting features. The spatial relationships between brightness values in grayscale images are analysed via pixel arrangement and texture characteristics. It calculates second-order texture properties by examining the input image's pixel pair relationships and separations. In this study, five specific textural features, such as contrast, dissimilarity, homogeneity, correlation and energy, were examined, and we used three GLCM windows ([1] [0], [3] [0], & [5] [0]) with zero radians.

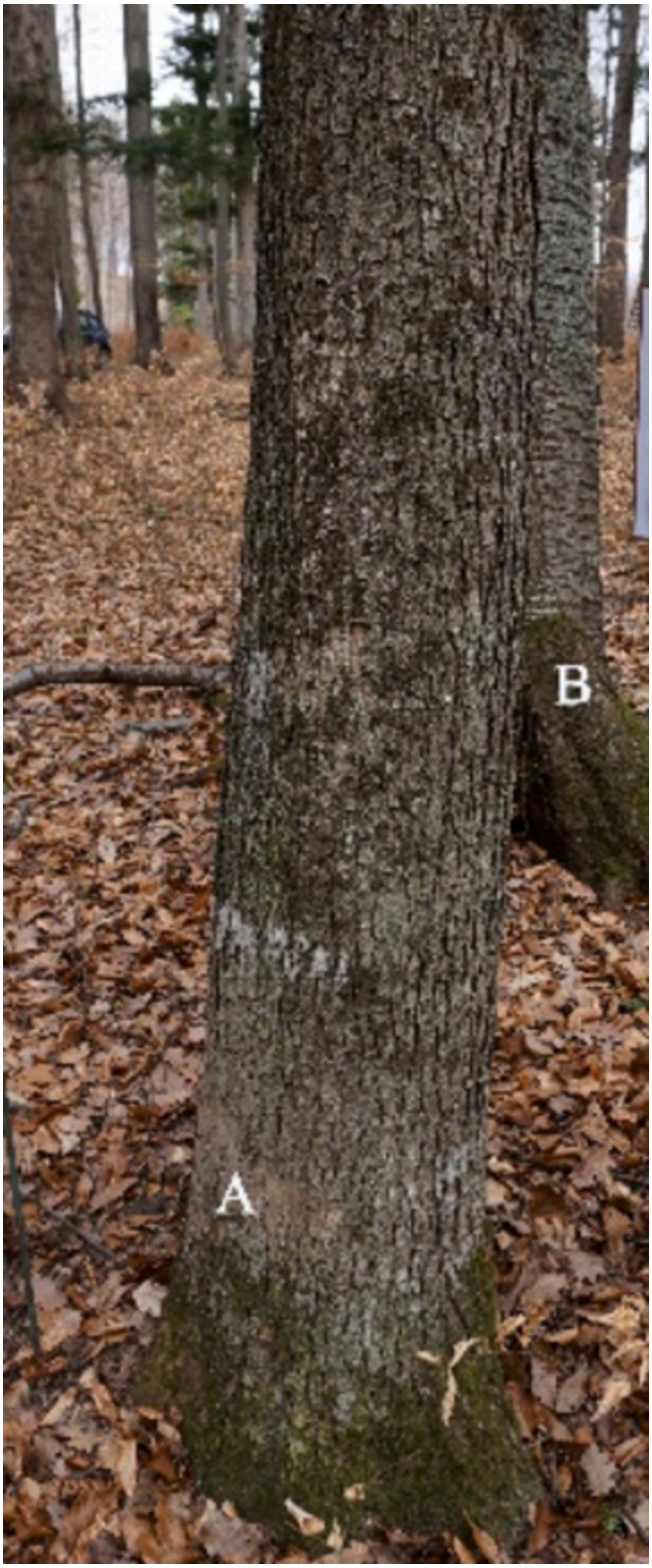


Fig. 3. Trees with occlusion: This is an example of a tree with an occlusion problem. The figure consists of two trees marked as A and B. Finding the tree's boundary is challenging because of nearby trees.

- Contrast: This feature quantifies the difference between neighboring pixel values, with weights increasing exponentially, as expressed in the following equation:

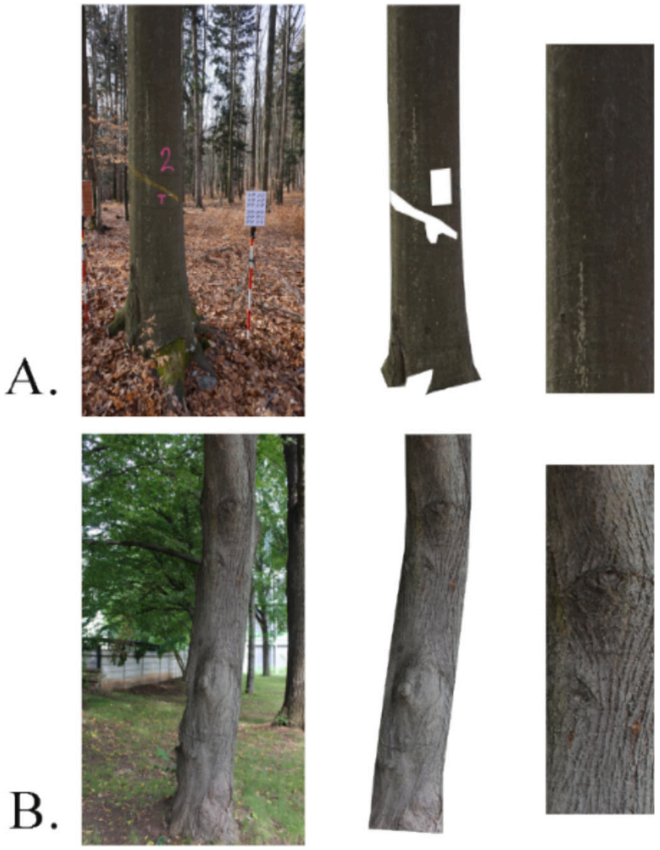


Fig. 4. Datasets: The first row represents the Slovak datasets (A), and the second row represents the CZU dataset (B).

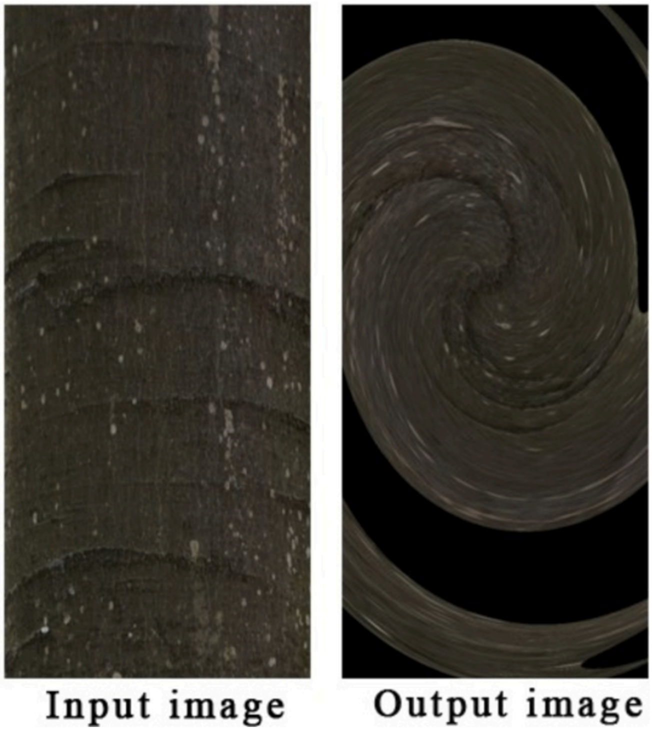


Fig. 6. Nonlinear data: The figure on the left side shows the original data, and the figure on the right side shows the generated nonlinear data with a swirl effect.

$$Contrast = \sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2 \tag{4}$$

Where $P_{i,j}$ represents the probability of occurrence of the gray-level values (*i,j*). If both *i* and *j* are equal, indicating identical pixel values, no contrast exists between these pixels. For example, *i*-*j* = 0 signifies that the pixels are similar. *N* is the number of possible values.

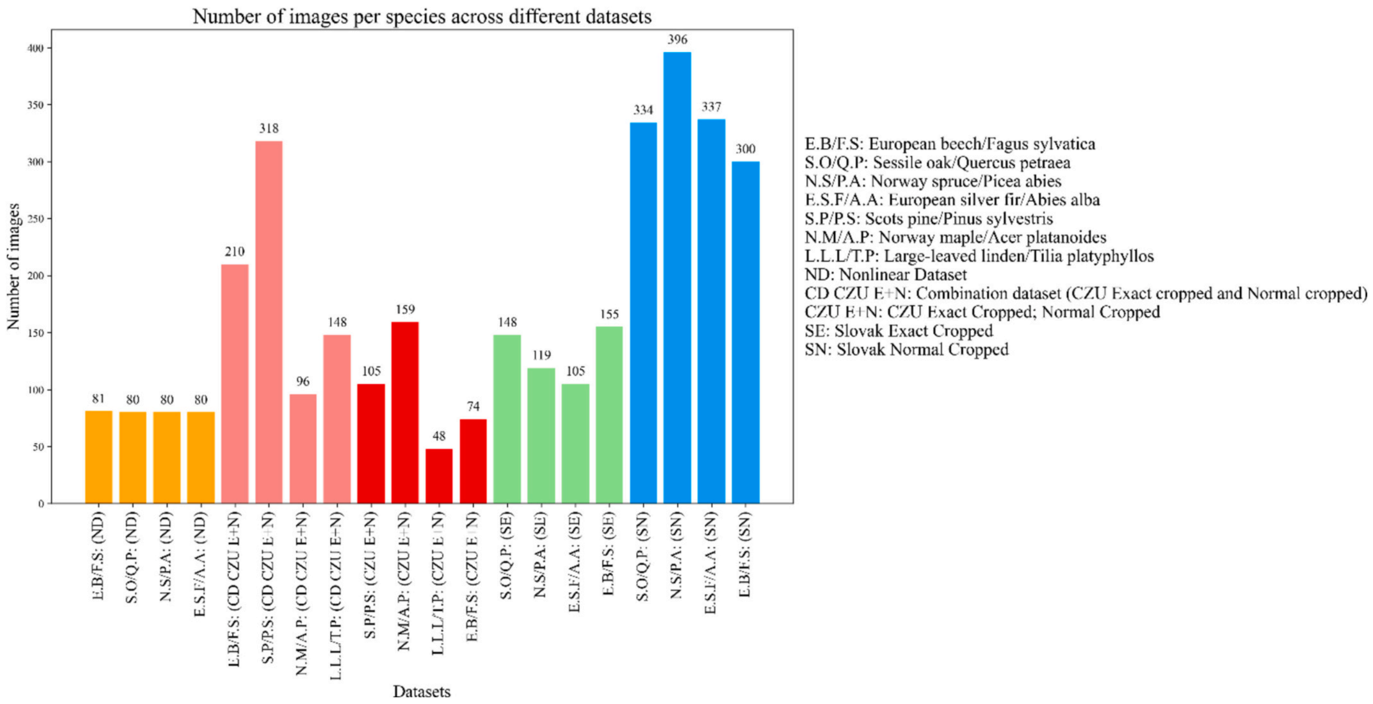


Fig. 5. Datasets: The figure shows various datasets utilised in this study with the corresponding number of images.

- Dissimilarity: This measures the linear distinction between pixel values, with dissimilarity weights increasing linearly.

$$Dissimilarity = \sum_{i,j=0}^{N-1} P_{i,j}|i-j| \tag{5}$$

- Homogeneity: The relationship between contrast and homogeneity values follows an inverse proportionality known as Inverse Difference Moment (IDM). The homogeneity measurement in images yields higher values for more minor differences in gray tones within pairs and peaks when all the elements in the image are identical.

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2} \tag{6}$$

- Correlation: This feature quantifies the linear association between neighboring pixel values. Here, μ_i and μ_j are the means and σ_i^2 and σ_j^2 are the variances corresponding to indices i and j.

$$Correlation = \sum_{i,j=0}^{N-1} P_{i,j}\left[\frac{(i-\mu_i)\left(j-\mu_j\right)}{\sqrt{\left(\sigma_i^2\right)\left(\sigma_j^2\right)}}\right] \tag{7}$$

- Energy: This represents the textural consistency of pixel pairs, often referred to as uniformity or angular second moment.

$$Energy = \sum_{i,j=0}^{N-1} P(i,j)^2 \tag{8}$$

2.5. Classical machine learning and neural network algorithms

We selected prominent classical machine learning and neural network algorithms in this experiment. We considered CNNs for classification because of their effectiveness and widespread use. The following machine learning models are widely utilised for classification problems: photogrammetry and remote sensing (Abdali et al., 2024; Bolyn et al., 2022; Kanda et al., 2021). In this research, I applied these machine learning algorithms for the task of tree species classification based on bark images.

- **Support Vector Machine (SVM):** SVM was used to classify bark species by identifying optimal hyperplanes in multi-dimensional space. These hyperplanes create boundaries between different species based on bark features, ensuring precise separation of classes (Y. Zhang, 2012).
- **k-Nearest Neighbors (k-NN):** This algorithm was employed to classify barks by calculating the distances between feature points in various dimensions. By setting a predefined number of neighbors (k), the algorithm grouped bark samples based on their similarity, enabling accurate species classification (Syriopoulos et al., 2023).
- **Gradient Boosting (GB):** Gradient Boosting was applied to refine the classification model by iteratively minimizing errors from previous iterations. This process allowed the model to improve the accuracy of bark species predictions with each step, combining gradient descent with boosting techniques (Bentéjac et al., 2021).
- **Gaussian Naïve Bayes (GNB):** GNB was used to classify bark samples under the assumption that bark features follow a Gaussian distribution. This approach proved useful for species where feature distribution closely approximates normality (H. Zhang, 2004).
- **Decision Tree (DT):** A decision tree structure was utilised to classify species based on bark characteristics. The model made sequential decisions at each node, leading to a classification based on the bark's features (Fürnkranz, 2010).
- **Random Forest (RF):** RF was leveraged for its robust performance in classifying tree species. The ensemble of decision trees generated from random subsets of bark features voted collectively, improving the classification accuracy (Louppe, 2014).
- **Multi-layer Perceptron (MLP):** This neural network architecture, consisting of multiple perceptron layers, was used to classify bark images. MLP's ability to capture complex patterns in data allowed for enhanced species classification (Rudolf et al., 2022).
- **Convolutional Neural Networks (CNNs):** CNNs were employed to classify tree species by autonomously learning and extracting features from bark images. Given CNN's superior performance in image classification tasks, it was particularly effective in identifying subtle visual patterns in bark textures (O'Shea and Nash, 2015). The CNN algorithm consists of various parameters (Appendix Table 1) that play a crucial role in providing better accuracies.

