Using Hybrid Models of AI for Identification of Trees by UAV Images of Forests: I. Machine-learning Component of the Models

ZURAB BOSIKASHVILI[1], GIORGI KVARTSKHAVA[2]
[1]Faculty of Informatics and Control Systems,
Georgian Technical University,
77, Kostava Str., Tbilisi, 0160,
GEORGIA

[2]Faculty of Agricultural Science and Biosystems Engineering,
Georgian Technical University,
77, Kostava Str., Tbilisi, 0160,
GEORGIA

Abstract: - Artificial intellect models (machine learning, logical reasoning, etc.) are currently the focus of many remote sensing approaches for forest inventory management. Although they return satisfactory results in many tasks, some challenges remain, especially in the case of the highly dense distribution of trees in forests. In this paper, we propose a novel hybrid approach using together deep learning models and symbolic logic methods for identifying single-tree species in highly dense areas. The use of deep learning methods in solving high dimensional problems in face recognition has some **issues** due to low accuracy and interpretability of results. The paper proposes a hybrid approach for solving complex image classification problems. This approach involves the use of both machine learning methods and symbolic knowledge. The paper presents the structure and formal model of the hybrid system, which includes a new component, an operations manager. The first part of the paper proposes a new architecture of deep neural networks with attentional mechanisms built on blocking meta-functions. The corresponding module has been developed in Python language. The results of the module's work are provided to the knowledge base. As a result of symbolic conclusions, the teaching module is reorganized. The experiments conducted showed the effectiveness of the presented approach.

Key-Words: - deep learning; hybrid model; knowledge base; convolutional neural network; blocking function; sliding window; internal and external horizon; tree species identification.

Received: March 15, 2023. Revised: March 17, 2024. Accepted: June 2, 2024. Published: July 4, 2024.

1 Introduction

Forest inventory processing is one of the most important and time-consuming parts of environmental protection issues. Forest protection tasks are particularly challenging for each country whose territory is covered by forest. Around 40% of Georgia is covered by forest (2.8 million ha). Most of the forest is mountain forest and only 2% is lowland forest. It is much more difficult if forests are natural and many species of trees and shrubs grow in different densities, especially if it is problematic in the case of the highly dense distribution of trees in forests. Full and precise inventory and automatization of such a kind of forest is very actual.

The first step is to develop a monitoring system that allows a rapid and reliable method to survey this type of forest and additionally to be able to identify tree species. The use of aerial imagery in combination with Artificial intelligence (AI) and machinery learning (ML) approaches are essential tools and techniques that can build the basis for the improvement of monitoring methodologies in forestry research, [1], [2].

In solving the mentioned problems, it is important to separate the following two tasks:

- Selection of data collection methods and tools.

- Identification of trees with the obtained data (identification of species and characteristics).

The first issue was analyzed in detail in the work, [3], [4], [5]. It was mentioned that many studies have been conducted using remote sensing data. So far, remote sensing research with these objectives has mainly employed satellites or aircraft. In the past, much attention has been devoted to multispectral Landsat satellites, which, because of their low cost, enable the coverage of vast areas, i.e. on a country scale. However, its resolution is 30 m, which does not allow easy identification of tree species. Since 2000, many studies have used very high-resolution data for tree species classification from commercial satellites with resolutions of 0.5 m/2.0 m and 0.6 m/2.4 m for panchromatic/multispectral data, respectively. Further, in recent years, studies using aircraft have succeeded in identifying several tree species. The spatial resolution of images used therein is also very high: approximately 0.2–3.0 m. Most of these studies used specialized hardware such as multispectral, hyperspectral, and LiDAR sensors, and achieved high performance in identifying tree species with 80%- 85% accuracy, but the equipment involved therein is highly expensive. These approaches utilizing multi/hyperspectral data have often experienced problems. One of these involves the spectral features, which can differ not only between species, but also across densities of leaves, health conditions, and background noises such as understory vegetation or bare soil. When there is a shadow, the spectrum of the shadow differs from that of the no-shadow area, resulting in a lower accuracy. In identifying a mixed forest, the performance might be lower because multiple species are included in one pixel. In recent decades, unmanned aerial vehicles (UAVs) have been used experimentally in forestry applications. Compared to manned aircraft, UAVs are easy-to-use, low-cost tools for remote sensing of forests. Regarding tree identification, the most important difference between manned aircraft and UAVs is that UAVs can fly near canopies and acquire extremely high-resolution images; the images from UAVs have a spatial resolution. That is why in this paper we use UAV forest images for tree identification.

In general, the problem of tree identification by UAV images of forests is an object detection problem from remote sensing data, that has been studied well. There are many scientific papers on these issues, [3], [4].

Several scientific directions are geared toward detecting objects of interest in remote sensing images, for further regional analysis and classification. These algorithms are categorized into three groups:

a. template matching-based,
b. knowledge-based,
c. machine learning-based

methods. A detailed review and analysis of these approaches for detecting trees from forest images and classifying them are given in [5]. Let's briefly review the last two approaches.

1.1 Main Challenges and Goals

In modern artificial intelligence (AI) systems, these skills are typically realized through machine learning and symbolic inference mechanisms. Machine learning technologies, such as deep neural networks, have made significant strides in solving perceptual problems; Meanwhile, logic-based AI systems have made significant strides in human-level reasoning capabilities. But these two parts developed for the most part independently of each other, which gives rise to the following problems:

- The scarcity of expressiveness of the results of training with deep neural networks in humans leads to a decrease in trust in the results, even when the results are obtained with high accuracy.
- Most methods of making logical inferences work with low accuracy with incomplete and inaccurate data.
- Although most such systems provide information and knowledge acquisition and generalization processes, they are poorly developed or do not have adaptation and self-development mechanisms at all, making such systems difficult to use.
- The main problem is that the set task has many stages and the mistake made at the initial stage is revealed only at the last stage. Systems created with these approaches are not adaptive, and the elimination of errors is done only by humans at the end.

However, AI methods and systems can play significant roles in solving the identification of tree tasks, but unfortunately, there is no existing perfect model that can fully perform all tasks or several together.

The main goal of this work is to present the hybrid machine learning (HML) framework and demonstrate the effectiveness of its use for solving a specific face recognition problem based on union deep neural networks of machine learning and symbolic reasoning mechanisms, which will carry out abduction learning. To fulfill this task, we established a new architecture of deep neural network and logic reasoning system for canopy segmentation and identification of tree species.

2 Problem Formulation

The integration of machine learning with logical reasoning is still an open research problem, [6], [7], [8], and therefore even a small step in this direction is considered an innovation in the creation of intelligent systems.

