IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 January 2023, accepted 18 March 2023, date of publication 23 March 2023, date of current version 28 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3261129

RESEARCH ARTICLE

Estimation of Vegetation Indices With Random Kernel Forests

DMITRY A. DEVYATKIN
Federal Research Center "Computer Science and Control," RAS, 119333 Moscow, Russia
e-mail: devyatkin@isa.ru

This work was supported by the Ministry of Science and Higher Education of the Russian Federation under Project 075-15-2020-805.

ABSTRACT Vegetation indexes help perform precision farming because they provide useful information regarding moisture, nutrient content, and crop health. Primary sources of those indexes are satellites and unmanned aerial vehicles equipped with expensive multispectral sensors. Reducing the price of obtaining such information would increase the availability of precision farming. Several studies have proposed deep neural network methods to estimate the indexes from RGB color images. However, these studies report relatively large errors for mature plants, when highly non-linear relationships between image RGB bands and vegetation indexes arise. One could apply multilayer random forest-based models (Deep Forests) to solve this problem, but they have limited discriminative power and ability on catching non-linear relationships between image features. The cornerstone of the Deep Forests is that at each layer they enrich original features with embeddings containing empiric class probabilities from previous layers, although these probabilities deliver limited information. In this paper, we propose methods, which combine ideas of Deep Forests, Random Forests of multivariate trees, and global pruning of Random Forests to tackle these problems. We applied oblique (linear) and kernel (non-linear) trees as basic classifiers of the Deep Forest to improve discriminative power. We also utilized a method to refine Random Forests with a global loss optimization. This method helps to generate more expressive embeddings at each layer of the Deep Forest, which significantly improves results of the data analysis. In the experiments, we compared those methods with AlexNet and ResNet-based neural networks on several image classification datasets as well as on the NDVI prediction task. The experiments on image classification show that the proposed Deep Forest-based methods provide competitive results on datasets with small and medium size of feature-set. The results of the NDVI prediction task indicate that these methods are robust to senescence of plants and outperform neural network-based solutions.

INDEX TERMS Random forests, deep forest, oblique trees, multivariate trees, forest refinement, NDVI, RGB images.

I. INTRODUCTION

Vegetation indexes help estimate many crucial agricultural indicators such as moisture, nutrient content, and crop health. One of the most frequently used vegetation indexes is NDVI (normalized difference vegetation index). Primary sources of those indexes are satellites and unmanned aerial vehicles (UAV) equipped with expensive multispectral sensors. Reducing the price of obtaining such information would increase the availability of precision farming for small farms. Several studies have proposed deep neural network methods to estimate the indexes from RGB color images. However, as it is pointed out in [1] the framework provides accurate results only for immature plants, and the reason is the highly non-linear relationship between RGB colors and vegetation for senescent plants. Consequently, the "frequency bias" phenomenon in neural network training comes into effect [2], [3].

In this paper, we propose to utilize a Deep Forest to deal with that issue because this approach does not use smooth models, but at the same time, it provides competitive results on image processing [4]. Deep forest uses a cascade structure to perform layer-by-layer processing of raw features. However, several studies report that Deep Forest

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

has a limited ability to catch dependencies between features, which can lead to poor performance in some cases [5]. It also uses Extremely Random Forests to control over-fitting, which in practice can lead to unstable results. We modified Deep Forest to overcome those issues as follows:

1) The use of Random Forests of decision trees with multivariate oblique or non-linear splits as the basic classification algorithm allows considering the relationships between the features of the analyzed objects, reducing the number of data processing layers and, consequently, improving performance [6], [7].
2) Instead of using Extremely Random Forests to control over-fitting, we apply refinement and pruning from study [8]. This approach also helps to generate more useful layer-level feature embeddings because the refinement utilizes a regularized ensemble-level loss optimization. This way we increase both the accuracy and the stability of the Deep Forest method.