2.6. Grid search

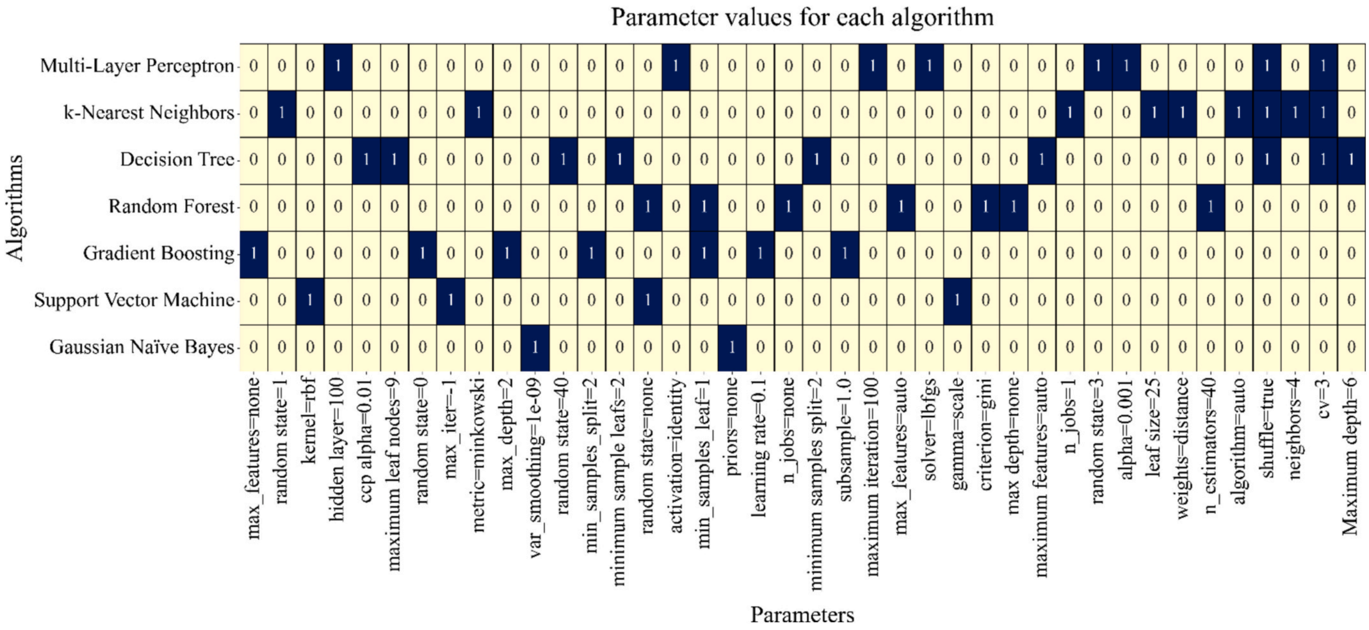
Machine learning algorithms are developed on the basis of predefined parameters, and adjusting these parameter values can significantly influence the learning process, thereby impacting the efficiency of the algorithms. These modifiable parameters are commonly referred to as hyperparameters. Grid search is a tuning technique used to determine the optimal values for these hyperparameters. For example, hyperparameters in an RF algorithm encompass parameters such as maximum depth (max_depth) and maximum leaf nodes (max_leaf_nodes). However, the specific hyperparameters and their corresponding values may vary across different algorithms. The current objective entails identifying the appropriate hyperparameters for each algorithm and defining their values before commencing the learning process. The grid search methodology is a valuable tool for parameter tuning, facilitating the discovery of optimal parameter values for each algorithm. In our experimental setup, we also employed this method to pinpoint the optimal values for each algorithm. Fig. 7 shows the parameter values used in our experiments. Appendix Table 2 presents an overview of the algorithms utilised in our study and their corresponding parameters.

In this study, the grid search is initially applied to unscaled data to assess whether improved parameter values yield better results. Additionally, the GNB is excluded from this study because it has only two hyperparameters: priors and regularisation. SVMs require significant time for grid searches, even on high-performance computing platforms. The longer processing time of SVMs leads to their termination during grid search. Moreover, other experiments utilised the scaling technique to determine the difference between scaled and unscaled accuracy deviations. We utilised StandardScaler (*StandardScaler — scikit-learn 1.5.0 Documentation*, 2024) from the sklearn library for this scaling process. We also utilised various cross-validations (Hastie, 2008) on all algorithms. Algorithms such as MLP, k-NN, DT, and RF were employed with 3-fold cross-validation only for GB; we used 15-fold cross-validations.

Our experiments used various accuracy metrics to calculate the training and testing accuracies, such as precision, recall, F1-score (Powers and Ailab., 2011), and accuracy, which were calculated via the 'score' method in the scikit-learn library. However, in the main results, we only added the score accuracy metric, but in the appendix, we added other scores for the listed experiments. The equation for calculating accuracy via the score method is given below:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{9}$$

In this study, we used an Azure Intel Xeon® CPU E5–2690 v4 with a clock speed of 2.60 GHz, which is utilised for specific grid search algorithms (RF, DT). In contrast, an HP Omen-17, which is equipped with an i7-7th generation CPU running at 2.80 GHz and 16 GB of RAM, is used for the other algorithms.



	max_features=none	random state=1	kernel=rbf	hidden layer=100	ccp alpha=0.01	maximum leaf nodes=9	random state=0	max_iter=-1	metric=minkowski	max_depth=2	var_smoothing=1e-09	random state=40	min_samples_split=2	minimum sample leafs=2	random state=none	activation=identity	min_samples_leaf=1	priors=none	learning rate=0.1	n_jobs=none	minimum samples split=2	subsample=1.0	maximum iteration=100	max_features=auto	solver=lbfgs	gamma=scale	criterion=gini	max depth=none	maximum features=auto	n_jobs=1	random state=3	alpha=0.001	leaf size=25	weights=distance	n_estimators=40	algorithm=auto	shuffle=true	neighbors=4	cv=3	Maximum depth=6
Multi-Layer Perceptron	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0
k-Nearest Neighbors	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1	0
Decision Tree	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1
Random Forest	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0
Gradient Boosting	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Support Vector Machine	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gaussian Naïve Bayes	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 7. Grid search parameters: The figure shows each algorithm's parameters and corresponding values in the grid search approach.

3. Results

3.1. Experiments on the Slovak dataset

This section and the following subsections provide various Slovak dataset results on classical machine learning and neural networks. In this research phase, the standard (pre-defined) parameter values are used in all the algorithms.

3.1.1. Classical machine learning algorithms on the Slovak dataset

Table 1 presents the overall classification accuracy, based on various metrics, for the classical machine learning algorithms mentioned in Section 2. A standard scaler was applied to standardize the features by subtracting the mean and dividing all values by the standard deviation.

The results indicate that these algorithms achieve average accuracies in this classification task. For the exact cropped dataset, RF and GB outperform the other models listed in the tables for both the scaled and unscaled datasets, while k-NN shows average performance on unscaled data. It can be concluded that SVM and GNB did not perform well on the training and testing sets. However, SVM's performance improved after scaling, though the algorithm tended to overfit in most iterations.

Moreover, without scaling, the k-NN algorithm attained 78 % training accuracy, whereas the testing accuracy was 66 %. Following scaling, the training and testing accuracies improved by 11 % and 20 %, respectively. Similar enhancements were noted with the SVM algorithm. These results, clearly indicate that scaling was only able to improve the outcomes of these algorithms by a marginal amount.

The accuracy of the Slovak normal cropped dataset is lower than that of the exact cropped dataset. The Slovak dataset experiments (Table 1) show that algorithms such as RF and GBs exhibit signs of overfitting in many instances. Moreover, SVMs tend to overfit but exhibit higher testing accuracy than training accuracy. Table 1 displays a random sampling of these algorithm results; however, testing with various states reveals consistent underperformance and overfitting issues. In contrast, after scaling, the algorithms perform well on the exact cropped dataset. Additionally, on the normal cropped dataset, algorithms struggle to perform adequately. Consequently, it can be inferred that the algorithms applied to the exact cropped datasets outperformed the results of the normal cropped datasets.

These experiments indicate that the choice of algorithms and dataset pre-processing (such as scaling) plays a relatively minor in achieving standard accuracy levels. The results suggest that the RF and GB algorithms provided better results than the other algorithms did. Moreover, the SVMs performed worst on both the training and testing sets.

Table 1
A Comparison of classical algorithms on various Slovak datasets: This table presents the results of six algorithms applied to both the exact and normal cropped versions of the Slovak dataset.

Dataset	Algorithm	Accuracy				Scaling Accuracy			
		Test	Precision	Recall	F1-score	Test	Precision	Recall	F1-score
Exact Cropped	k-Nearest Neighbor	**0.66**	**0.63**	**0.63**	**0.63**	0.86	0.85	0.86	0.85
	Decision Tree	0.82	0.80	0.80	0.80	0.83	0.79	0.81	0.80
	Random Forest	**0.86**	**0.85**	**0.84**	**0.84**	0.84	0.83	0.83	0.83
	Gradient Boosting	**0.85**	**0.84**	**0.84**	**0.84**	0.86	0.85	0.85	0.85
	Support Vector Machine	0.63	0.72	0.59	0.53	0.83	0.85	0.80	0.80
	Gaussian Naïve Bayes	0.65	0.64	0.65	0.65	0.63	0.61	0.60	0.59
Normal Cropped	k-Nearest Neighbor	0.46	0.46	0.46	0.45	0.67	0.68	0.69	0.68
	Decision Tree	0.62	0.67	0.64	0.64	0.65	0.68	0.66	0.66
	Random Forest	**0.72**	**0.72**	**0.72**	**0.72**	0.73	0.73	0.73	0.73
	Gradient Boosting	**0.71**	**0.71**	**0.72**	**0.72**	0.74	0.75	0.75	0.75
	Support Vector Machine	**0.48**	**0.39**	**0.47**	**0.42**	0.75	0.76	0.75	0.75
	Gaussian Naïve Bayes	0.58	0.56	0.58	0.56	0.58	0.55	0.58	0.56

3.1.2. Neural networks on the Slovak datasets

Both the MLP and CNNs are effective in image classification, where CNNs stand out for their ability to understand spatial relationships. Both algorithms are used in this experiment; the results are shown in Figs. 8 and 9. The MLP parameters were selected according to predefined guidelines (see appendix Table 2). The MLP yielded suboptimal results without scaling (Fig. 8), achieving only 71 % training accuracy, 70 % testing accuracy, 70 % precision, and 67 % recall, with corresponding F1-scores on the exact cropped dataset. For the normal cropped dataset, the accuracies were slightly better at 83 % and 80 %, with precision, recall, F1-scores of 81 %. However, significant improvements were observed after scaling, with training and testing accuracies both reaching 92 %, along with 91 % precision, recall, and F1-scores on exact cropped samples. For the normal cropped samples, accuracies were 84 % and 78 %, with precision, recall, and F1-scores of 70 %, 71 %, and 71 %, respectively. Compared to classical machine learning algorithms, scaling significantly enhances the performance of the neural network.

The CNN parameter values were adjusted on the basis of previous experimental findings (see appendix Table 2). For all the CNN models, the Rectified Linear Unit (ReLU) was used as the activation function for all the convolutional layers, and for the final layer, the Softmax activation function with 50 % dropout was used.

The CNN results are the best after fine-tuning. For both datasets, CNNs with similar configurations achieved better results (Fig. 9). The results are higher than those of classical machine learning algorithms and show promising accuracy.

A comparison of the precision, recall, and F1-score of the CNN model for four tree species (normal cropped result) is shown in Fig. 10. The results show that all tree species, except for European silver fir, demonstrated accuracies greater than 85 % across various metrics such as precision, recall, and the F1-score. It is imperative to account for several factors when working with CNNs, including the number of convolutional layers, kernel size, activation functions, batch size, and dropout rate. Here, we tuned these parameters on each dataset, and the best values were considered for this research.

A Comparison of the results of both algorithms (classical and neural networks) reveals that the CNN is more accurate in the dataset. However, when the scaled MLP result (Slovak exact cropped dataset) is considered, the CNN exhibits only a marginal increase in accuracy. Additionally, the correlation between the training and test accuracies for the MLP (scaled result) suggests potential unreliability, indicating a risk of overfitting in the future. Conversely, CNNs present lower risks of overfitting and underfitting than the MLP algorithm does.

Moreover, the CNN achieves superior accuracy to all algorithms utilised on the Slovak normal cropped and exact cropped datasets. Compared with the Slovak exact cropped dataset, classical machine learning algorithms and MLPs fail to maintain standard accuracy levels. The experiments on both datasets show that the exact cropped dataset performs well on both classical machine learning algorithms and neural networks. In contrast, the normal cropped dataset performs well only with the CNN. Furthermore, CNNs consistently provide standard accuracy rates across various datasets.