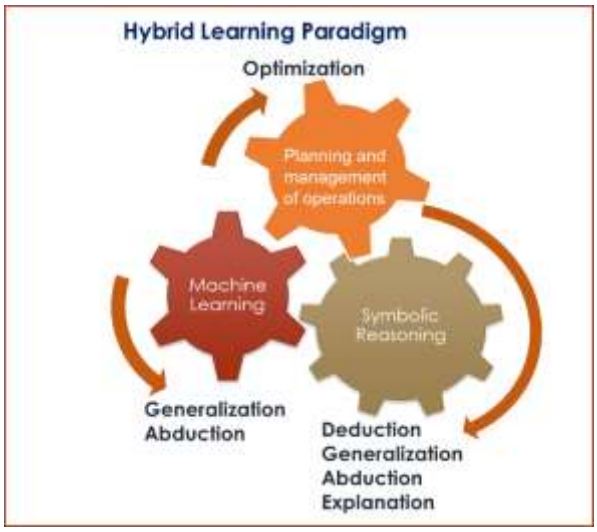


Fig. 1: Hybrid machine-learning paradigm

Significantly, the proposed hybrid machine learning paradigm also plays a crucial role in the data acquisition, enrichment, configuration, management, and optimization processes (Figure 1). According to the paradigm, based on a priori numerical data, machine learning is used to generalize data in the form of symbolic descriptions of concepts, on which further conclusions are made to solve specific tasks. In addition, Reinforce Learning mechanisms optimize learning and inference processes, giving the system self-development and adaptation skills.

2.1 Proposed Methodological Framework

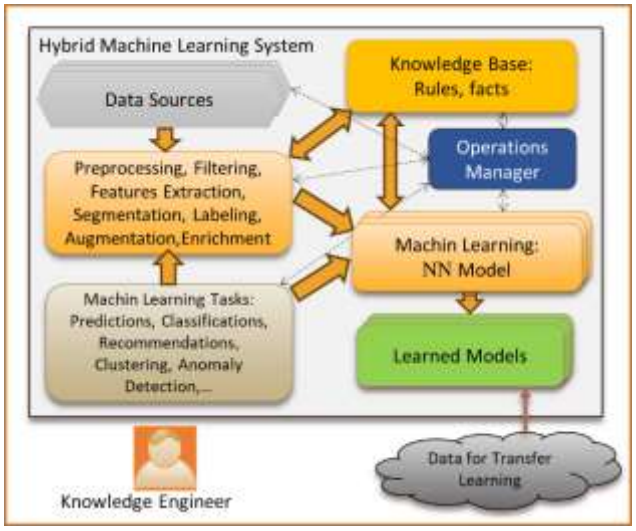


Fig. 2: A multi-layer model of the hybrid machine learning process

In general, machine learning is a multi-step process. In the case of HML, this process becomes even more complicated and two important new components are added (Figure 2).

A crucial aspect of the system is the machine learning module, which comprises numerous neural network models for diverse purposes such as classification, segmentation, clustering, enrichment, etc. The models write their performance outcomes to the knowledge base, and contextual information is also received from the knowledge base. The system further consists of a knowledge base that stores facts after every step. An operations manager ensures efficient system operation and manages the learning process and conclusions. Additionally, system hyper-parameters are not directly changed during the model training, but the operations manager utilizes the whole system in the training. The system learning settings can enhance the system's performance and can be applied to any component. The formal model of HML takes the following form:

$$\boldsymbol{H} = \langle \widehat{\boldsymbol{D}}, \widehat{\boldsymbol{M}}, \widehat{\boldsymbol{U}}, \widehat{\boldsymbol{T}}, \widehat{\boldsymbol{G}}, \boldsymbol{K_B}, \boldsymbol{C_O} \rangle, \quad (1)$$

where $\widehat{\boldsymbol{D}} = \{\boldsymbol{D1_1}, \boldsymbol{D_2}, .., \boldsymbol{D_k}\}$ - set of data sources, $\widehat{\boldsymbol{M}} = \{\boldsymbol{M_1}, \boldsymbol{M_2}, \ldots, \boldsymbol{M_m}\}$ -set of machine learning models and $\boldsymbol{M_i}$ is an operator representing a model of neural networks, $\boldsymbol{M_i}: \boldsymbol{E_i} \times \boldsymbol{W_i} \times \boldsymbol{\Theta_i} \rightarrow \boldsymbol{Z_i}$, where $\boldsymbol{Ei} = \boldsymbol{Dom(M_i)}$ - is an input collection of training or testing data set, $\boldsymbol{W_i} \subseteq \mathbb{R}^{\boldsymbol{ki}}$ and $\boldsymbol{w} \in \boldsymbol{W_i}$ -is the vector of weights of NN, $\boldsymbol{\Theta_i} \subset \mathbb{R}^{\boldsymbol{v}}$ -is set of

systems parameters. $\boldsymbol{Z_i}$- output collection of $\boldsymbol{M_i}$ and $\forall\, \boldsymbol{e}(\boldsymbol{e} \in \boldsymbol{E_i})\, \exists \boldsymbol{w} \exists \boldsymbol{\theta}\ (\boldsymbol{M_i}(\boldsymbol{e}, \boldsymbol{w}, \boldsymbol{\theta}) \in \boldsymbol{Z_i}))$ **i**s true.

$\widehat{\boldsymbol{U}} = \{\boldsymbol{U_1}, \boldsymbol{U_2}, \dots, \boldsymbol{U_n}\}$ - set of operators presenting a system utilities, $\boldsymbol{U_i}: \boldsymbol{X_i} \times \boldsymbol{\Theta_i} \to \boldsymbol{Y_i}$, $\boldsymbol{X_i} = \boldsymbol{Dom}(\boldsymbol{U_i})$, $\boldsymbol{\Theta_i} \subset \mathbb{R}^{\boldsymbol{v}}$ - is a set of systems parameters that can changed only by the system or by an operator and are used during learning. $\forall\, \boldsymbol{x}(\boldsymbol{x} \in \boldsymbol{X_i})\, \exists \boldsymbol{\theta}\ \boldsymbol{U_i}(\boldsymbol{e}, \boldsymbol{\theta}) \in \boldsymbol{Y_i}))$ - this logical expression is true.

$\widehat{\boldsymbol{T}} = \{\boldsymbol{T_1}, \boldsymbol{T_2}, \dots, \boldsymbol{T_l}\}$ - set of ML tasks (prediction, recommendation, anomaly detection, segmentation, labeling…)

$\widehat{\boldsymbol{G}} = \{\boldsymbol{G_1}, \boldsymbol{G_2}, \dots, \boldsymbol{G_l}\}$ - set of tasks evaluation graphs, where $\boldsymbol{G_i} = \boldsymbol{(V_i, L_i)}$, is the evaluation graph of the $\boldsymbol{i}$-th task .$\boldsymbol{V_i}$ is a set of vertices, each of which $v_j \in V_i$ represents either $m \in M^j$ or a machine learning operator, or $u \in U^j$ a system utility, which are respectively selected from the subsets of machine learning models or system utilities associated with this vertex. $\boldsymbol{G_i}$ graph can have 1 or more input vertices and only one output vertex. $\boldsymbol{L_i}$ is a set of edges/links which are ordered pairs of distinct vertices: $\boldsymbol{L_i} \subseteq \{(\boldsymbol{v_p}, \boldsymbol{v_q}) \in \boldsymbol{V_i} \times \boldsymbol{V_i}\ \&\ \boldsymbol{v_p} \neq \boldsymbol{v_q}\}$. Figure 3 shows an example of one graph. In each vertex of the graph, along with the choice of model and utility, their parameters are also selected. Of course, the choice at each vertex is influenced by the choices of its previous vertices. Therefore, for each $\forall v_{ij} \in V_i$ vertex of the given computational graph, define the sets of selectable operators and parameters associated with it, $\{\widetilde{M}^{ij} = Pr_M(v_{ij}),\ \widetilde{U}^{ij} = Pr_U(v_{ij}),\ \widetilde{W}^{ij} = Pr_W(v_{ij}),\ \widetilde{\Theta}^{ij} = Pr_\theta(v_{ij})\}$, where $\widetilde{M}^{ij} \subseteq \widehat{\boldsymbol{M}}$, $\widetilde{U}^{ij} \subseteq \widehat{\boldsymbol{U}}$, $\widetilde{W}^{ij} \subseteq \widehat{\boldsymbol{W}}$, $\widetilde{\boldsymbol{\Theta}}^{ij} \subseteq \widehat{\boldsymbol{\Theta}}$.