II. RELATED WORK

Visual analysis of land is the primary tool of precision agriculture. For example, vegetation indices obtained by the analysis of multispectral images help monitor crop health. Paper [9] proposes a remote sensing recognition method based on a convolutional neural network. They combine 4 channels (red, green, blue, and near-infrared) to reveal the changing features of the landslide. Finally, the convolutional neural network was applied to solve the problem. The experiments showed that the method is more accurate than traditional methods. The high cost of multispectral cameras led researchers to focus on the analysis of pure RGB color images. The paper [1] uses a convolutional neural network to reveal the non-linear relationship between a color land image and related vegetation indexes. This network obtains vegetation indexes of various crops. Experiment results show that the obtained values agree with ground-estimated indexes. However, they also revealed that the method provides accurate outputs only until appearance of senescence.

Paper [10] proposes a deep-learning-based method to generate a multi-spectral (MSI) data from pure RGB images. They first reproduced RGB images from multi-spectral ones through a rendering model, which was built with a benchmark hyperspectral image dataset. Then, they used the obtained RGB and MSI pairs to train a reconstruction model. They revealed that the mean relative absolute error and spectral information divergence losses are the most effective in terms of accuracy and robust to different seasons and plant species. They also performed estimation of NDVI index based on the obtained multispectral values to assess the reliability of the proposed method to solve practical problems. They used HSCNN+ convolution networks with residual connections as the model to recover MSI from RGB. This model combines multiple convolution layers with residual connections. It is proven to provide accurate hyperspectral predictions based on RGB inputs [11]. In this paper we also tested residual and convolutional-based networks as competitive baselines for the proposed method.

Paper [12] shows a method to estimate vegetation indexes with a cheap RGBN (RGB + near infrared) camera and machine learning algorithms. Experiment results provide a comparison of the results obtained with a multispectral camera and the predictions of the RGBN camera-based solution to analyze corns under different nitrogen and water treatments. They show that the proposed approach achieved high performance at estimating vegetation indexes with the machine learning model. Study [13] proposes a method to process high-resolution drone images consisting of RGB and near-infrared bands to detect vegetation indexes. The experimental results provide insight into applying drones and neural networks as a solution for precision agriculture.

In situations where there is no direct access to UAVs or specific agricultural areas, researchers often use open information from Sentinel satellites. For example, in the paper [14] Sentinel 2 data and a multilayer perceptron are used to classify a forest structure. The researchers utilized field surveys to obtain forest vertical structures of a particular forest region. Finally, they built the forest vertical classification map, the perceptron based on the Sentine's RGB images. The evaluation shows that the model achieves fair level of classification accuracy.

All the studies above utilize neural networks to estimate vegetation levels. However, as it was pointed out above the framework provides accurate results only for immature plants. We believe the reason is that the ''frequency bias'' phenomenon of training comes into effect [3]. Besides, as is pointed out in [4], modern deep neural network require a lot of labeled data to be trained reliably.

One of the possible solutions to that problem is to use Random Forest-based ensembles. These models are widely utilized in different areas because they achieve more robust and stable performance than others [15]. Deep Forest is a multilayer cascade model based on non-differentiable modules (Random Forests) in contrast to deep neural networks. Deep forests were initially proposed to solve classification tasks [4]. Paper [16] proposes a modified Deep Forest where prediction values of the input layer forest module are processed to obtain the layer regression vector, which is combined with the raw feature vector as the input of the middle layer forest model. This method has the advantages of simple hyper-parameter setting criteria.

Paper [17] presents a detailed analysis that shows Deep Forests have sufficient model complexity with enough depth, and the cascaded structure boosts the feature representations layer by layer instead of the predictions. Many experiments show that Deep Forest has comparable performance to deep neural networks; therefore, it has been applied to solve many real-world data and text mining problems. Primary efforts in developing this approach focus on tuning it to solve various machine learning settings. For example, study [18] proposes a Deep Forest algorithm for multi-instance learning.

The experiments show this algorithm achieves competitive results. Yang with colleagues present a multi-label learning Deep Forest algorithm, which utilized measure-aware feature re-use and layer growth to solve a multi-label learning problem [19]. Paper [20] presents an adaptive weighted Deep Forest. The training procedure of this forest assigns weights to each training sample at each level of the model just like the AdaBoost approach.

Although Deep Forests show competitive results on many problems, there is still room for improvements related to considering various feature interactions.