3.1.3. Grid search on the Slovak dataset

An evaluation of the grid search results shown in Fig. 11 indicates that the MLP gains greater accuracy after parameter tuning. Moreover, DT has the lowest accuracy among all the grid search results. Moreover, in this case, scaling also increases the accuracy of the algorithms except for the RF. The corresponding accuracy metrics, including precision, recall, and F1 scores for RF and GB algorithms, are provided in Appendix Table 5.

Additionally, it is essential to note that the k-NN algorithms provided better accuracy without overfitting after scaling. The DT boosted the training accuracy to a maximum of 100 % and the testing accuracy by 8

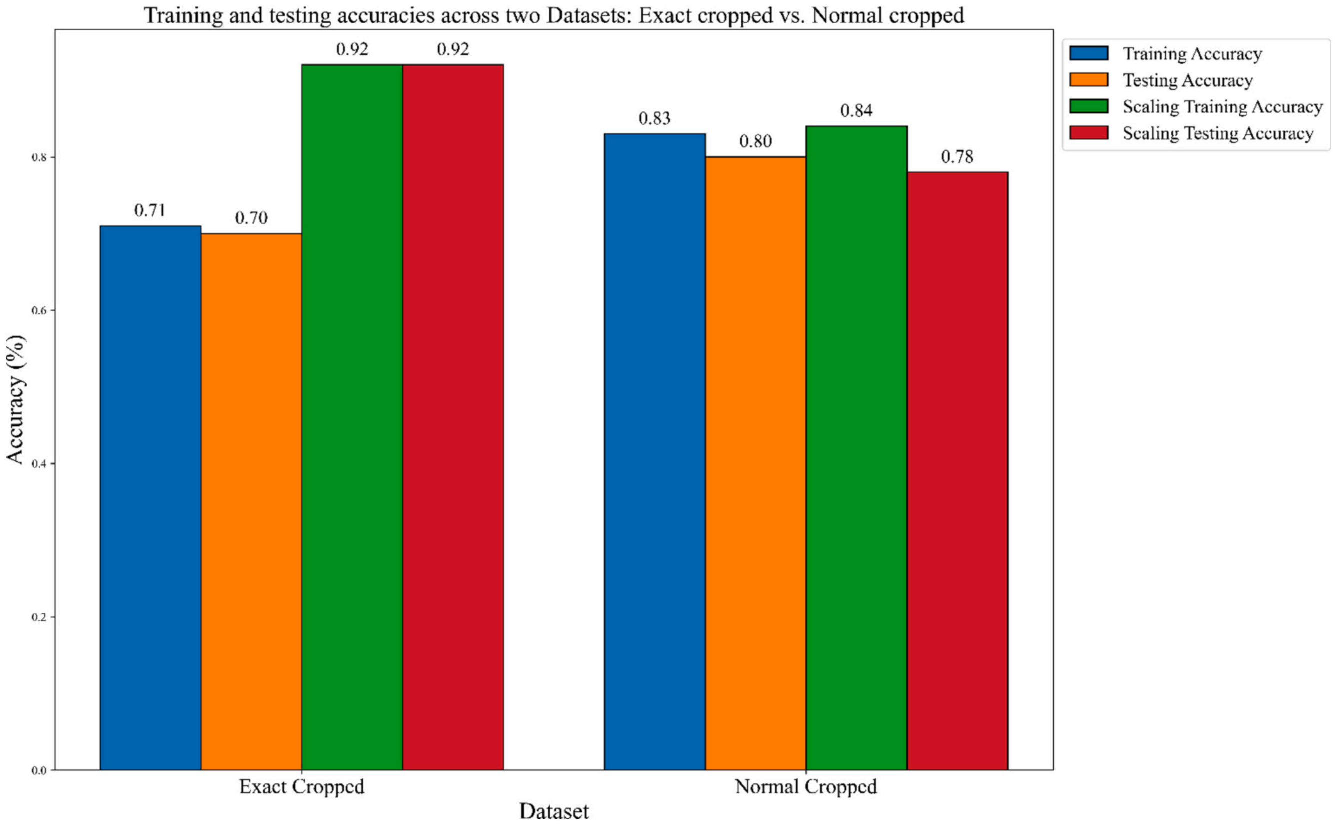


Fig. 8. Multilayer perceptron on the Slovak dataset: This figure illustrates the results of the MLP on both the exact and normal cropped versions of the Slovak dataset. The corresponding accuracy metrics, including precision, recall, and F1-scores, are provided in Appendix Table 3.

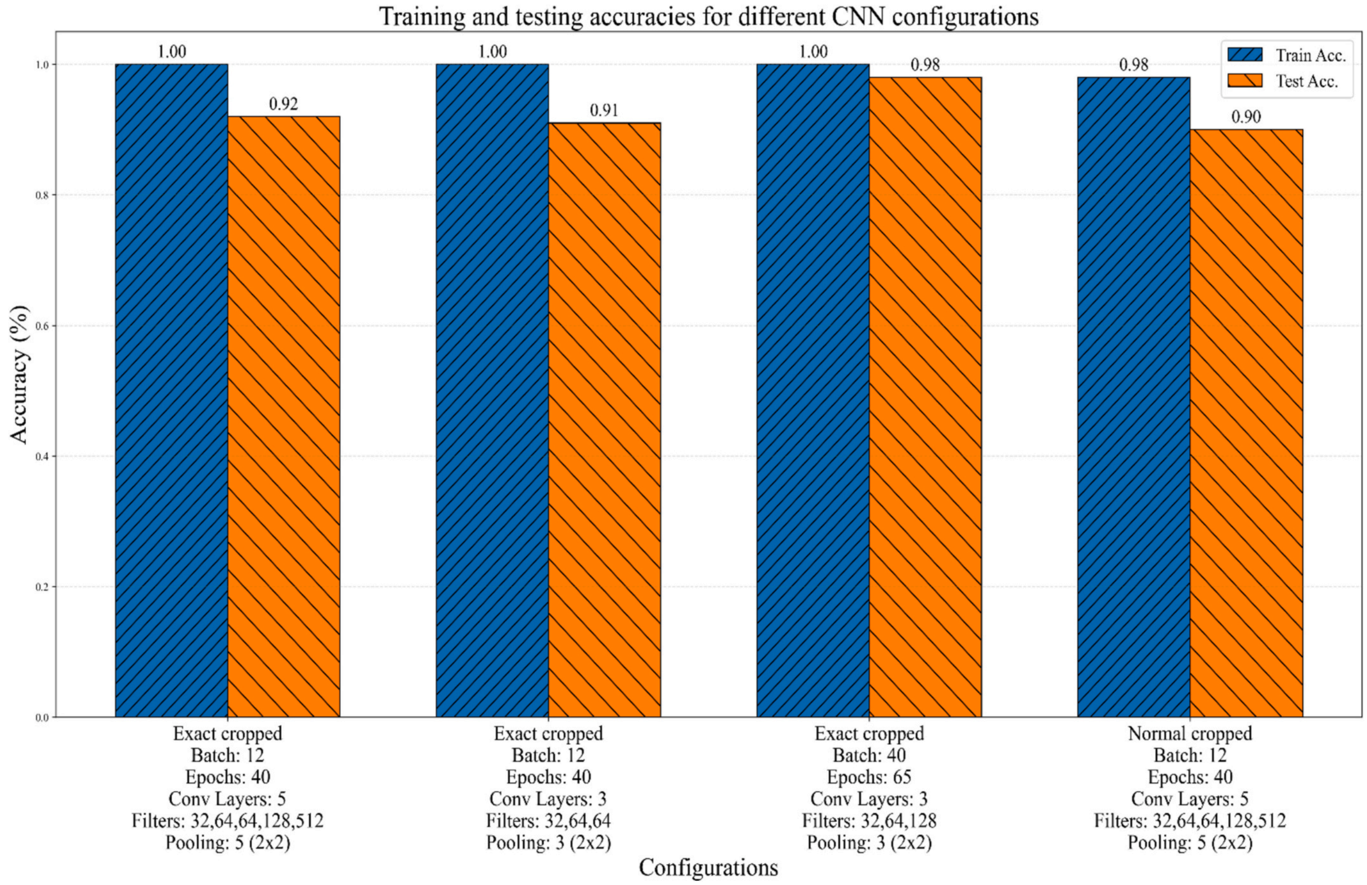


Fig. 9. CNN parameters and results on the Slovak exact cropped and normal cropped datasets: This figure represents the results obtained using the specified parameters across various values, with a 3 × 3 kernel applied to all filters. The corresponding accuracy metrics are provided in Appendix Table 4.

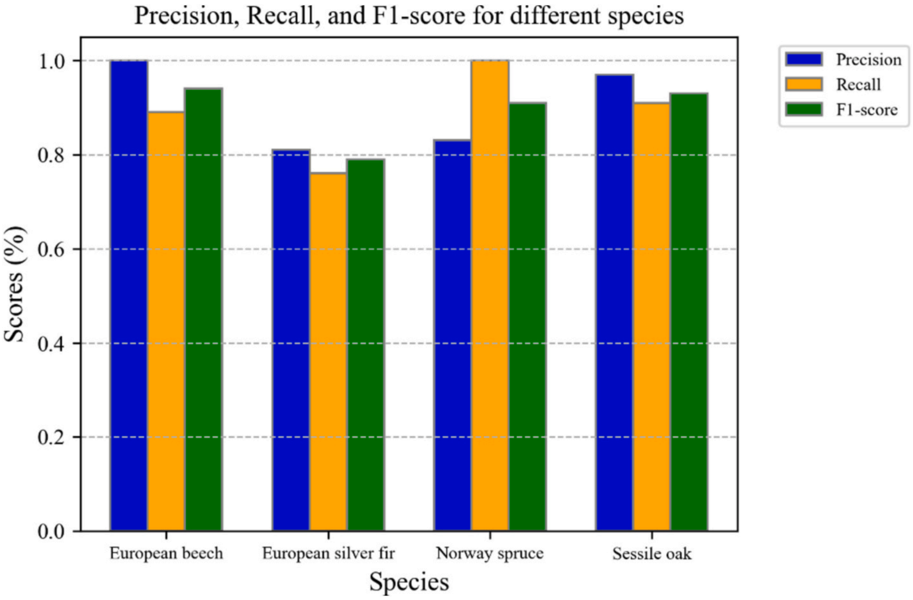


Fig. 10. Tree species classification accuracy graph: This figure illustrates the class-wise accuracy of the CNN model on the Slovak normal cropped dataset.

%, reaching 80 %. However, the RF could not increase the testing accuracy; instead, the accuracy decreased by 6 %. The final algorithm, the GB, also increased the testing accuracy by 7 %. Moreover, we can see that the algorithms tend to overfit the training datasets, mainly because of the amount of data considered in the study. Moreover, we have only a few parameters (GLCM); therefore, the algorithms can easily overfit the training datasets, leading to higher accuracies in training sets.

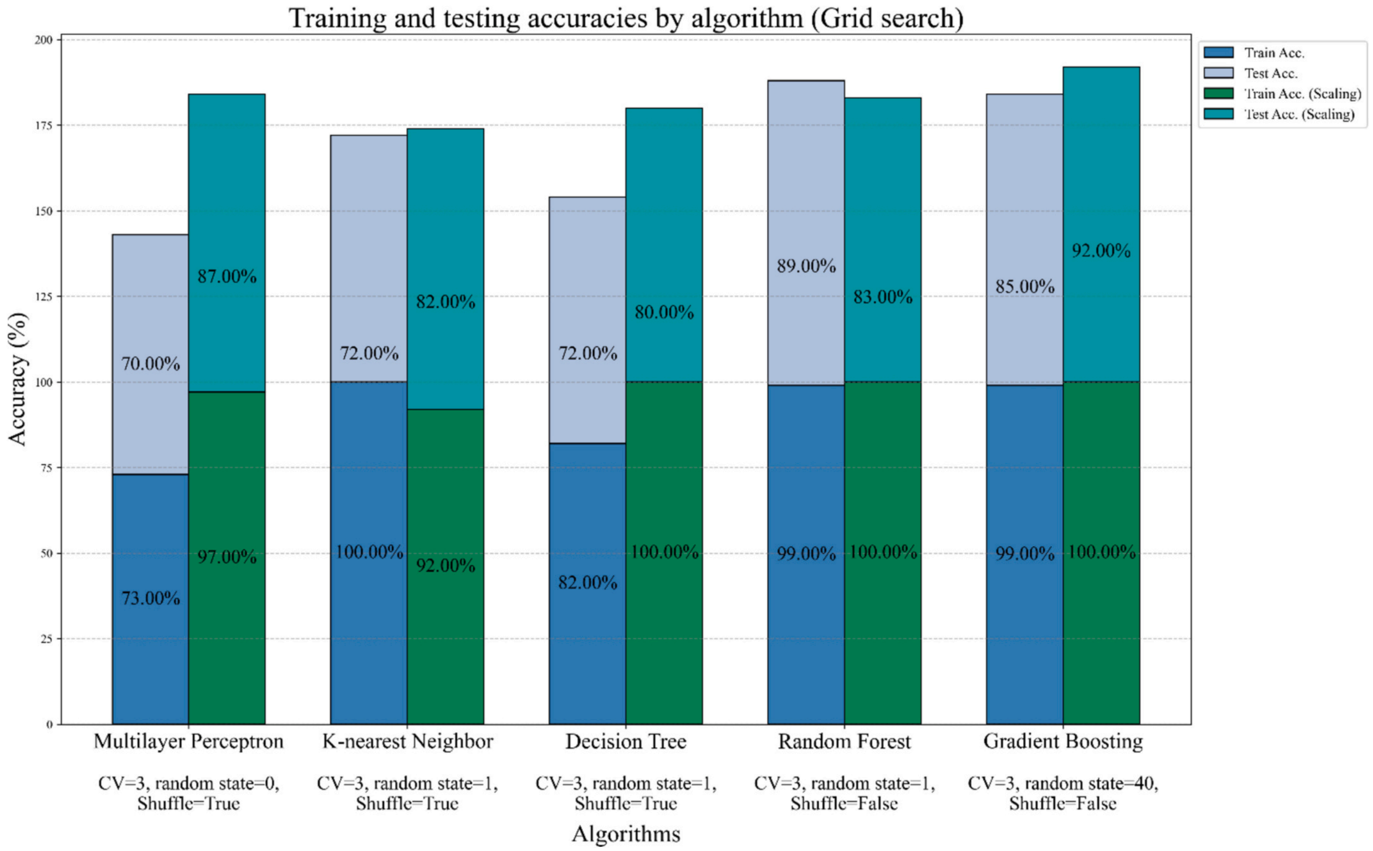


Fig. 11. Grid search on Slovak exact cropped datasets: The figure shows the performance comparison of five models on Slovak exact cropped datasets obtained through grid search optimisation.