$\boldsymbol{K_B} = \langle \boldsymbol{R}, \boldsymbol{\Phi}, \boldsymbol{\Psi} \rangle$ – is a knowledge base with $\boldsymbol{R}$-rules, $\boldsymbol{\Phi}$ -facts, and $\boldsymbol{\Psi}$ –is a module that includes deductive, inductive, and abductive mechanisms of making conclusions, generalization, and explanation mechanisms.

$\boldsymbol{C_O}$ - Operations Manager-Controller includes MS's overall process management model, which consists of performance and integrity optimization as well as solution explanation models. It also has the function of mutual transfer and transformation of information between machine learning modules and knowledge base.

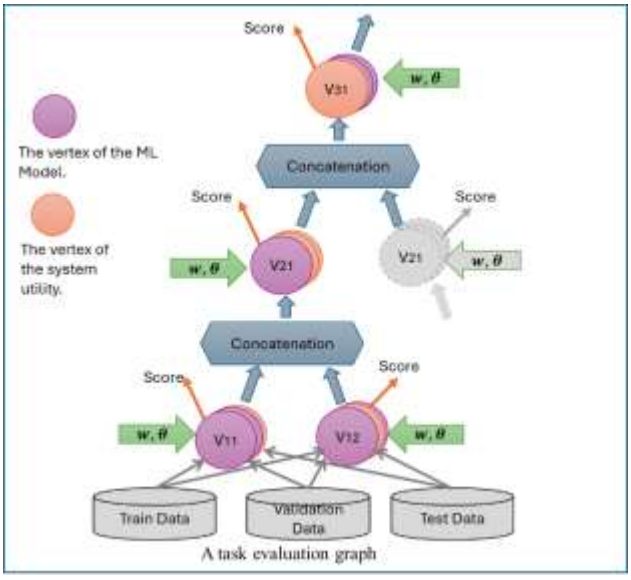


Fig. 3: A task evaluation graph

To plan the work of the system, effectively manage it, and self-study, the operations manager needs certain criteria and measurements to evaluate the performance of his work. In general, there are many approaches: but only the following two are presented in the paper:

1. Performance assessment**.** This involves evaluating key metrics such as:
   - Accuracy
   - Precision
   - Recall
   - F1-score
   - Log loss.
   - Mean Squared Error (MSE)
   - Mean Absolute Error (MAE).
   - Root Mean Squared Error (RMSE)
   - R2
2. Benchmarking of tasks workload and resources evaluation
   - Model benchmark parameters: number of parameters, speed per FLOPs, image size, etc...
   - Training time to reach a target accuracy of 90%.
   - Throughput in terms of images processed per second.

To calculate the presented metrics, the paper introduces the ***Score*** operator, which is called by the operations manager at the output vertex of the computational graph to optimally select the operators

and parameters by finding the extremum of the function given in formula (2).

$$S_i^* = \underset{[m,u,w,\theta]}{\mathbf{argextr}}\left[\bigcup_{i=1,k} Score([m,u,w,\theta])\right] \quad (2)$$

here $[m,u,w,\theta] = \boldsymbol{Gen}(\widetilde{M}^{io}, \widetilde{U}^{io}, \widetilde{W}^{io}, \widetilde{\Theta}^{io})$. ***Gen*** is an operator that generates the selection vectors $\boldsymbol{k}$ time for $\boldsymbol{G_i}$ graph.

At the end of this section, we can summarize that the formal hybrid model has been developed, incorporating machine learning, symbolic reasoning, and system operation control, enabling the right system design and implementation.

3 Problem Solution

During the last decade, researchers from different institutions have been suggesting different methods in the segmentation and classification of plants. The overall accuracy of these methods sometimes reaches ~85%, [9]. Several algorithms for detecting objects of interest in remote sensing images and subsequent classification have been devised, and these include template matching-based methods, machine learning, and knowledge-based methods.

The problem of solving the task of identifying trees according to the forest remote sensing image is prevented by:

I. Large size of forest images (6000x5000 pixels),
II. Complex semantics of the image (in the forest you can find different types of plants that can cover each other).

In the paper, for the simplicity of reasoning, the logical model of the forest will be presented in the form shown in Figure 4. The proposed hybrid approach to solve the image recognition problems involves the following steps:

- Hierarchical representation of the image based on the dimensions of the forest components, such as the forest image, branch image, and leaves image.
- Identification of the types (classes) and relationships between the components based on their images. This can be done using machine learning methods or classical algorithms, such as Watershed or U-Net models. It can also be done by a combination of training sets and benchmarks involving both humans and machines.
- The operation may involve one method with multiple hyperparameters or several methods running in parallel. The operation manager selects the best result, which can be recalculated with alternative parameters or methods.
- The knowledge base receives the results of the previous methods in the form of facts, including the guessed type of the tree based on the relationships between its components (semantics).

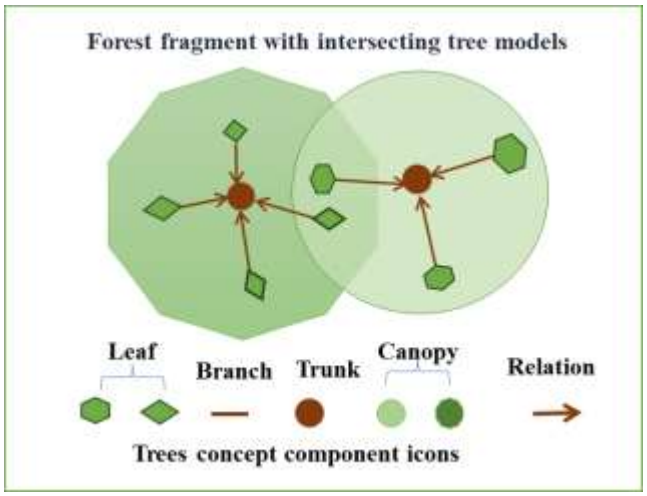


Fig. 4: A logical model of the forest area where trees intersect

Let's consider an example to understand the need for a hybrid model. Suppose we need to classify different types of trees based on location images. Let's assume we have a machine-learning model that classifies tree types based on their leaves. This model is trained on the leaves of two types of trees, and the training data includes the attribute of leaf orientation relative to the center of the image. If we present the classifier with a combined image of two trees whose branches are intertwined, it would be impossible to definitively determine the tree species using only this classifier. However, if we supplement the classifier's output with information about the location of tree trunks and the rule that leaf orientation is calculated from the tree trunk, then through simple reasoning, we can accurately determine the tree species. In other words, we solve the problem using both machine learning and symbolic reasoning.

4 Machine Learning Components of Hybrid Models

4.1 Review of Machine Learning-Based Works

In the field of computer vision, transformers, and convolutional networks have been the predominant models for machine learning tasks for the last five years.