3.2. *Experiments on the CZU datasets*

The following subsections provide the experimental results of the classical and neural network algorithms on the CZU datasets. Like the previous experiments on various datasets, we also consider the pre-scaling and post-scaling of the datasets on pre-defined parameters and parameter values.

3.2.1. *Classical machine learning algorithms on the CZU dataset*

Considering the results of classical machine learning algorithms on the CZU exact cropped dataset (Table 2) and the Slovak exact cropped datasets (Table 1), the CZU dataset did not attain the same accuracy range as the Slovak dataset. In the CZU exact cropped dataset, the DT, RF, and GB algorithms tend to overfit the dataset. The study revealed that the GNB algorithm displayed the lowest average accuracy among traditional algorithms (when scaling), achieving a rate of 51 %. Conversely, the SVM and GNB classifiers achieved the lowest accuracy of 63 % for the Slovak exact cropped dataset.

In summary, compared with the Slovak exact cropped dataset, the machine learning algorithms applied to the CZU dataset did not achieve higher accuracy. Additionally, when comparing the results after scaling on both datasets, the CZU dataset failed to outperform the Slovak exact cropped dataset in terms of accuracy. The results obtained from the CZU normal dataset (Table 2) are less accurate than those obtained from the

Table 2
Performance comparison of classical algorithms on CZU datasets: This table presents the results of various classical machine learning algorithms on both the exact cropped and normal cropped datasets, with and without scaling. The results highlight that Random Forest and Gradient Boosting achieved the highest accuracy scores of 0.76 and 0.77, respectively, after scaling, indicating their superior performance in classifying the dataset. In contrast, Gaussian Naïve Bayes and Support Vector Machine exhibited the lowest accuracy, particularly on the normal cropped dataset.

Dataset	Algorithm	Accuracy				Scaling Accuracy			
		Test	Precision	Recall	F1-score	Test	Precision	Recall	F1-score
Exact Cropped	k-Nearest Neighbor	0.65	0.57	0.58	0.57	0.76	0.73	0.76	0.74
	Decision Tree	**0.64**	**0.60**	**0.57**	**0.56**	0.69	0.65	0.63	0.63
	Random Forest	**0.75**	**0.66**	**0.67**	**0.64**	0.73	0.70	0.66	0.64
	Gradient Boosting	**0.76**	**0.66**	**0.69**	**0.66**	0.77	0.70	0.70	0.69
	Support Vector Machine	0.54	0.41	0.45	0.41	0.74	0.71	0.70	0.68
	Gaussian Naïve Bayes	0.55	0.57	0.56	0.54	**0.51**	**0.50**	**0.52**	**0.48**
Normal Cropped	k-Nearest Neighbor	0.57	0.51	0.59	0.52	0.68	0.67	0.71	0.68
	Decision Tree	0.44	0.41	0.42	0.39	0.57	0.42	0.57	0.46
	Random Forest	0.70	0.70	0.78	0.72	0.72	0.75	0.74	0.73
	Gradient Boosting	0.63	0.59	0.68	0.61	0.70	0.70	0.74	0.71
	Support Vector Machine	0.51	0.13	0.25	0.17	0.64	0.71	0.64	0.60
	Gaussian Naïve Bayes	0.37	0.35	0.38	0.30	0.43	0.49	0.55	0.44

other datasets. Moreover, algorithms such as DT, RF, SVM, and GB exhibit overfitting even after scaling. This overfitting is attributed primarily to the dataset size and the proportion of data utilised for the study. Tasks such as tree species classification and detection demand substantial data for optimal performance with classical machine learning algorithms and standard/basic neural networks such as the MLP. Moreover, refining the data focus, such as exact cropping, can yield improved outcomes.

3.2.2. Neural networks on the CZU datasets

The MLP applied to the unscaled CZU exact cropped dataset (Table 3) did not yield improved results compared with previous unscaled results from other datasets, particularly when considering MLP performance across different datasets. It achieves 80 % training accuracy and 76 % testing accuracy on unscaled data, which increases slightly to 89 % training accuracy and 81 % testing accuracy after scaling. The increased overlap between the bark images helps maintain consistent average accuracy rates. Additionally, the intra-class similarity between tree species is lower in the CZU dataset than in the Slovak dataset. Thus, this approach minimises misclassification between different tree species.

The results obtained from the CZU normal dataset (Table 3) are less accurate than those obtained from the other datasets. Furthermore, the lower number of sample species per class in the CZU dataset contributes to the classification results. These findings suggest the importance of considering more overlapping images for improved classification, especially when employing classical machine learning and standard neural networks.

When examining the results of the CNN (Table 4), the above-mentioned observations regarding classical algorithms hold true. The CNN algorithm achieves an accuracy of 93 %, aligning closely with other CNN results across different datasets. The algorithm achieves a minimum accuracy above 75 % with the parameters in Table 4. Notably, the Large-leaved linden species were highly misclassified into Norway maple in most of the iterations. (See Table 5.)

Additionally, the classification accuracy for Large-leaved linden was slightly lower than that for the other species (see Appendix Table 6), with a precision of 43 %. However, for all other species, the accuracy of the calculations consistently exceeded 73 %. The CNN outcomes for the CZU normal dataset (Table 4) remain consistent with prior findings despite variations in layer configurations. The ReLU emerged as the optimal activation layer for this classification task through experimentation. Convolutional layers exhibit thresholds defined by their minimum and maximum values, contingent upon dataset size and features. Exploring diverse kernel combinations, such as 2 × 3, 3 × 3, and 3 × 4, is viable. However, the 3 × 3 kernel size generally proves superior for most scenarios.

These experiments underscore the critical role of data volume in determining neural network accuracy. Although scaling and augmentation techniques can enhance results, their efficacy is not limited. Therefore, it is crucial to carefully define the model architecture and parameters according to the dataset size to achieve optimal performance.

3.3. CZU exact cropped and normal cropped dataset combination

The combination of exact cropped and normal cropped results yields only marginal improvement. These results show that better datasets and feature combinations do not provide better accuracy. In this experiment, the performance of the SVM was especially remarkable, as it encountered difficulties in accurately classifying two classes, resulting in a 0 % success rate for Large-leaved linden and Scots pine (refer Appendix Table 7). Furthermore, even with scaling, all the algorithms failed to yield enhanced results.

In CNNs, the accuracy remains consistent across all datasets. This consistent accuracy suggests that CNNs can perform effectively across diverse datasets without specific data segmentation, such as exact cropping. In the context of CNNs, the testing accuracy exceeds 82 %, using identical parameters across both configurations (as detailed in Fig. 12). Furthermore, slight variations are observed when examining individual species classification accuracy compared with other CNN results.

Research has revealed that merging normal cropped and exact cropped datasets does not substantially increase the accuracy of classical machine learning algorithms. This combination may lead to a decrease in the accuracy of individual species classification. However, the CNN algorithms attempt to maintain better accuracy.

3.4. Nonlinear dataset experiments

Nonlinear datasets are applied to both classical machine learning and neural networks to understand the effectiveness of these algorithms. The results from the Nonlinear dataset, which are shown in Table 6, indicate that half of the algorithms (DT, RF, and GB) exhibit overfitting tendencies, with GNB methods occasionally showing overfitting. This study employs pre-defined parameters and parameter values for both classical and neural network algorithms.

Despite this, compared with the results of the Slovak normal cropped and exact cropped datasets, the Nonlinear dataset poses a challenge in achieving the desired outcomes. However, specific algorithms, such as k-NN, RF, DT, and SVM, demonstrate improved testing accuracy on the scaled dataset (Nonlinear) by 10 %, 5 %, 3 % and 35 %, respectively, compared with the Slovak normal cropped dataset.

For the Nonlinear dataset, the MLP achieves lower accuracy on unscaled data, with 21 % training accuracy and 16 % testing accuracy, representing its lowest performance compared with all algorithms across different datasets. However, following the scaling process, the algorithm shows promising results, with 85 % training accuracy and 78 % testing accuracy (refer Appendix Table 8). Compared with the MLP performance on the Slovak normal cropped dataset, this model achieves a minimum accuracy of 78 % on the scaled dataset. However, the dataset without scaling proves to be an inefficient model.

In the case of the CNN algorithm (Fig. 13), the dataset performs well, exhibiting excellent accuracy compared with the Slovak exact cropped dataset results. However, compared with the exact parameters used in the dataset, the Nonlinear dataset yields slightly lower accuracy than the exact cropped dataset. Furthermore, the Nonlinear dataset fails to achieve 100 % training accuracy even after 150 epochs. It shows the effectiveness of the nonlinear patterns in training neural networks; even

Table 3
Performance of MLP on the CZU dataset: This table presents the accuracy, precision, recall, and F1-score for the Multilayer Perceptron on both the exact cropped and normal cropped datasets, with and without scaling. The results indicate a notable improvement in performance after scaling, with the exact cropped dataset achieving an accuracy of 81 %, compared to 76 % without scaling. In contrast, the normal cropped dataset showed a significant increase from 43 % to 75 % accuracy after scaling, highlighting the effectiveness of scaling in enhancing model performance.

Dataset	Accuracy				Scaling Accuracy			
	Test	Precision	Recall	F1-score	Test	Precision	Recall	F1-score
Exact Cropped	0.76	0.72	0.73	0.72	0.81	0.76	0.74	0.73
Normal Cropped	0.43	0.43	0.43	0.33	0.75	0.76	0.75	0.74

Table 4
CNN parameters and performance results on CZU datasets: This table summarizes the performance metrics, including test accuracy, precision, recall, and F1-score, for the CNN model applied to both the exact cropped and normal cropped datasets, utilising 5 convolutional layers with filters set to 32, 64, 64, 128, and 512, and a pooling size of 2 × 2. The results demonstrate a significant improvement in accuracy on the exact cropped dataset, increasing from 81 % at 40 epochs to 93 % at 65 epochs. For the normal cropped dataset, the model achieved an accuracy of 80 % at 40 epochs, which slightly decreased to 79 % at 65 epochs. In this experiment, we utilised a 3 × 3 kernel, a batch size of 12, and a dropout rate of 50 % for both models.

Dataset	Epochs	No. Conv. layers	Filters	Pooling	Test Acc.	Precision	Recall	F1-score
Exact cropped	40	5	32,64,64,128,512	5 (2*2)	0.81	0.79	0.72	0.75
	65	5	32, 64,64,128,512	5 (2*2)	0.93	0.92	0.88	0.90
Normal cropped	40	5	32, 64,64,128,512	5 (2*2)	0.80	0.82	0.77	0.78
	65	5	32, 64,64,128,512	5 (2*2)	0.79	0.80	0.79	0.76

Table 5
Performance of classical algorithms on combined CZU datasets: This table presents the accuracy, precision, recall, and F1-score of various classical machine learning algorithms applied to the combined exact and normal cropped datasets, both with and without scaling. The results indicate that while k-Nearest Neighbor and Random Forest achieved relatively consistent performances, the Decision Tree algorithm experienced a noticeable decrease in accuracy after scaling, dropping from 54 % to 37 %. This decline suggests that scaling may not benefit all algorithms equally and highlights the need for careful consideration when selecting models for classification tasks.

Algorithm	Accuracy				Scaling Accuracy			
	Test	Precision	Recall	F1-score	Test	Precision	Recall	F1-score
k-Nearest Neighbor	0.56	0.52	0.55	0.53	0.60	0.56	0.58	0.57
Decision Tree	0.54	0.61	0.54	0.55	0.37	0.44	0.40	0.37
Random Forest	0.70	0.68	0.70	0.69	0.59	0.55	0.64	0.53
Gradient Boosting	0.70	0.68	0.70	0.69	0.54	0.51	0.51	0.50
Support Vector Machine	0.43	0.17	0.26	0.20	0.56	0.52	0.42	0.41
Gaussian Naïve Bayes	0.36	0.35	0.39	0.31	0.39	0.42	0.41	0.35

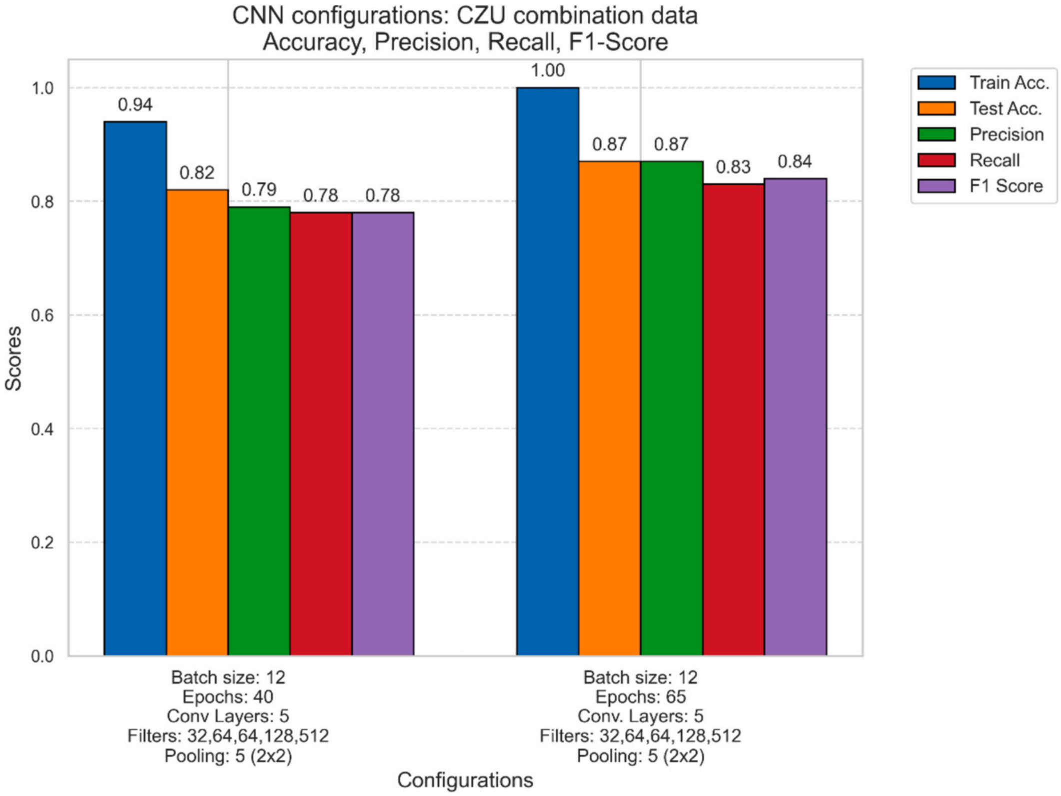


Fig. 12. CNN parameters and results of the combination of the exact cropped and normal cropped CZU datasets.

on smaller datasets, it can provide better accuracy without overfitting. Additionally, upon examining the confusion matrix (see appendix Table 9), all species demonstrated a minimum classification accuracy of over 80 %. Notably, the CNN performs well on Nonlinear datasets.

The experiments show that the dataset's quality is essential for better accuracy, especially in classical machine learning algorithms. Preprocessing techniques such as scaling can increase the accuracy of these algorithms to a certain level; additionally, these algorithms cannot provide constant accuracies over various datasets, especially Nonlinear datasets. However, the CNN algorithm maintained higher accuracy in various datasets even though the CNN provided better accuracy in Nonlinear datasets. The CNN algorithms learned from more complex

Table 6
Performance of classical machine learning algorithms on nonlinear data, with and without scaling: The table highlights accuracy, precision, recall, and F1-score for each algorithm across both configurations. Notably, scaling improves the performance of most algorithms, particularly the SVM, which shows a substantial jump in accuracy from 42 % to 77 % when scaling is applied. Gradient Boosting consistently delivers strong results across all metrics, achieving the highest accuracy of 75 %, with minimal differences between the scaled and unscaled configurations. Conversely, k-NN and RF exhibit moderate improvements, while DT and GNB show more stable performance.

Algorithm	Accuracy					Scaling Accuracy		
	Test	Precision	Recall	F1-score	Test	Precision	Recall	F1-score
k-Nearest Neighbor	0.62	0.65	0.63	0.63	0.72	0.72	0.73	0.71
Decision Tree	0.67	0.67	0.67	0.67	0.70	0.72	0.72	0.70
Random Forest	0.64	0.65	0.66	0.65	0.69	0.70	0.71	0.68
Gradient Boosting	0.75	0.76	0.76	0.76	0.75	0.76	0.76	0.75
Support Vector Machine	0.42	0.50	0.43	0.40	0.77	0.81	0.79	0.77
Gaussian Naïve Bayes	0.65	0.69	0.66	0.64	0.67	0.72	0.67	0.66

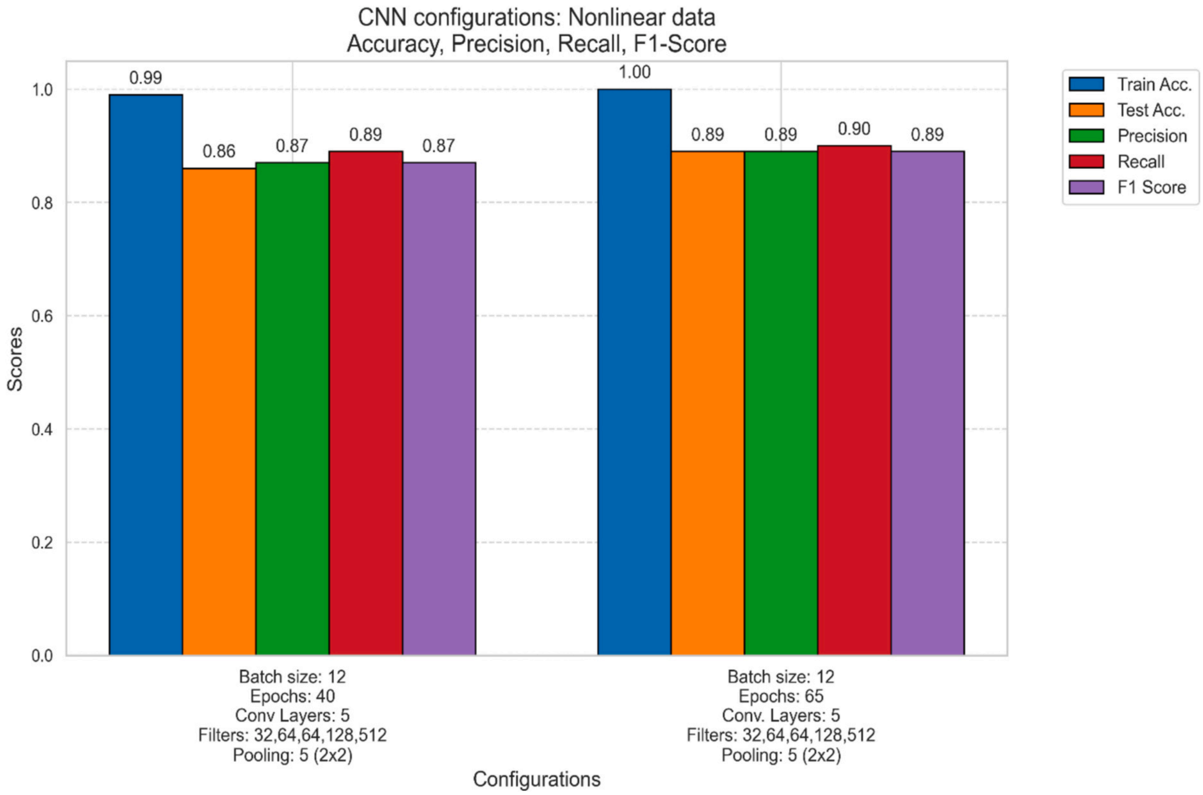


Fig. 13. CNN parameters and results on the Nonlinear dataset.

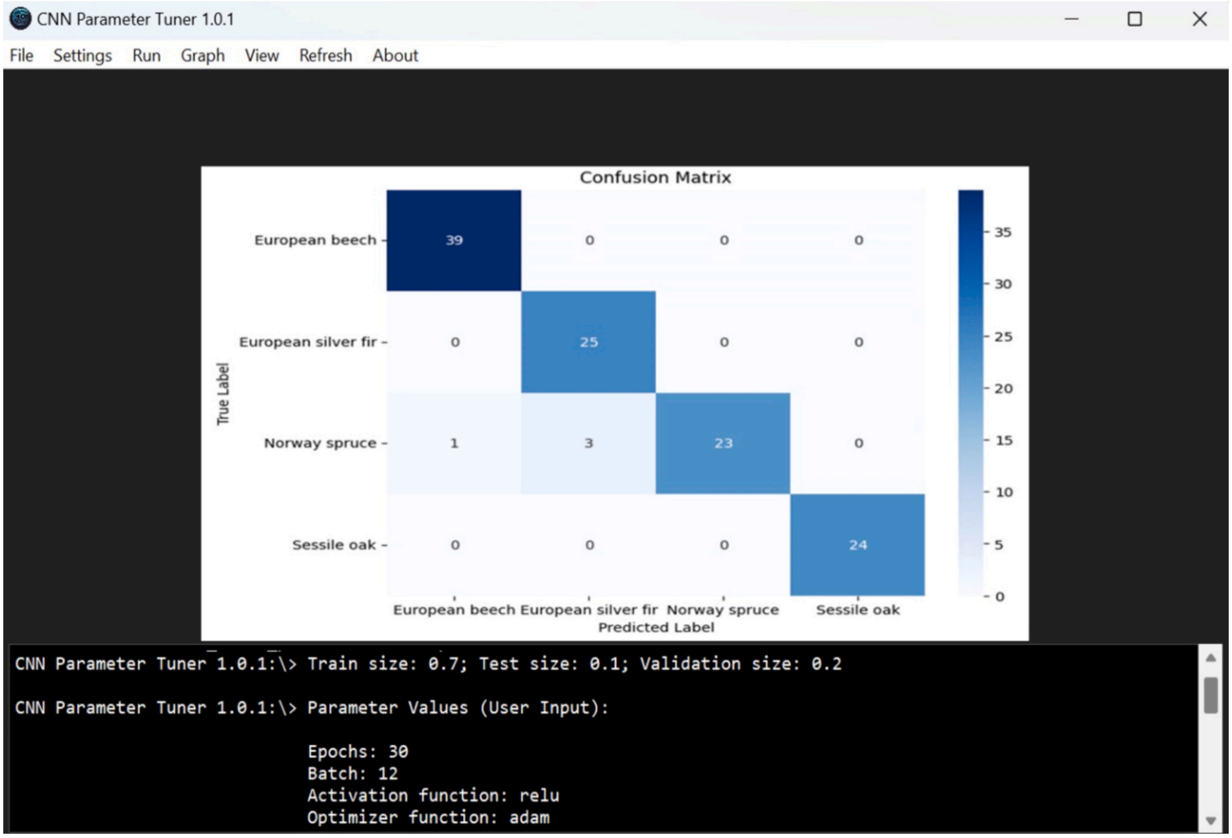
nonlinear patterns, making the image classification as robust as all other algorithms.

However, we need to change the parameter values on the basis of our datasets. The algorithms can provide better results even when less computational power is utilised. In this experiment, we found that the CNN is the best algorithm for tree species classification; however, fine-tuning the CNN is the most complex task for interdisciplinary researchers. Therefore, we developed a user-friendly Windows-based application for fine-tuning CNNs for any image dataset (jpg and png); the following section delves into this application.

3.5. CNN parameter tuner

Our research revealed that CNNs are more robust than other algorithms are. Furthermore, we emphasise the necessity of parameter tuning tailored to the dataset under scrutiny. As a solution, we have developed Windows-based research software capable of loading image data (in formats such as jpg and png) for classification purposes, as illustrated in the accompanying figure below (Fig. 14). This application has integrated essential features, including loading and analysing various image formats. Additionally, the application facilitates classification by employing folder names as class labels. Each class label is automatically converted to numerical values for easier processing. The users are empowered to specify the training-testing-validation ratio; without user-defined values, the application automatically sets it to 70 % for training and allocates the remaining for testing and validation. Furthermore, we have incorporated options for users to adjust parameters such as the number of epochs and batch size, dynamically adapting to dataset sizes.

The convolutional layers of the network utilise the ReLU activation function, whereas the final dense layer allows users to select their preferred activation function. Similarly, users can customise optimiser and loss functions according to their preferences. Additionally, in the results section, users can visualise outputs through interactive confusion matrices and graphs, with the added functionality of saving results as reports for future reference. Additionally, to facilitate ease of use, we developed a comprehensive user handbook (Kottilapurath Surendran and Mokros, 2024). This research software was developed via Python.



```
CNN Parameter Tuner 1.0.1:\> Train size: 0.7; Test size: 0.1; Validation size: 0.2

CNN Parameter Tuner 1.0.1:\> Parameter Values (User Input):

                             Epochs: 30
                             Batch: 12
                             Activation function: relu
                             Optimizer function: adam
```

Fig. 14. CNN parameter tuner 1.0.1: This figure shows the results of tree species classification via the CNN parameter tuner software. In this particular tree species classification task, we achieved an accuracy of 96.5 %, and the individual classification accuracy (per class) was also higher.

Initially, we developed various backend functions to load the datasets, define training testing sizes, and other options such as epochs, batch size, and extracting features from various image formats. Furthermore, the other essential functions for defining activation functions are optimisation, loss, etc. After this, the main component of the machine learning model is defined as another function. The proposed CNN model's training testing, evaluation, and other essential functions are defined here. For GUI development (frontend), we used the 'tkinter' library. The main window (root) is initialised, and various widgets, such as buttons, labels, and text areas, are used. All the user inputs from these widgets are transmitted to the corresponding functions. The final script is subsequently transformed into a standalone application via the PyInstaller package. Furthermore, we leveraged the Inno Setup Compiler software to convert the Python standalone application into a standalone Windows application (.exe).

4. Discussion

4.1. Tree species classification using leaves and bark

Tree species classification presents many challenges, especially when a single organ or a distinct part of the tree is considered. The tree species classification was initially centred on tree leaves (Zhou et al., 2016); both the classical and neural network algorithms provided better accuracies. Vizcarra et al., 2021 created a large leaf dataset consisting of 59,441 images for species classification, and they deployed various deep learning algorithms, such as AlexNet, VGG-19, ResNet-101, and DenseNet-201. Their VGG-19 model achieved 96.64 % training accuracy and 96.52 testing accuracy.