Machine learning-based approaches follow a multi-stage scheme. Firstly, segmentation methods are used to separate regions or objects from the initial image. Then, separate numerical features are calculated for each object, or image features are calculated using convolutional networks (CNNs). These features are merged, dimension reduction is performed, and finally, classification is done using various techniques like Support Vector Machines (SVM), k-nearest Neighbor (KNN), Sparse Representation-based Classification (SRC), AdaBoost, Conditional Random Forest (CRF), and DNNs like AlexNet, VGGNet, GoogLeNet, FCN, UNet, SegNet, DeepNet, and ResNet. Scientists revealed that some DNN systems achieved an accuracy of more than 90% for classifying tree species and specific tree species, [4]. The findings from the study indicate that CNN accurately classified trees according to their unique biological structures, including those found within forest stand structures. Furthermore, the study found that machine learning classifiers were highly effective in mapping forest structures, with SVM classifiers yielding an impressive 85% accuracy rate - surpassing the accuracy of the ANN classifiers, [10]. These results further demonstrate the tremendous potential of machine learning algorithms in accurately classifying and mapping complex biological structures. For machine learning-based approaches higher performance computing is required to get better accuracy results of the created results, [11]. Pixel-based approaches can present difficulties when analyzing high spatial-resolution images, [12]. This is because these methods utilize only spectral information, and do not consider spatial aspects, making them less effective for complex composite images containing objects with different relationships and hierarchies. For instance, if objects cover each other or if classes such as "trees" and "grass" are defined, it may be impossible to define a vegetation class without it being previously defined as a superclass. Additionally, these methods do not offer the ability to add spatial rules. For example, a "bush" cannot be found under a "Pinus" but may be found in a field. As mentioned above, DNN-based models are mainly used for image processing and prediction. Among them, two main directions are distinguished: CNN models and attention-based models. Let's briefly review these two directions.

Before 2017, Convolutional Neural Networks (CNNs) were considered the most accurate method for solving image classification, object detection, image retrieval, and other related applications. Several well-known CNN models, such as VGG [11], and ImageNet [13], were initially used for image classification and have since demonstrated their effectiveness in other image-processing tasks.

Attention-based models have made significant progress in recent years. Human vision can quickly scan global input information and screen out specific targets, [14], [15]. However, humans use task-driven top-down or data-driven bottom-up mechanisms of attention. These mechanisms automatically change the resolution of the image at each stage of solving the task, creating hierarchical multi-scalar representations of images.

The combination of deep learning and attention mechanisms has received considerable attention in research. The first successful model in the direction of using attention was ViT (Visual Transformer), [12]. A key element of the self-attention mechanism in ViT allows for capturing contextual representations via attending to both distant and nearby tokens. Following this trend, Vision Transformer (ViT) proposed to utilize image patches as tokens in a monolithic architecture with minor differences compared to the encoder of the original Transformer. ViT-based models have achieved SOTA (state-of-the-art) or competitive performance in various computer vision tasks. The self-attention mechanism in ViT allows for learning more uniform short and long-range information in comparison to CNN. However, the monolithic architecture of ViT and the quadratic computational complexity of self-attention make their swift application challenging.

Various architectures, including Swin Transformer, have attempted to balance short- and long-range spatial dependencies by proposing multi-resolution architectures, where self-attention is computed in local windows. In this approach, interactions across different regions are modeled using cross-window connections such as window

shifting. However, the limited receptive field of local windows challenges the capability of self-attention to capture long-range information, and window-connection schemes such as shifting only cover a small neighborhood in the vicinity of each window. To address these limitations, a new model named Global Context Vision Transformer (GCViT), [16] has been introduced. It represents a symbiosis of approaches that combines local and global contexts using global query tokens and includes a parameter-efficient downsampling module. The GCViT architecture has achieved state-of-the-art results on various tasks such as image classification, object detection, instance segmentation, and semantic segmentation with an accuracy of 85%. This method uses global context self-attention modules in combination with standard local self-attention to model both long and short-range spatial interactions effectively and efficiently. This is achieved without the need for expensive operations, such as computing attention masks or shifting local windows.

4.2 Proposed DNN Model

Although there have been great successes in the use of neural network models, there are still some challenges that need to be addressed, [17], [18]. These include issues related to accuracy, interpretation of results, and the demand for large computing resources. To ensure that models are effective, they must be simple, scalable, and easily adaptable to new tasks.

To simplify the problems, we have divided models into two classes in the paper:

1. Models that are used to solve monolithic object classification, segmentation, grouping, and other tasks using static or aggregated features.
2. Models that will work for machine learning tasks involving composite objects.

For complex and large-scale imaging tasks involving composite objects, we have represented the initial task as a sequence of these two classes of models. For the first-class models, we have used the existing classical multi-layer multilayer perceptron (MLP) -based DNN architecture. Training and test data for these models are computed from the aggregated spectral, geometric, and texture features of color images.

For the problems of the second class, we turn the initial image into a gray image, which significantly reduces the dimension of the initial problem and increases the possibilities of explanation. We will not stop at the first class, and we will show their use of examples.

To address the specific challenges posed by composite objects such as forests and trees (Figure 4), a new architecture was developed for second-class models. This is because these objects are composite objects of large dimensions and require a specific approach.

The architecture for the proposed model is displayed in a generalized form in Figure 5. The components that were developed in previous works, [19], [20] have been utilized to construct the new architecture. In addition to this, a transformer built on new blocking functions with an attention mechanism, [19] has been incorporated. This transformer has been tested in multivariate time series forecasting tasks, specifically in hypertensive disease prediction and recommendation tasks, [20].

The proposed model is a multi-layered structure that can be arranged in a hierarchical structure in either a data-driven bottom-up or a task-driven top-down manner. Each layer can also operate independently. The ***FC*** and ***PatchPartitioner*** components of the model match the architectural components of GCVit. In the PatchPartitioner, a sliding window function, [21] has been added to work with objects of different resolutions. This is shown in Figure 6, and it is different from the sliding window used in Swin, [22], which uses the entire image and requires a lot of computing resources. Sliding the window in each layer is done only by the size of the patch page in both (x,y) directions.

An original component in the *Attention Blocking Transformer* (ABT) module is the ***Blocking Transformer component***, which has two inputs: scaled image and patch tensors. For each element of the input tensor are introduced two horizons, inner and outer.

The inner horizon is formed by the mutual attention of the pixels of the image patch, and the outer horizon by the aggregated characteristics of neighboring patches (average value, maximum, minimum, etc.), which are taken from patch input.

The values of the blocking functions in each element (pixel) increase the informativeness of the pixel, which is an important innovation for the attention mechanism to implement.

For every element of the tensor, four new layers of the blocking function [20] are built - two layers for

internal and two for external blocking in both x and y directions. You can see this in Figure 7.