However, we cannot depend on leaves, especially for species classification, because of their instability (not a constant feature or organ). Therefore, researchers have moved on to combining leaves and bark, providing better accuracy (Jendoubi et al., 2020; Zhao et al., 2020). Bertrand et al., 2018 investigated how to combine the features extracted from leaf and bark images to recognise trees and developed an application named Folia. In their approach, they initially extracted leaf characteristics such as shape, apex, margin, etc. In the second phase, the bark edges and other features were extracted, and an SVM classifier was used. Ameur et al., 2016 employed a fusion system (two subclassifications) for tree species classification using leaves. The authors used an RF classifier based on five morphological features: shape, lobe shape, base, apex, and margin. In the initial phase, they used expert knowledge to map classifier outputs from a subclass level to the species level. The second approach directly maps the classifier outputs to the species level. The authors combined the classification results with an adaptive fusion system with a hierarchical cascade strategy. Additionally, it is essential to note that the bark images for this study also considered a small portion of the bark, similar to other studies.

Researchers who have focused mainly on bark patterns for tree species classification have focused on small areas of the bark (Blaanco et al., 2016; Bressane et al., 2015; Sulc and Matas, 2013; Zhi-Kai et al., 2006). Bressane et al., 2015 study used co-occurrence descriptors to identify tree species. The authors transformed RGB images to HSV colour space and generated co-occurrence matrices from the grayscale images to extract texture descriptors such as contrast, correlation, energy, and homogeneity. They applied these descriptors to the Binary Decision Tree (BDT) for the final classification process and they achieved an accuracy of 87 %. Sulc and Matas, 2013 employed feature-mapped multi-scale descriptors formed by concatenating Local Binary Patterns (LBPs) histograms. These feature maps were able to approximate the histogram intersection kernel, resulting in a significantly improved accuracy. Ganschow et al., 2019 achieved 96.7 % accuracy in species classification using bark patterns while concentrating on smaller portions of the bark for classification tasks. The literature shows that while using ConvNeXt, CNNs achieve a minimum accuracy of 97 % for classifying nearly 33 tree species (Cui et al., 2023); research focused on small portions of the bark images has yielded higher accuracy. Wu et al., 2021 developed a portable application for bark identification named Deep BarkID; here, the authors used CNN architectures such as ResNet and MobileNet. They also investigated the possibility of transfer learning. In the other studies, the authors (Benassi et al., 2024; Debaleena et al., 2020) considered only a small portion of the bark for classification purposes; they did not investigate the other possibilities.

Li et al., 2023 constructed a new tree trunk dataset and proposed a deep learning model called TrunkNet to detect and segment tree trunks. The proposed model uses a multiscale attention-based mechanism to effectively combine local and global contextual information, enabling it to accurately identify and segment tree trunks. The model performed well compared with other deep learning models. Moreover, few studies (Homan and du Preez, 2021) have focused on developing automated tree species identification systems, especially those focused on semi-supervised learning. Here, the authors used labelled and unlabelled data to increase classification accuracy. Therefore, the proposed methodology involves a two-step process: first, tree features such as leaf and bark characteristics are recognized via using binary classification, and second, species classification is conducted separately for these features. The authors used various algorithms, such as CNNs and Semi-Supervised Learning (SSL), enhanced by EfficientNet, and they achieved accuracies of 94.04 % for leaf classification and 83.04 % for bark classification.

The wide use of neural networks in interdisciplinary areas has forced most researchers to apply these algorithms in forestry, especially for tree species classification tasks. The most exciting fact is that the researchers did not investigate the main factors that can deviate from the standard accuracies, such as datasets, fusion approaches in classical machine learning, etc. Additionally, it is crucial to ensure that we need to tune the algorithms on the basis of our datasets. These are the most significant knowledge gaps obtained through the literature.

4.2. *Assessment of achieved results*

This research revealed that dataset format and algorithm selection are the most important aspects of increasing accuracy in classification tasks. This study used various classical machine learning and deep learning algorithms across various datasets and their combinations. Here, we discuss the three main datasets utilised in this study: Slovak exact cropped, CZU exact cropped, and Non-linear dataset. The diverse nature of the datasets used in this study played a major role in understanding the weaknesses of various algorithms. To explore the differences in algorithm performance across these datasets scientifically, we conducted a Friedman test (Pereira et al., 2015). The results, with a Friedman test statistic of 29.099 and a *p*-value of 0.00013, indicate significant differences in performance across the algorithms when applied to these diverse datasets. This statistical result suggests that the dataset's characteristics influence the algorithm's performance more. Following the Friedman test, we performed a Nemenyi post-hoc test to identify which specific algorithm pairs exhibited significant differences in performance (Table 7).

The results show significant differences between various algorithm pairs, such as SVM and CNN; here, the SVM underperformed compared with CNN ($p = 0.045$), especially in the, CZU and Non-linear datasets. Additionally, the CNN and GNB methods also showed marginal differences ($p = 0.001$). No significant differences were detected for most algorithm pairs, such as MLP, k-NN, DT, RF and GB.

Moreover, the results from segment-specific datasets and CNN algorithms performed well compared to other datasets and algorithms. It indicates that segment-specific datasets can provide consistent accuracy across all algorithms. While considering the algorithms, it shows that CNN can provide higher accuracy on any dataset, even complex ones; however, there is a higher chance of overfitting.

4.3. *Challenges: Dataset*

Our study focused on a consistent attribute, the bark, as in previous studies. Tree bark is a fundamental characteristic that preserves consistent traits and results in relatively few seasonal fluctuations. Establishing a proficient classification task centred on bark necessitates the availability of a robust dataset. Regrettably, existing scholarly discourse highlights a deficiency in adequate datasets in Europe (Boudra et al., 2022; Fiel and Sablatnig, 2011; TRUNK12, 2022), notably a scarcity of accessible open-source datasets within the European context.

Creating a quality dataset is crucial for improving the classification methodology. To conduct this study, we developed two primary datasets from the Czech Republic and Slovak Republic, adhering to the criteria and methodologies established in the previously utilised BarkNet 1.0 dataset (Carpentier et al., 2018) and associated research methodologies. We also generated a Nonlinear dataset from the Slovak dataset to find the most robust algorithms. Moreover, a dataset combination is also considered the fourth dataset for this study. Notably, we considered the dataset size similar to that of other researchers. However, it is important to note that those datasets' sizes are significantly smaller, particularly for CNN algorithms. Therefore, we need to create a large dataset with more images; this is another limitation of the proposed study.

4.4. *Challenges: Algorithm overfitting*

The deployment and evaluation of algorithms constitute crucial aspects of this study. It is not uncommon for algorithms to confront obstacles such as overfitting and underfitting when processing data. Consequently, fine-tuning parameters according to the dataset's inherent characteristics and features becomes imperative to mitigate these challenges effectively. While scaling techniques can help alleviate these issues, they may not always provide a complete solution. Therefore, careful parameter tuning is essential to address these concerns. Finding suitable parameters and corresponding values might take longer; in this research, we only discussed the best parameter and the corresponding values we achieved on our datasets.

Our experiments revealed that classical algorithms such as DT, RF and GB overfit on most training datasets (Figs. 9, 13, Table 4). However, these algorithms do not overfit the test or validation sets. The main reason is that the tree species classification tasks centred on bark images

Table 7
Nemenyi post-hoc test results: Here, the diagonal cells represent the comparison of an algorithm with itself, and the off-diagonal cells represent the p-value for the pairwise comparison between corresponding algorithms.

Algorithm	MLP	CNN	k-NN	DT	RF	GB	SVM	GNB
MLP	1.000	0.900	0.900	0.833	0.900	0.900	0.587	0.021
CNN	0.900	1.000	0.262	0.137	0.622	0.900	0.045	0.001
k-NN	0.900	0.262	1.000	0.900	0.900	0.798	0.900	0.364
DT	0.833	0.137	0.900	1.000	0.900	0.622	0.900	0.552
RF	0.900	0.622	0.900	0.900	1.000	0.900	0.900	0.102
GB	0.900	0.900	0.798	0.622	0.900	1.000	0.364	0.006
SVM	0.587	0.045	0.900	0.900	0.900	0.364	1.000	0.798
GNB	0.021	0.001	0.364	0.552	0.102	0.006	0.798	1.000

do not have enough data to fit these models. Therefore, these models were highly overfit on training sets. Another question is as follows: Why are the test and validation sets not overfitted on these models? The reason is that, in classical machine learning, we must define feature extractors separately to extract the necessary features from the datasets. On the basis of these extracted features, the algorithms perform classification tasks. Here, we used the GLCM as a feature extractor, and this feature extractor focuses on variables such as contrast, dissimilarity, homogeneity, correlation and energy and the different angle sets of these particular features. In the GLCM, the function performs feature extraction while converting the original images into a grayscale format, leading to a loss in colour information. Therefore, the algorithms are restricted only to work on the abovementioned variables and tend to provide less accuracy on the testing and validation sets.

Moreover, the application of deep neural networks such as CNNs also causes overfitting in training and testing sets (Fig. 9). Unlike classical machine learning algorithms, in deep neural networks, we do not need to specify the feature extractors; the algorithm itself contains a function for extracting feature sets. In the CNN, the feature extractor also takes colour information, and it is able to extract more than 12,000 feature combinations from smaller datasets. In the bark images, it is clear that most species are slightly different from each other on the basis of their colour properties and patterns. This colour feature provides more stability in classifying species with higher accuracies in deep neural networks. Moreover, deep neural networks require a large amount of data to provide standard results without overfitting. We also do not have enough data in our experiments; we generated datasets such as the already available bark datasets in Europe. This is why the CNN models overfit the training and testing sets. Importantly, the CNN algorithm is not overfit in our Nonlinear dataset. In the Nonlinear dataset, we tried to make complex patterns while deviating from the linearity in the images; therefore, the dataset became complex for the CNN model (Fig. 13), so the model was not overfitting. However, the CNN algorithm could understand and differentiate between each tree species even in nonlinear patterns, so the classification results were higher in the nonlinear experiments.

We acknowledge that attaining 100 % accuracy in the training and testing datasets may not signify a robust model. Indeed, surpassing training accuracy with testing accuracy could hint at an unfavourable scenario of overfitting. Researchers must remain vigilant in discerning and mitigating these challenges, as they are diligently addressed within this research to ensure the cultivation of a resilient and dependable model.

Upon examining the RF accuracy as presented in Table 1, it becomes apparent that under standard circumstances, the algorithm may exhibit signs of overfitting the data, as evidenced by achieving 100 % training accuracy. This phenomenon may arise from two potential factors: a limited dataset with fewer data instances and features for the training process or repetitive training and testing solely on the same dataset. Moreover, it is crucial to recognise that such a model may not effectively translate to real-time applications or serve its intended purposes optimally.

Insufficient dataset size or overly simplistic model architectures may lead to overfitting, where the model fails to generalise effectively. Another form of overfitting arises when the testing set accuracy exceeds the training set accuracy and is often overlooked by researchers, posing a significant challenge in machine learning endeavors. To address this, it is necessary to increase model complexity, which involves augmenting feature richness and data volume while maintaining data integrity by minimizing noise. Cross-validation and data augmentation are the best methods for addressing overfitting concerns. In cross-validation, the dataset is divided into k groups, with each iteration utilising one group as the testing set and the remaining as the training set. This iterative process ensures comprehensive evaluation across all folds, enhancing model generalisability and robustness. In the context of our study, repeated stratified K-fold cross-validation involves iteratively repeating the cross-validation process k times and subsequently reporting the mean performance across all folds.

4.5. *Challenges: Algorithm selection*

The research also endeavors to pinpoint the most suitable algorithms for the classification task at hand. Traditionally, researchers often consult algorithms utilised in diverse domains and adopt their corresponding parameter values. However, in most cases, this approach will not guarantee better accuracy because they tune those algorithms on the basis of their specific dataset or purposes. Additionally, it is essential to note that machine learning algorithms do not provide standard or better accuracy for every dataset. Therefore, it is necessary to identify appropriate parameters, and algorithms tailored to a specific task become imperative. In our research, parameter-tuning methodologies are employed to accomplish this aim, with our preference being the grid search approach for parameter fine-tuning. This process enables us to optimise the utilisation of algorithms effectively, thereby yielding improved classification outcomes on our datasets.

5. Conclusion

This study answers two key research questions: why do researchers often focus on segment-specific portions of bark images for tree species classification, and why are deep neural networks widely used for this task?. We used classical machine learning and neural network algorithms on various datasets (normal, normal cropped, exact cropped, nonlinear and combination datasets) to address these research questions.

Initially, classical machine learning algorithms with pre-defined parameters were employed. However, these parameters often prove insufficient for optimal performance. The study also explored the significance of feature scaling, revealing that scaling significantly enhances the effectiveness of classical machine learning algorithms. The grid search methodology was used to investigate optimal parameter values for classical algorithms, but the results were less promising.

Furthermore, neural networks, particularly CNNs, were also examined. The CNNs demonstrated robust performance, with the dataset structure posing minimal difficulties and achieving an average accuracy of approximately 90 % across all datasets when dropout regularisation was applied. Nonlinear deformation was implemented to evaluate the reliability of the algorithms, particularly classical machine learning algorithms, and to determine optimal parameter values. Despite these efforts, the performance on Nonlinear datasets was less promising than that on exact cropped (segment-specific) datasets. However, CNNs applied to Nonlinear datasets exhibited superior performance, consistently maintaining an average accuracy of 85 %.

Only the exact cropped dataset showed potential across both classical machine learning and neural networks, underscoring the importance of segment-specific datasets in classification tasks. Compared with other algorithms, CNNs display superior adaptability across diverse datasets. However, meticulous parameter tuning tailored to specific datasets is crucial for achieving optimal outcomes, which presents a complex challenge for interdisciplinary researchers. To address this challenge, we developed a research software, CNN Parameter Tuner 1.0, to facilitate efficient parameter tuning for CNNs. Moreover, we have made all the scripts used in this research available on GitHub (Gokul, T. C. R *Detecting forest threats with Artificial Intelligence – AZO – Space of Innovation,* 2023) and the main software accessible through Zenodo (Kottilapurath Surendran and Mokros, 2024).

Our findings indicate that deep learning models are most effective for tree species classification via bark images. Segment-specific datasets consistently yield higher accuracies across most algorithms, which explains why researchers have focused on these data formats. Additionally, two interesting questions for future research emerged. First, investigating ways to enhance classical machine learning algorithms, such as by identifying the best feature extractor and employing voting classifier techniques, is essential. Second, the absence of a large European dataset for tree species classification (bark) limits the application of transfer learning approaches. Therefore, we have initiated efforts to increase the accuracy of support vector machines for bark classification and have started collecting a large dataset of bark images from various regions in the Czech Republic. These future works aim to develop more effective classification techniques and potentially automate bark segmentation with large pre-trained models.

Declaration of Generative AI and AI-assisted technologies in the writing process

Statement: During the preparation of this work the author(s) used Curie (Springer Nature) AI improve the readability and language of the work. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

CRediT authorship contribution statement

Gokul Kottilapurath Surendran: Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Deekshitha:** Writing – review & editing. **Martin Lukac:** Writing – review & editing. **Jozef Vybostok:** Data curation. **Martin Mokros:** Supervision.

Acknowledgement

We express our gratitude to IGA-3168 (FLD, CZU) for their generous support in funding this research. The paper was supported by Slovak Research and Development Agency APVV-20-0391 “Monitoring of forest stands in three-dimensional space and time by innovative close-range approaches” and by the Ministry of Education of Slovak Republic grant project VEGA 1/0604/24 “Development of a prototype mobile scanning system for monitoring the condition and evolution of forest ecosystems”. The paper was also supported by the project ReForest (Grant Agreement No. 101060635), funded by the European Union.

The views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

Appendix A. Appendix

Appendix Table 1
CNN parameters and descriptions.

Parameters	Description
Input layer	The input layers represent the image (input) raw pixel values to be analysed.
Convolutional layer	The number of convolutional layers is essential to any neural network. The convolutional layers consist of three main sections. **Filters/Kernels:** Filters are small matrices that slide over the given input image and perform convolution operations (element-wise multiplication and addition). The default kernel size is 3 × 3; even different combinations, like 4 × 4, 2 × 3, and 7 × 7, can provide more accurate results, but not every time. **Stride:** it is the number of pixels which the filter moves across the input matrices. For example, a stride of 2 means the filter moves 2 pixels simultaneously. **Padding:** It is the method of adding zeros around the borders of input matrices to control the spatial size of the output. Commonly, we use two types, one with no padding (‘valid’) and the padding, to ensure the output size is the same as the input size (‘same’). Note: There will be a minimum and a maximum number of layers based on the size and characteristics of the dataset. According to the previous experiments, we determined that the minimum number of neural networks for this dataset is three convolutional layers, including the input layer, the minimum number of layers is two, and the maximum number of layers with dropout for this dataset is 6.
Activation function	Activation functions introduce non-linearity into the models. ReLU (Rectified Linear Unit) is the best activation function for the convolutional layers.
Pooling layer	Pooling is used to reduce the spatial dimensions. **Max Pooling:** It takes the maximum value from a portion of the input matrix. We can decide on the patch size, such as 2 × 2, 3 × 3, etc. **Average Pooling:** Takes average value from a portion of the input matrix.
Fully connected layer (Dense layer)	In the fully connected layer, each neuron is applied a linear transformation into the input vector through the weight matrix. **Weights:** It is the strength of the connection between the neurons learned during the training process. **Biases:** These are the additional parameters learned during training that shift the activation function.
Output layer	This is the final layer in the neural network. It contains two main variables. **Number of Classes:** The total number of categories the network tries to classify from the input images. **Activation function:** It converts the outputs into probabilities for each class. Generally, we use the ‘Softmax function’. The Softmax is used for multi-class classification. However, ‘sigmoid’ is used for binary and multi-label classification tasks where the classes are not mutually exclusive.
Loss function	The loss function measures the difference between the predicted probability and the true. Cross-Entropy Loss is commonly used for classification tasks.
Optimisation	Optimisation algorithms adjust the network parameters, such as weights and biases, during training to minimise the loss function. Adam, Stochastic Gradient Descent (SGD) are examples of optimisation algorithms.
Regularisation	Regularisation techniques are used to prevent overfitting and perform well on unseen data.

(*continued on next page*)

Appendix Table 1 (*continued*)

Parameters	Description
Hyperparameters	**Dropout:** Randomly set a few input units to zero during the training to prevent overfitting. It can decide the efficiency of a network. Better accuracy in higher dropout means the model can perform well on new datasets. **Batch size:** Number of training samples used in one forword/backword pass. The batch size can be defined based on the dataset size; for larger datasets, a larger batch size will be good, and the model will take more time to train. The batch size will define how many samples can pass through each iteration. Furthermore, larger batch sizes also cause irrelevant results in lesser datasets. **Number of Epochs:** It is the number of times the entire training dataset passed through the network. Example 20, 50 epochs.

Appendix Table 2
Hyperparameters for various machine learning algorithms.

Algorithms	All Hyperparameters
Random Forest	max_depth, n_estimators, max_features, n_jobs, min_samples_leaf, min_samples_split, bootstrap, criterion, ccp_alpha, max_leaf_nodes, class_weight, max_samples, min_impurity_decrease, min_weight_fraction_leaf, oob_score, random_state, verbose, warm_start
Decision Tree	ccp_alpha, class_weight, criterion, max_depth, max_features, max_leaf_nodes, min_impurity_decrease, min_samples_leaf, min_samples_split, min_weight_fraction_leaf, random_state, splitter
Support Vector Machine	c, break_ties, cache_size, class_weight, coef0, decision_function_shape, degree, gamma, kernel, max_iter, probability, random_state, shrinking, tol, verbose
Gradient Boosting	ccp_alpha, criterion, init, learning_rate, loss, max_depth, max_features, max_leaf_nodes, min_impurity_decrease, min_impurity, split, min_weight_fraction_leaf, n_estimators, n_iter_no_change, random_state, subsample, tol, validation_fraction, verbose, warm_start
k-Nearest Neighbors	algorithm, leaf_size, metric, metric_params, n_jobs, n_neighbors, p, weights
Gaussian Naïve Bayes	priors, var_smoothing
Multi-layer Perceptron	Activation, alpha, batch_size, beta_1, beta_2, early_stopping, epsilon, hidden_layer_sizes, learning_rate, learning_rate_init, max_fun, max_iter, momentum, n_iter_no_change, nesterovs_momentum, power_t, random_state, shuffle, solver, tol, validation_fraction, verbose, warm_start
Convolutional Neural Networks	Batch_size, pool_type, conv_activation, epochs, dropout_rate

Appendix Table 3
Evaluation Metrics of MLP on the Slovak exact cropped and normal cropped data. This table compares the accuracy, precision, recall, and F1-score of the MLP model before and after applying scaling. The results indicate a significant improvement in performance with scaling, particularly on the exact cropped dataset, where test accuracy increased from 70 % to 92 %.

Dataset	Accuracy					Scaling Accuracy		
	Test	Precision	Recall	F1-score	Test	Precision	Recall	F1-score
Exact Cropped	0.70	0.70	0.67	0.67	0.92	0.92	0.91	0.91
Normal Cropped	0.80	0.81	0.81	0.81	0.78	0.79	0.79	0.79

Appendix Table 4
Performance of CNN on Slovak exact and normal cropped datasets: This table shows the accuracy, precision, recall, and F1-score for various batch sizes, epochs, and number of layers applied to the exact and normal cropped versions of the Slovak dataset. The results demonstrate strong performance, particularly with higher epochs and layers, where accuracy reached up to 98 %.

Dataset	Parameters				Accuracy		
	Batch	Epochs	No. of Layers	Test	Precision	Recall	F1-score
Exact Cropped	12	40	5	0.92	0.92	0.91	0.91
	12	40	3	0.91	0.91	0.90	0.90
	40	65	3	0.98	0.99	0.98	0.98
Normal Cropped	12	40	5	0.90	0.90	0.89	0.89

Appendix Table 5
Grid search metrics: This table presents additional accuracy metrices for the Gradient Boosting (GB) and Random Forest (RF) algorithms.

Algorithm	Accuracy			
	Test	Precision	Recall	F1-score
RF	0.85	0.86	0.85	0.85
GB	0.83	0.85	0.80	0.81

Appendix Table 6
Results of the CZU CNN experiment (40 epochs with scaling): This table presents the precision, recall, and F1-score for various tree species classifications. Notably, the large-leaved linden showed significant misclassification, indicating challenges in accurately identifying this species compared to others, such as the European beech and Scots pine, which demonstrated excellent precision and recall scores.

	Precision	Recall	F1-score
European beech	1.00	0.84	0.91
Large-leaved linden	0.43	0.27	0.33
Norway maple	0.73	0.90	0.81
Scots pine	1.00	0.88	0.94
		0.81	

Appendix Table 7
Class-wise precision, recall, and F1-score for the SVM model without scaling: The model performs well for certain classes such as Norway maple, but fails completely in distinguishing Large-leaved linden and Scots pine, resulting in a precision, recall, and F1-score of 0.00 for these categories. Moreover, the performance for European beech is also poor, with an F1-score of only 0.20. The overall performance reflects the model's difficulty in handling specific tree species in this two-class classification task.

	Precision	Recall	F1-score
European beech	0.20	0.20	0.20
Large-leaved linden	0.00	0.00	0.00
Norway maple	0.48	0.85	0.62
Scots pine	0.00	0.00	0.00
		0.43	

Appendix Table 8
Accuracy metrics for Multi-Layer Perceptron (MLP) on nonlinear data, with and without scaling: The table illustrates a significant improvement in performance when scaling is applied. Without scaling, the MLP achieves an accuracy of just 16 %, with low precision, recall, and F1-score values. However, after applying scaling, the test accuracy jumps to 78 %, and precision, recall, and F1-scores also rise notably, indicating that scaling plays a crucial role in enhancing the MLP's performance on this dataset.

Accuracy					Scaling Accuracy		
Test	Precision	Recall	F1-score	Test	Precision	Recall	F1-score
0.16	0.10	0.17	0.12	0.78	0.81	0.80	0.78

Appendix Table 9
Class-wise precision, recall, and F1-score for the CNN algorithm after 65 epochs on the nonlinear dataset. The results demonstrate strong overall performance, with an accuracy of 89 %. European beech achieves near-perfect classification with an F1-score of 0.98, while Sessile oak also performs well with a recall of 0.93 and an F1-score of 0.88. European silver fir and Norway spruce exhibit slightly lower precision and recall but still maintain solid F1-scores of 0.88 and 0.83, respectively. These results indicate that the CNN model effectively distinguishes between different classes, particularly for European beech.

	Precision	Recall	F1-score
European beech	1.00	0.95	0.98
European silver fir	0.86	0.90	0.88
Norway spruce	0.87	0.80	0.83
Sessile oak	0.82	0.93	0.88
accuracy		0.89	

Data availability

The data, scripts, and software utilised in this work are openly available for public access.

Datasets: https://zenodo.org/communities/forestry_investigation_datasets/records

Scripts: https://github.com/Gokultcr/A-Forestry-Investigation-Scripts

Software: https://zenodo.org/records/13881151

References

Abdali, E., Valadan Zoej, M.J., Taheri Dehkordi, A., Ghaderpour, E., 2024. A parallel-cascaded Ensemble of Machine Learning Models for crop type classification in Google earth engine using multi-temporal Sentinel-1/2 and Landsat-8/9 remote sensing data. Remote Sens. 16 (1), 127. https://doi.org/10.3390/rs16010127.

Aguiar, A.S., Dos Santos, F.N., De Sousa, A.J.M., Oliveira, P.M., Santos, L.C., 2020. Visual trunk detection using transfer learning and a deep learning-based coprocessor. IEEE Access 8, 77308–77320. https://doi.org/10.1109/ACCESS.2020.2989052.

Ameur, R. Ben, Valet, L., Coquin, D., 2016. Sub-classification strategies for tree species recognition. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 2139–2144. https://doi.org/10.1109/ICPR.2016.7899952.

Aszalós, R., Thom, D., Aakala, T., Angelstam, P., Brūmelis, G., Gálhidy, L., Gratzer, G., Hlásny, T., Katzensteiner, K., Kovács, B., Knoke, T., Larrieu, L., Motta, R., Müller, J., Ódor, P., Roženbergar, D., Paillet, Y., Pitar, D., Standovár, T., Keeton, W.S., 2022. Natural disturbance regimes as a guide for sustainable forest management in Europe. Ecol. Appl. 32 (5), e2596. https://doi.org/10.1002/eap.2596.

Barré, P., Stöver, B.C., Müller, K.F., Steinhage, V., 2017. LeafNet: A computer vision system for automatic plant species identification. Eco. Inform. 40, 50–56. https://doi.org/10.1016/j.ecoinf.2017.05.005.

Benassi, A., Kardous, F., Grayaa, K., 2024. Almond tree variety identification based on bark photographs using deep learning approach and wavelet transform. Arab. J. Sci. Eng. 49, 12525–12535. https://doi.org/10.1007/s13369-024-08743-x.

Bentéjac, C., Csörgő, A., Martínez-Muñoz, G., 2021. A comparative analysis of gradient boosting algorithms. Artif. Intell. Rev. 54 (3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5.

Bertrand, S., Ben Ameur, R., Cerutti, G., Coquin, D., Valet, L., Tougne, L., 2018. Bark and leaf fusion systems to improve automatic tree species recognition. Eco. Inform. 46, 57–73. https://doi.org/10.1016/j.ecoinf.2018.05.007.

Blaanco, L.J., Travieso, C.M., Quinteiro, J.M., Hernandez, P.V., Dutta, M.K., Singh, A., 2016. A bark recognition algorithm for plant classification using a least square support vector machine. In: 2016 Ninth International Conference on Contemporary Computing (IC3), pp. 1–5. https://doi.org/10.1109/IC3.2016.7880233.