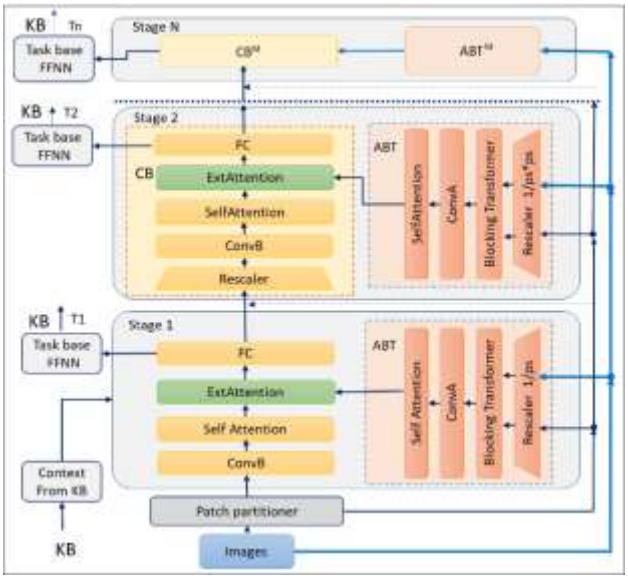


Fig. 5: Proposed architecture of deep neural networks with attention mechanisms based on blocking meta-functions

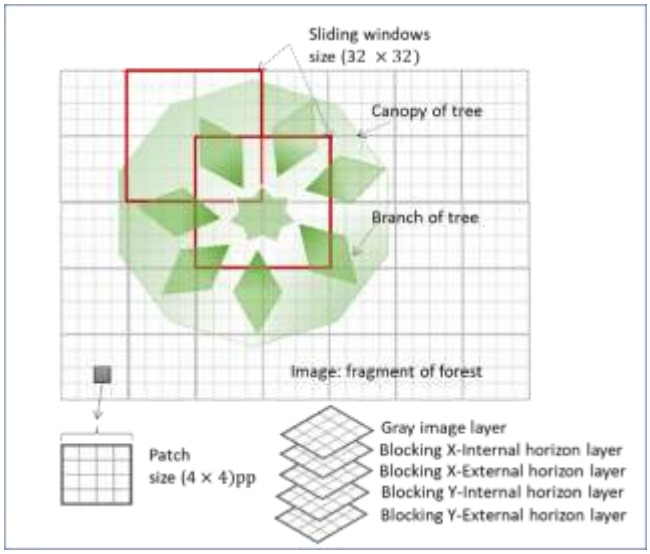


Fig. 6: A sliding window with patch partitioning and the horizons of blocking

SelfAttention and ExtAttention are components used for multi-head attention calculation. They are used to calculate self-attention and external (top-down) attention. The inputs to these components are transformed into one-dimensional flat tensors, and their initial dimensions are restored during output.

An important factor to note here is that we do not use positional embedding. This is because we do not need neighborhood information. Instead, this information is embedded in the layers (channels) of each element. This significantly simplifies the calculation scheme.

The output tensor from the ***ExtAttention*** component is then sent to the ***FC*** neuron block and the next layer. In the last stage, the tensor of the patches does not have an external attention tensor, so we only calculate self-attenuation, unlike in the lower stages.

The Rescaler module reduces the resolution of the patch size - times, both for the initial image and for the patches, as well as for the tensors coming from the previous layer.

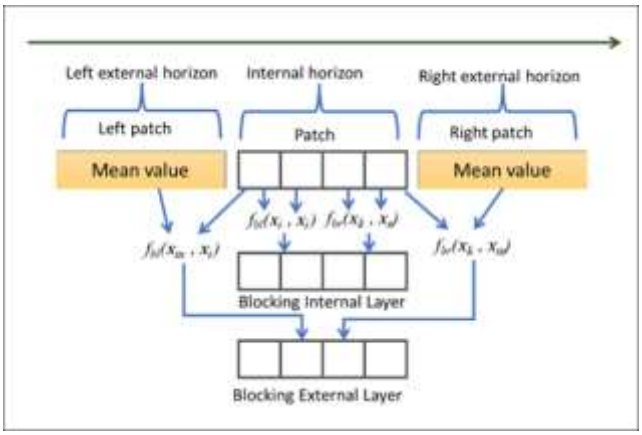


Fig. 7: An example of calculating blocking functions for internal and external horizons

ConvB the module is a sequence of CNN operators of the following types:

1. $x = Conv2D_{(ps+1)\times(ps+1)}(p_1, x, f_{a1})$,
2. $x = MaxPooling2D_{2\times2}(x) + x$,
3. $x = DW\_Conv2D_{3\times3}(p_2, x, f_{a2})$,
   (2,3-operators repeated from 1 to k)
4. $x = Conv2D_{p4\times p4}(p_5, x, f_{a3})$,
5. $x = Conv2D_{1\times1}(p_6, x, f_{a3})$

ps - is a patch size parameter, p_4-is a kernel size, p_1, p_2, p_3, p_5, p_6 – are neural latent unit numbers,
k- is a repeat -number of the 3 operators.

Operator 1 provides the organization of the sliding window on the input image, it automatically calculates the stride and the size of the embedding vector according to the parameters and creates the corresponding vectors according to the patches;
MaxPooling2D – reduces the size twice and the value of the element is calculated by the maximum of the values of the reduced elements.

DW_Conv2D - operator is a 3 × 3 depth-wise convolution operator, that provides desirable features such as inductive bias and modeling of inter-channel dependencies.

4,5 operators are used to calculate upper-level features. f_{ax} - is an activity function.

ConvA differs from ***ConvB*** in only the repeat number of operators 2 and 3.

The output tensors from the stages are sent to the ***Task base FFNN*** component.

FFNN - module represents a task-based feed-forward neural net, with corresponding activation f_a and loss f_l functions. Their solution is recorded in the knowledge base. Output value can be presented as

$$Y = FFNN\,(x, n, f_a, f_l)\,.$$

The above model was implemented in Python using the Keras library and tested on the Fashion MNIST dataset, which consists of 60,000 samples with an image size of 28x28. The model was used for a clothing classification task. Figure 8 shows the accuracy and loss function plots of our model (named "Bzik_Model") and the fast sliding window model discussed in [23] (named "FSW_Model") during training and validation. The training lasted for 15 epochs.

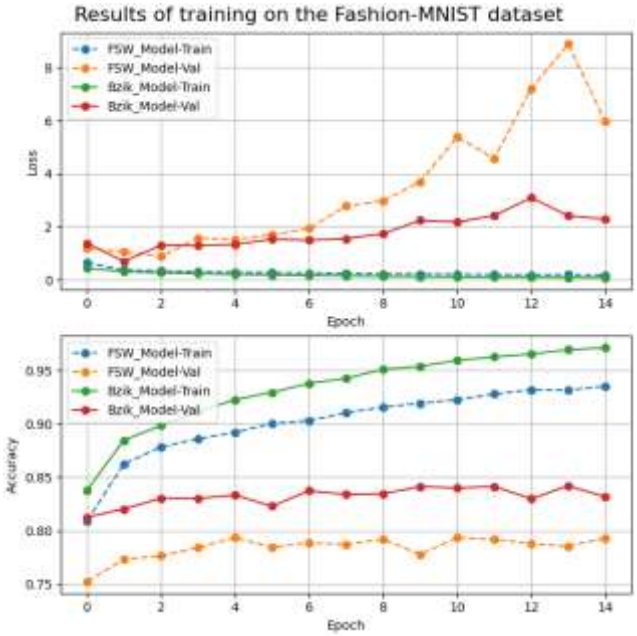


Fig. 8: Training results for the "FSW_Model" and "Bzik_Model" models on the Fashion-MNIST dataset

Based on Figure 8, our approach demonstrates superior performance. However, our model is relatively slow. On the CPU, our model takes 41 seconds to compute, while the standard model takes 29 seconds.