Bolyn, C., Lejeune, P., Michez, A., Latte, N., 2022. Mapping tree species proportions from satellite imagery using spectral–spatial deep learning. Remote Sens. Environ. 280, 113205. https://doi.org/10.1016/j.rse.2022.113205.

Boudra, S., Yahiaoui, I., Behloul, A., 2021. A set of statistical radial binary patterns for tree species identification based on bark images. Multimed. Tools Appl. 80 (15), 22373–22404. https://doi.org/10.1007/s11042-020-08874-x.

Boudra, S., Yahiaoui, I., Behloul, A., 2022. Tree trunk texture classification using multi-scale statistical macro binary patterns and CNN. Appl. Soft Comput. 118, 108473. https://doi.org/10.1016/j.asoc.2022.108473.

Bressane, A., Roveda, J.A.F., Martins, A.C.G., 2015. Pattern recognition in trunk images based on co-occurrence descriptors: A proposal applied to tree species identification. In: 2015 Latin America Congress on Computational Intelligence (LA-CCI), pp. 1–6. https://doi.org/10.1109/LA-CCI.2015.7435983.

Carpentier, M., Giguere, P., Gaudreault, J., 2018. Tree species identification from bark images using convolutional neural networks. In: IEEE International Conference on Intelligent Robots and Systems, pp. 1075–1081. https://doi.org/10.1109/IROS.2018.8593514.

Chaki, J., Dey, N., Moraru, L., Shi, F., 2019. Fragmented plant leaf recognition: bag-of-features, fuzzy-color and edge-texture histogram descriptors with multi-layer perceptron. Optik 181, 639–650. https://doi.org/10.1016/J.IJLEO.2018.12.107.

Chen, X., Wang, S., Zhang, B., Luo, L., 2018. Multi-feature fusion tree trunk detection and orchard mobile robot localization using camera/ultrasonic sensors. Comput. Electron. Agric. 147, 91–108. https://doi.org/10.1016/j.compag.2018.02.009.

Cui, Z., Li, X., Li, T., Li, M., 2023. Improvement and assessment of convolutional neural network for tree species identification based on bark characteristics. Forests 14 (7), 1292. https://doi.org/10.3390/f14071292.

da Silva, D.Q., dos Santos, F.N., Sousa, A.J., Filipe, V., 2021. Visible and thermal image-based trunk detection with deep learning for forestry Mobile robotics. J. Imag. 7 (9), 176. https://doi.org/10.3390/jimaging7090176.

da Silva, S.D.P., Eugenio, F.C., Fantinel, R.A., de Amaral, L.P., dos Santos, A.R., Mallmann, C.L., dos Santos, F.D., Pereira, R.S., Ruoso, R., 2023. Modeling and detection of invasive trees using UAV image and machine learning in a subtropical forest in Brazil. Eco. Inform. 74, 101989. https://doi.org/10.1016/j.ecoinf.2023.101989.

Debaleena, Misra, Crispim-Junior, C., T. L, 2020. Patch-based CNN evaluation for bark classification. In: Bartoli, A., Fusiello, Adrien (Eds.), Computer Vision – ECCV 2020 Workshops. Springer International Publishing, pp. 197–212. https://doi.org/10.1007/978-3-030-65414-6_15.

Detecting forest threats with Artificial Intelligence – AZO – Space of Innovation. Retrieved August 10, 2023, from. https://space-of-innovation.com/detecting-forest-threats-with-ai/.

Fekri-Ershad, S., 2020. Bark texture classification using improved local ternary patterns and multilayer neural network. Expert Syst. Appl. 158, 113509. https://doi.org/10.1016/j.eswa.2020.113509.

Fiel, S., Sablatnig, R., 2011. Automated identification of tree species from images of the bark, leaves or needles. In: Proceedings of the 16th Computer Vision Winter Workshop, pp. 67–74.

Fürnkranz, J., 2010. Decision tree. In: Sammut, G.I., Webb, Claude (Eds.), Encyclopedia of Machine Learning. Springer US, pp. 263–267. https://doi.org/10.1007/978-0-387-30164-8_204.

Ganschow, L., Thiele, T., Deckers, N., Reulke, R., 2019. Classification of tree species on the basis of tree bark texture. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 42(2/W13), pp. 1855–1859. https://doi.org/10.5194/isprs-archives-XLII-2-W13-1855-2019.

Haralick, R.M., 1979. Statistical and structural approaches to texture. Proc. IEEE 67 (5), 786–804. https://doi.org/10.1109/PROC.1979.11328.

Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. IEEE Trans. Syst. Man Cybern. SMC-3 (6), 610–621. https://doi.org/10.1109/TSMC.1973.4309314.

Hastie, T., 2008. Estimating the error rate of a prediction rule: Improvement on cross-validation. In: Morris, R., Tibshirani, Carl N. (Eds.), The Science of Bradley Efron: Selected Papers. Springer, New York, pp. 240–259. https://doi.org/10.1007/978-0-387-75692-9_12.

Homan, D., du Preez, J.A., 2021. Automated feature-specific tree species identification from natural images using deep semi-supervised learning. Eco. Inform. 66, 101475. https://doi.org/10.1016/j.ecoinf.2021.101475.

Hu, G., Yao, P., Wan, M., Bao, W., Zeng, W., 2022. Detection and classification of diseased pine trees with different levels of severity from UAV remote sensing images. Eco. Inform. 72, 101844. https://doi.org/10.1016/j.ecoinf.2022.101844.

Jendoubi, S., Coquin, D., Boukezzoula, R., 2020. Evidential two-step tree species recognition approach from leaves and bark. Expert Syst. Appl. 146, 113154. https://doi.org/10.1016/j.eswa.2019.113154.

Kanda, P.S., Xia, K., Sanusi, O.H., 2021. A Deep Learning-Based Recognition Technique for Plant Leaf Classification. IEEE Access, p. 1. https://api.semanticscholar.org/CorpusID:245146839.

Kim, T.K., Hong, J., Ryu, D., Kim, S., Byeon, S.Y., Huh, W., Kim, K., Baek, G.H., Kim, H.S., 2022. Identifying and extracting bark key features of 42 tree species using convolutional neural networks and class activation mapping. Sci. Rep. 12 (1), 4772. https://doi.org/10.1038/s41598-022-08571-9.

Kottilapurath Surendran, G., Mokros, M., 2024. CNN Parameter Tuner 1.0. Zenodo. https://doi.org/10.5281/zenodo.12601079.

Krita | Digital Painting, 2024. Creative Freedom. Retrieved February 13, 2024, from. https://krita.org/en/.

Li, R., Sun, G.D., Wang, S., Tan, T.Z., Xu, F., 2023. Tree trunk detection in urban scenes using a multiscale attention-based deep learning method. Eco. Inform. 77, 102215. https://doi.org/10.1016/j.ecoinf.2023.102215.

Liang, X., Hyyppä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., Holopainen, M., Brolly, G., Francesco, P., Hackenberg, J., Huang, H., Jo, H.W., Katoh, M., Liu, L., Mokroš, M., Morel, J., Olofsson, K., Poveda-Lopez, J., Trochta, J., Wang, Y., 2018. International benchmarking of terrestrial laser scanning approaches for forest inventories. ISPRS J. Photogramm. Remote Sens. 144, 137–179. https://doi.org/10.1016/j.isprsjprs.2018.06.021.

Louppe, G., 2014. Understanding Random Forests: From Theory to Practice. https://arxiv.org/abs/1407.7502v3.

Minowa, Y., Kubota, Y., Nakatsukasa, S., 2022. Verification of a deep learning-based tree species identification model using images of broadleaf and coniferous tree leaves. Forests 13 (6), 943. https://doi.org/10.3390/f13060943.

Munisami, T., Ramsurn, M., Kishnah, S., Pudaruth, S., 2015. Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers. Procedia Comp. Sci. 58, 740–747. https://doi.org/10.1016/j.procs.2015.08.095.

Nikolić, G., Vujović, F., Golijanin, J., Šiljeg, A., Valjarević, A., 2023. Modelling of wildfire susceptibility in different climate zones in Montenegro using GIS-MCDA. Atmosphere 14 (6), 929. https://doi.org/10.3390/atmos14060929.

O'Shea, K., Nash, R., 2015. An introduction to convolutional neural networks. Int. J. Res. Appl. Sci. Eng. Technol. 10 (12), 943–947. https://doi.org/10.22214/ijraset.2022.47789.

Pereira, Dulce G., A. A, Medeiros, F.M., 2015. Overview of Friedman's test and post-hoc analysis. Commun. Stat. Simul. Comp. 44 (10), 2636–2653. https://doi.org/10.1080/03610918.2014.931971.

Powers, D., Ailab., 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol. 2, 2229–3981. https://doi.org/10.9735/2229-3981.

Pushpa, B.R., Rani, N.S., Chandrajith, M., Manohar, N., Nair, S.S.K., 2024. On the importance of integrating convolution features for Indian medicinal plant species classification using hierarchical machine learning approach. Eco. Inform. 81, 102611. https://doi.org/10.1016/j.ecoinf.2024.102611.

Remeš, V., Haindl, M., 2019. Bark recognition using novel rotationally invariant multispectral textural features. Pattern Recogn. Lett. 125, 612–617. https://doi.org/10.1016/j.patrec.2019.06.027.

Robert, M., Dallaire, P., Giguère, P., 2020. Tree bark re-identification using a deep-learning feature descriptor. In: 2020 17th Conference on Computer and Robot Vision (CRV), pp. 25–32. https://doi.org/10.1109/CRV50864.2020.00012.

Rudolf, Kruse, Mostaghim, S., B. C. and B. C. and S. M, 2022. Multi-layer Perceptrons. In: Computational Intelligence: A Methodological Introduction. Springer International Publishing, pp. 53–124. https://doi.org/10.1007/978-3-030-42227-1_5.

Salavati, G., Saniei, E., Ghaderpour, E., Hassan, Q.K., 2022. Wildfire risk forecasting using weights of evidence and statistical index models. Sustainability (Switzerland) 14 (7), 3881. https://doi.org/10.3390/su14073881.

Scikit-image: Image Processing in Python — Scikit-image. Retrieved February 13, 2024, from. https://scikit-image.org/.

StandardScaler — scikit-learn 1.5.0 Documentation. Retrieved June 4, 2024, from. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

Sulc, M., Matas, J., 2013. Kernel-mapped histograms of multi-scale LBPs for tree bark recognition. In: 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), pp. 82–87. https://doi.org/10.1109/IVCNZ.2013.6726996.

Sun, X., Shi, Y., 2023. The image recognition of urban greening tree species based on deep learning and CAMP-MKNet model. Urban For. Urban Green. 85, 127970. https://doi.org/10.1016/j.ufug.2023.127970.

Syriopoulos, P.K., Kalampalikis, N.G., Kotsiantis, S.B., Vrahatis, M.N., 2023. kNN classification: a review. Ann. Math. Artif. Intell. 98. https://doi.org/10.1007/s10472-023-09882-x.

TRUNK12, 2022. Tree Bark Image Data Set | ViCoS Lab. https://www.vicos.si/resources/trunk12/.

Uriarte, M., Canham, C.D., Thompson, J., Zimmerman, J.K., Murphy, L., Sabat, A.M., Fetcher, N., Haines, B.L., 2009. Natural disturbance and human land use as determinants of tropical forest dynamics: results from a forest simulator. Ecol. Monogr. 79 (3), 423–443. https://doi.org/10.1890/08-0707.1.

Veras, H.F.P., Ferreira, M.P., da Cunha Neto, E.M., Figueiredo, E.O., Corte, A.P.D., Sanquetta, C.R., 2022. Fusing multi-season UAS images with convolutional neural networks to map tree species in Amazonian forests. Eco. Inform. 71, 101815. https://doi.org/10.1016/j.ecoinf.2022.101815.

Vizcarra, G., Bermejo, D., Mauricio, A., Zarate Gomez, R., Dianderas, E., 2021. The Peruvian Amazon forestry dataset: A leaf image classification corpus. Eco. Inform. 62, 101268. https://doi.org/10.1016/j.ecoinf.2021.101268.

Wojtkowska, M., Kedzierski, M., Delis, P., 2021. Validation of terrestrial laser scanning and artificial intelligence for measuring deformations of cultural heritage structures. Measurem. J. Int. Measurem. Confed. 167, 108291. https://doi.org/10.1016/j.measurement.2020.108291.

Wu, F., Gazo, R., Benes, B., Haviarova, E., 2021. Deep BarkID: a portable tree bark identification system by knowledge distillation. Eur. J. For. Res. 140 (6), 1391–1399. https://doi.org/10.1007/s10342-021-01407-7.

Zhang, H., 2004. The Optimality of Naive Bayes. Retrieved June 4, 2024, from. www.aaai.org.

Zhang, Y., Liu, L.Y.A., Wang, Chunfeng, 2012. Support vector machine classification algorithm and its application. In: *Information Computing and Applications*, 308. Springer, Berlin Heidelberg, pp. 179–186. https://doi.org/10.1007/978-3-642-34041-3_27.

Zhao, Y., Gao, X., Hu, J., Chen, Z., Chen, Z., 2020. Tree species identification based on the fusion of bark and leaves. Math. Biosci. Eng. 17 (4), 4018–4033. https://doi.org/10.3934/MBE.2020222.

Zhi-Kai, Huang, Huang, D.-S., D. J.-X, Q. Z.-H, G. S.-B, 2006. Bark classification based on Gabor filter features using RBPNN neural network. In: J. C. L.-W, King, W.D., Wang, Irwin (Eds.), Neural Information Processing. Springer, Berlin Heidelberg, pp. 80–87.

Zhou, H., Yan, C., Huang, H., 2016. Tree species identification based on convolutional neural networks. In: Proceedings - 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2016, 2, pp. 103–106. https://doi.org/10.1109/IHMSC.2016.144.