For the test data, two samples were chosen from the training set, and a test image sized 56x56 was created from their warped images.

The comparison of models on test data is shown in Table 1.

Table 2 compares the average accuracy of our model with other models discussed in [24] on the Fashion MNIST Dataset.

Table 1. The result of the classification was performed using models " FSW_ Model " and " Bzik Model " on a test image synthesized from two types of clothing

N	Training status	Class of clothing	FSW Model	Bzik Model
1	Trained	Class A	0.983	0.999
2		Class B	0.988	0.997
3	Not trained	Class A	0.995	0.999
4		Class B	0.912	0.974

This result is not final, and our method includes possibilities for its further development.

Table 2. Comparing the mean accuracy of our model with that of other models on the Fashion MNIST Dataset

	Models (Methods)	Accuracy
1	Three-layer Neural Network	87.23%
2	Support Vector Classifier with RBK kernel	89.70%
3	Evolutionary Deep Learning Framework	90.60%
4	CNN using the SVM activation function	90.72%
5	CNN using Softmax activation function	91.86%
6	CNN with Batch-normalization	92.22%
7	CNN with Batch Normalization and Residual skip	92.54%
8	Decision Tree Classifier	79.50%
9	Our Approach-Bzik DNN model	94.02%

5 Proposed Model Implementation

In general, identification of forest plants with images taken from UAVs includes such traditional computer vision tasks as data collection, object-based tree crown segmentation, ground truth label attachment to tree crown map, and machine learning. To solve these tasks, a prototype was developed using Python 3.9, Keras 2.13.1, and OpenCV2 libraries, based on

the developed methodology and models. Let's consider each task separately.

5.1 Data Collection

For testing our system, we utilized high-resolution Ortho-mosaic RGB images of the Sochi Forest from the Racha region of Georgia, obtained through UAV remote sensing. The National Forestry Agency of Georgia provided these images, which had a resolution of 72 pixels per inch and were 3648 pixels × 3648 pixels. In addition, the GPS coordinates of the image center and orientation were known. At this point, we did not consider the surface slope since determining tree height was not the objective. Also, we don't consider data preprocessing because it is a technical task.

5.2 Object-based Tree Crown Segmentation

Separating objects from an image is a crucial and challenging task in computer vision. This task becomes even more difficult when it comes to separating trees in a dense forest. Incorrect segmentation can significantly reduce the accuracy of subsequent species identification.

Our approach involves solving the same problem using multiple methods and comparing the results obtained from the knowledge base. Along with the traditional semantic segmentation method (UNet), we also use a modified iterative watershed method, which we implemented using the corresponding function of the OpenCV library.

The idea behind the modified iterative watershed method is to use it with high sensitivity in the first iteration and then reduce the sensitivity in subsequent iterations. Figure 9 illustrates the results of these methods after two iterations. Segmentation accuracy reached 91.11%. The algorithm found 135 crowns, out of which 12 were wrong, and of small size.

5.3 Ground Truth Label Attachment to Tree Crown Map

Labeling the different parts of a tree, such as its branches, leaves, trunks, and other components, is a crucial and time-consuming task. If labeling is done incorrectly, it can negatively impact the learning process. To make this process more efficient, we have developed a Python application that automates the labeling process. Additionally, we extract object features during the labeling process and input them into a machine-learning model for classification. This helps us determine the accuracy of the labeling. By using this process, we can obtain various patterns through augmentation, which adds to our accumulated knowledge.

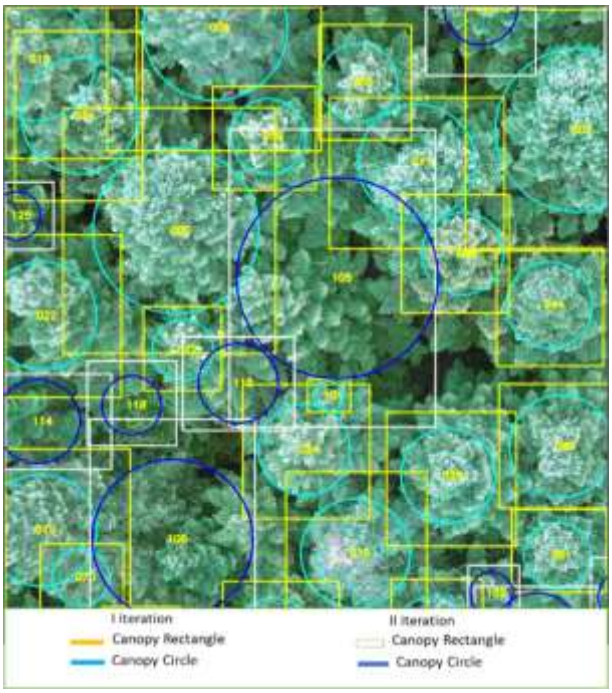


Fig. 9: Tree crown segmentation results of the Sochi Forest from Georgia's Racha region

We use images of coniferous forests for experiments, particularly those consisting mainly of Sochi (Abies) trees. To train the machine learning model, we separated samples into four classes of Abies trees:

normal, unhealthy, dry, and "other". These classes were tagged as {"Abies_Normal", "Abies_Unhealthy", "Abies_Dry", and "Other"}. Figure 10 shows the images of these components cut out from the segmented image of a coniferous forest. The objects can be cut out from the image either by the operator or the system, with the operator correcting the labels as necessary. In our case, all selected objects are reduced to the same dimension (96x96).

We then apply augmentation techniques such as flip, rotate, shift, and scale operators, while the system calculates the values of the image features. The results of the augmentation are shown in Figure 11.

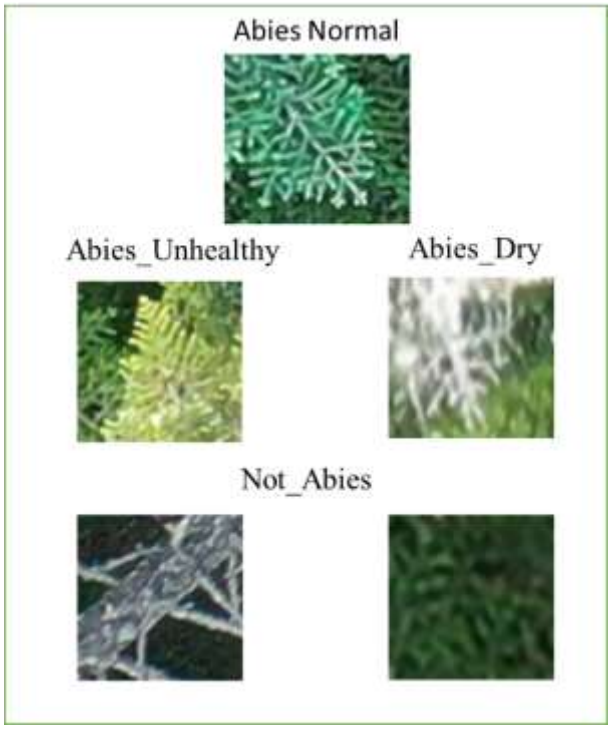
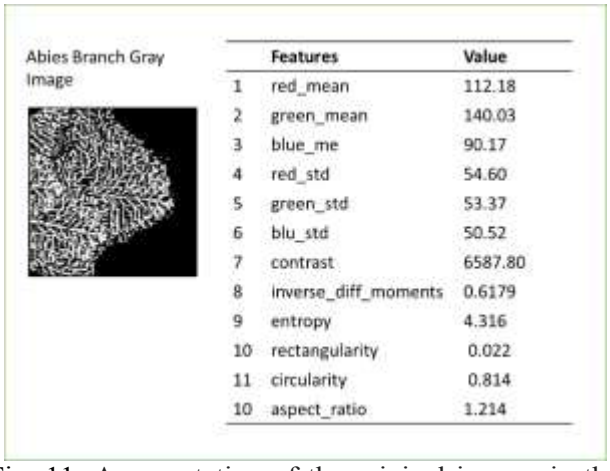


Fig. 10: The images of the Abies tree sub-classes



	Features	Value
1	red_mean	112.18
2	green_mean	140.03
3	blue_me	90.17
4	red_std	54.60
5	green_std	53.37
6	blu_std	50.52
7	contrast	6587.80
8	inverse_diff_moments	0.6179
9	entropy	4.316
10	rectangularity	0.022
11	circularity	0.814
10	aspect_ratio	1.214

Fig. 11: Augmentation of the original images in the training dataset and their static features display

After the operator's verification, the created sample is classified by the system using previously learned samples. If necessary, the operator corrects the features, or the sample is discarded.

5.4 Deep Learning

In our current research, we are employing two deep-learning models simultaneously to classify tree species. These models include the "Bzik_Model," previously introduced, and the classical MLP model. The "Bzik_Model" utilizes image data, while the MLP model uses static image characteristics as illustrated in Figure 11. This methodology allows us to assess the performance of each model and develop an effective strategy. To derive an outcome, we choose species with the highest accuracy from the evaluation result of each model. Moving forward, we aim to integrate knowledge-based and logical reasoning methods.

The proposed hierarchical machine learning model for tree species classification assumes that the size of the images of the composite objects (such as the canopy of a tree and its branches or leaves) is the same for the training samples and matches the size of the sliding window (Wr, Wc). However, when dealing with real dimensions in the classification of forest plants, the size of the forest image is much larger than the window size $(Ir, Ic) \succ\succ (Wr, Wc)$. Additionally, the size of the tree canopy is also larger than the window size $(Cr, Cc) \succ (Wr, Wc)$.

During the operation of the classification model, multiple sliding windows can be placed in the exercise area, resulting in several branches with varying classification accuracy, and in some cases, even opposite results. This becomes more complicated with the addition of two more classification models, one based on images and the other based on static features. A complex logic is required for the evaluation process. However, since this logic can be incorporated into the knowledge base by rules, we have developed a simple algorithm (Algorithm 1) to test our classification model.

5.5 Experimental Results

For the experimental study, we worked with six forest images. Through segmentation, cropping, and labeling, we generated 500 tree branch images (48x48 size). These images were then augmented, feature calculated, and used to create two training sets, each containing 22,000 samples.

Since the images were taken only for the Abies tree forest, the datasets were compiled from only three species of trees: Abies normal, Abies dray, and other.

Training and validation of the classification models created were then conducted using these datasets, with the parameters are set as follows: Epoch number -15, Batch size -32. The results of training the "Bzik-model" can be seen in Figure 12.

Algorithm 1: Assigning species to tree canopies while moving the floating window

```
Loop sw in SlidingWindowses:
    Canopies = getCanopies(sw)
    Loop cnp in Canopies:
        Branches = getBranchsMatches(cnp)
        Loop br in Branches:
            Br_lab = getMaxWeightedLabel(br)
            Br_w = getMaxWeightValue(br)
            If Cnp.specie == None:
                Cnp.specie = Br_lab
                Cnp.weight = Br_w
            Else:
                If Cnp.specie == Br_lab:
                    Cnp.weight = max (Cnp.weight, Br_w)
                Else:
                    If Cnp.weight < Br_w:
                        Cnp.specie = Br_lab
                        Cnp.weight = Br_w
```

Fig. 12: The results of training the "Bzik-model"

To create a test dataset, images of three types of trees (Abies normal, Abies dray, and Other) were randomly extracted from forest images and combined to form sets of four images each (each image size 96x96). These synthetic datasets were used to assess the models. In Figure 13, the test results for a sample are illustrated through heat maps. The left image displays the heat map generated by the Bzik DNN model, while the right image shows the heat map produced by the MLP NN model. It is evident from the figure that the Bzik model fails to recognize the two bottom images, whereas the MLP model does not recognize the two images on the left.

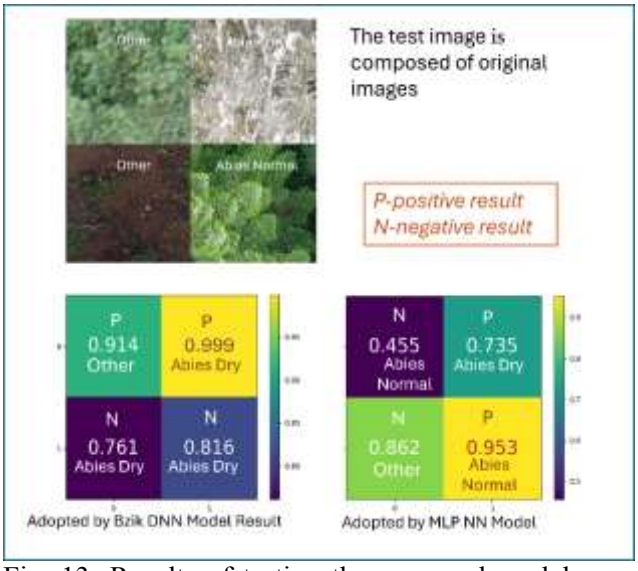


Fig. 13: Results of testing the proposed model on a composite image of the original images of branches

However, through a combined analysis of these models' test outcomes, accurate categorization of tree species is achieved. The experiment demonstrated that through the integration of various machine learning models, along with their collective knowledge and inference mechanisms, we can significantly enhance the classification accuracy even for intricate images.

6 Conclusion

Our research paper introduces a new hybrid artificial intelligence model for identifying trees using RGB images captured by UAVs. This model combines machine learning, symbolic inference, and system operations management in a unified formal framework. A key element of the hybrid model is a deep neural network with a novel hierarchical architecture, featuring multi-layer convolutional neural networks, attention transformers, and a new type of transformer based on blocking meta-functions. This model offers two important features: it allows problem-solving from both bottom-up and top-down approaches, and information from each layer can be sent to the knowledge base, enhancing both model explainability and problem-solving accuracy. The model's implementation and

experimental research on toy tasks and real tasks have demonstrated strong performance.

In future studies, we plan to develop the knowledge base of the hybrid model's second component using abductive symbolic reasoning and to conduct experimental research in conjunction with the machine learning model under different environmental and time conditions. We also aim to create a self-learning operation management module for the unified system, further enhancing the efficiency of tree species identification.

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

During the preparation of this work the authors, the authors used Google Translate, Grammarly (free version), and ChatGPT 3.5 in order to correct and improve the article's translation from Georgian to English. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] Christoph Kleinn, Gerald Kändler, Heino Polley, Thomas Riedel, Friedrich Schmitz, The National Forest Inventory in Germany: Responding to Forest-Related Information Needs. *Allgemeine Forst und Jagdzeitung*, Vol. 191, H. 5/6p. 97-118, 2020, https://doi.org/10.23765/afjz0002062.

[2] Sarah Kentsch, Savvas Karatsiolis, Andreas Kamilaris, Luca Tomhave, Maximo Larry Lopez Caceres, Identification Of Tree Species In *Japanese Forests Based On Aerial Photography And Deep Learning*. [cs.CV], arXiv:2007.08907 [pdf], 2020.

[3] Xianggang Chen Xin Shen, Lin Cao, Tree Species Classification In Subtropical Natural Forests Using High-Resolution UAV RGB And Superview-1 Multispectral Imageries Based On Deep Learning Network Approaches: A Case Study Within The Baima Snow Mountain National Nature Reserve, China., *Journals Remote Sensing*, Vol. 15, Issue 10, 2697, DOI: 10.3390/rs15102697, 2023.

[4] Masanori Onishi, Takeshi Ise, Explainable identification and mapping of trees using UAV RGB image and deep learning. *Scientific Reports*, Vol. 11, article number: 903 (2021), https://www.nature.com/articles/s41598-020-79653-9 (Accessed Date: June 25, 2024).

[5] Clopas Kwenda, Mandlenkosi Gwetu, Jean-Vincent Fonou Dombeu, Learning Methods for Forest Image Analysis and Classification: A Survey of the State of the Art, IEEE*Access, Multidisciplinary, Open Access Journal*, Vol. 10, 2022, 45290-45316, DOI: 10.1109/ACCESS.2022.3170049.

[6] Vaishak Belle, Symbolic Logic meets Machine Learning: A Brief Survey in Infinite Domains, *Scalable Uncertainty Management: 14th International Conference, SUM 2020*, Bozen-Bolzano, Italy, September 23–25, 2020, DOI: 10.48550/arXiv.2006.08480.

[7] Giuseppe Marra, Francesco Giannini, *Integrating Learning and Reasoning with Deep Logic Models*, Michelangelo Diligenti, Marco Gori, arXiv:1901.04195,2019, DOI: 10.48550/arXiv.1901.04195.

[8] Wang-Zhou Dai, Qiuling Xu, Yang Yu, Zhi-Hua Zhou, Bridging Machine Learning and Logical Reasoning by Abductive Learning, *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, [Online]. https://www.researchgate.net/publication/338750187_Bridging_Machine_Learning_and_Logical_Reasoning_by_Abductive_Learning (Accessed Date: June 25, 2024).

[9] Kamarulzaman, A.M.M., Wan Mohd Jaafar, W.S.; Abdul, Maulud, K.N.; Saad, S.N.M.; Omar, H., Mohan, M. Integrated, Segmentation Approach with Machine Learning Classifier in Detecting and Mapping Post Selective. *Logging Impacts Using UAV Imagery. Forests* 2022, 13, 48, DOI: 10.3390/f13010048.

[10] Yasushi Minowa, Yui Nagasaki, Convolutional Neural Network Applied to Tree Species Identification Based on Leaf Images, *Japan Society of Forest Planning, Journal of Forest Planning*, 26: 1-11 (2020), DOI: 10.20659/jfp.2020.001.

[11] He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. DOI**:** 10.1109/CVPR.2016.90.

[12] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F. E., Feng, J., and Yan, S. *Tokens-to-token ViT: Training vision transformers from scratch on ImageNet*. In ICCV, arXiv:2101.11986 [cs.CV]. 2021, DOI: 10.48550/arXiv.2101.11986.

[13] Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *In Proceedings of the International Conference on Neural Information Processing Systems*, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105, [Online]. https://www.researchgate.net/publication/319770183_Imagenet_classification_with_deep_convolutional_neural_networks (Accessed Date: June 25, 2024).

[14] Yadong Yang, Chengji Xu, Feng Dong and Xiaofeng Wang, A New Multi-Scale Convolutional Model Based on Multiple Attention for Image Classification, *Applied Science*. 2020, 10, 101, DOI: 10.3390/app10010101.

[15] Zhang, J.M.; Bargal, S.A.; Lin, Z.; Brandt, J.; Shen, X.H.; Sclaroff, S. *Top-Down Neural Attention by Excitation Backprop*. arXiv:1608.00507v1 [cs.CV] 1 Aug 2016, DOI: 10.48550/arXiv.1608.00507.

[16] Ali Hatamizadeh, Hongxu Yin, Greg Heinrich, Jan Kautz, Pavlo Molchanov, *Global Context Vision Transformers*, arXiv:2206.09959v5 [cs.CV] 6 Jun. 2023, https://doi.org/10.48550/arXiv.2206.09959.

[17] Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Ozokerite J., Houlsby N., An image is worth 16x16 words: Transformers for image recognition at scale. *In International Conference on Learning Representations*, 2020. arXiv:2010.11929v2 [cs.CV] 3 Jun. 2021, DOI: 10.48550/arXiv.2010.11929.

[18] Simonyan, K.; Zisserman, *A. Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv 2015, arXiv: 1409.1556,DOI: 10.48550/arXiv.1409.1556.

[19] Zurab Bosikashvili, Giorgi Kvartskhava, Merab Machavariani - Identification and analyses of tree species using UAV images of forests and deep machine learning models, *International scientific-practical conference modern challenges and achievements in information and communication technologies* – 2023, Technical University, 2023, ISBN: 978-9941-512-06.

[20] Zurab Bosikashvili, Tamar Bosikashvili, Ketevan Bosikashvili - Using Blocking Meta-heuristics in the Models of Deep Learning, *International scientific-practical conference modern challenges and achievements in information and communication technologies* – 2023, Technical University, 2023 ISBN: 978-9941-512-06.

[21] Henry G. R. Gouk, Anthony M. Blake, Fast Sliding Window Classification with Convolutional Neural Networks, 2014, [Online]. https://www.kaggle.com/code/gaborvecsei/convolutional-fast-sliding-window-wip (Accessed Date: June 25, 2024).

[22] Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo, *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*, arXiv:2103.14030v2, 17 Aug 2021, arXiv:2103.14030 [cs.CV], DOI: 10.48550/arXiv.2103.14030.

[23] Jun Xu, Yumeng Wei, Aichun Wang, Heng Zhao, Damien Lefloch, Analysis of Clothing Image Classification Models: A Comparison Study between Traditional Machine Learning and Deep Learning Models, Research Article, *Fibres & Textiles in Eastern Europe,* 30(5), 2022, 66-78, DOI: 10.2478/ftee-2022-0046.

[24] Irene Pylypenko, Exploring Neural Networks with fashion MNIST, Medium, [Online]. https://medium.com/@ipylypenko/exploring-neural-networks-with-fashion-mnist-b0a8214b7b7b (Accessed Date: June 25, 2024).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Zurab Bosikashvili contributed to the development of the problem-solving methodology and formal model. Additionally, he was involved in the development of the deep learning model and its implementation in Python.
- Giorgi Kvartskhava contributed to setting the task, collecting, and characterizing the data, developing data preprocessing algorithms, testing the system, and organizing experiments.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study**.**

Conflict of Interest

The authors have no conflicts of interest to declare.