For example, Chen et al. argue that the prediction-based feature representation of Deep Forest is a critical deficiency because the predicted class probabilities deliver very limited information [5]. They present a deep forest model that utilizes high-order interactions of input features to generate more informative and diverse feature representations. They created a generalized version of Random Intersection Trees to reveal stable high-order relationships and apply detected linear combinations to transform them into hierarchical distributed representations. With these relationship-based representations one does not need to store Random Forest [21] models for the front layers, which improves computational efficiency. The provided experiment results show that the proposed forest achieves competitive classification scores with significantly reduced time and memory costs.

However, the method above can catch only linear feature combination, which may limit it performance on data with high-order dependencies. Another way to catch those feature interactions is to use more complex decision splits in ensemble trees. Recent studies propose to use deep neural networks as those complex models [22], [23], although, this way we lack primary benefits when analyze non-smooth dependencies. In this paper, we replaced standard Random forests in the layers with more complex forests of multivariate trees to tackle this problem [7].

III. METHODS TO PREDICT NDVI

A. CO2 FOREST

CO2 Forest (Forest of Continuously Optimized Oblique trees, CO2) [7] contains trees with multivariate oblique (linear) splitters. In the CO2 Forest, decision stump training is defined as a structured prediction problem with latent variables [24]. A convex-concave loss function is used to solve the problem. Note that this optimization can be done by the gradient method proposed in [25]. Although the loss function is an upper bound of empirical risk, the ''smoothness'' of that loss and tightness on the bound are affected by hyperparameters and scale of the features. In practice, that means the method requires thorough fine-tuining of the hyperparameters.

B. KERNEL FOREST

Kernel Forest is a Random Forest built from trees with multivariate non-linear decision splitters [26]. In general, the algorithm to train those trees utilizes the same top-down induction as for ordinary CART trees [27]. The primary difference is that it trains multivariate decision splitters via solving a convex optimization problem. Namely, for each splitter, the algorithm greedily finds a quasi-optimal distribution of classes to subtrees (in terms of impurity minimization) and then uses this distribution as labeled data to train this splitter as an SVM-like binary classifier. They use a margin re-scaling approach [24] to optimize both the margin between subtree data and arbitrary impurity criteria (Gini impurity, Information gain, etc.).

For each decision stump, parameters of the decision surface are obtained with the following optimization problem:

$$a^* = \arg\max_a -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K(x_i, x_j) + \sum_{i=1}^{n} a_i, \quad (1)$$

s.t.

$$\sum_{i=1}^{n} \frac{a_i}{L(h_i, -h_i)} \leq \frac{C}{n} \quad (2)$$

where x_i is features of the object with index i, $h_i \in \{-1, +1\}$ defines the target subtree for the sample with index i, a_{ij} is the weight of the training sample i (non-zero for the support vectors), $L(h_i, -h_i)$ reflects the growth of the impurity criterion in case of miss-classification, $K(x_i, x_j) : \mathbb{R}^f \times \mathbb{R}^f \rightarrow \mathbb{R}$ is a kernel function for objects with the feature-set size f, C is the regularization term, and n is the size of the training dataset.

C. KERNEL FOREST REGRESSOR

Many practical applications in agriculture such as vegetation index detection require solving regression problems. Regressors map object features to some target real vectors $X \rightarrow \mathbb{R}^m$ with m dimensions. In this study, we propose a way how to modify the classification method from [26] (Kernel Forest) to solve those problems. This method utilizes standard top-down induction of a decision tree, and at each step of this induction, it performs training of a kernel-based decision stump.

In a regression tree, the decision stump assigns some real vectors R_1 and R_2 to the left and right subtrees. The Kernel Forest requires those assignments to be done before actual training of the decision split. Therefore, we need to pick up those values in such a way as to minimize the average distance between all the training samples lying at each subtree. In this study, we utilized the K-Means clustering algorithm with $K = 2$ to find those values. K-Means algorithm clusters samples by separating them into groups of equal variance, minimizing within-cluster sum-of-squares, i.e. it fits out goal. Finally, we use the found cluster centroids as values of R_1 and R_2.

Another feature of the method [26] is that it scales the training sample weights accordingly to their effect on the impurity criterion (Gini impurity, information gain, etc.). In case of regression, we use the mean square error (MSE) between all the training samples of a particular subtree h to the values assigned to this subtree instead of impurity growth.

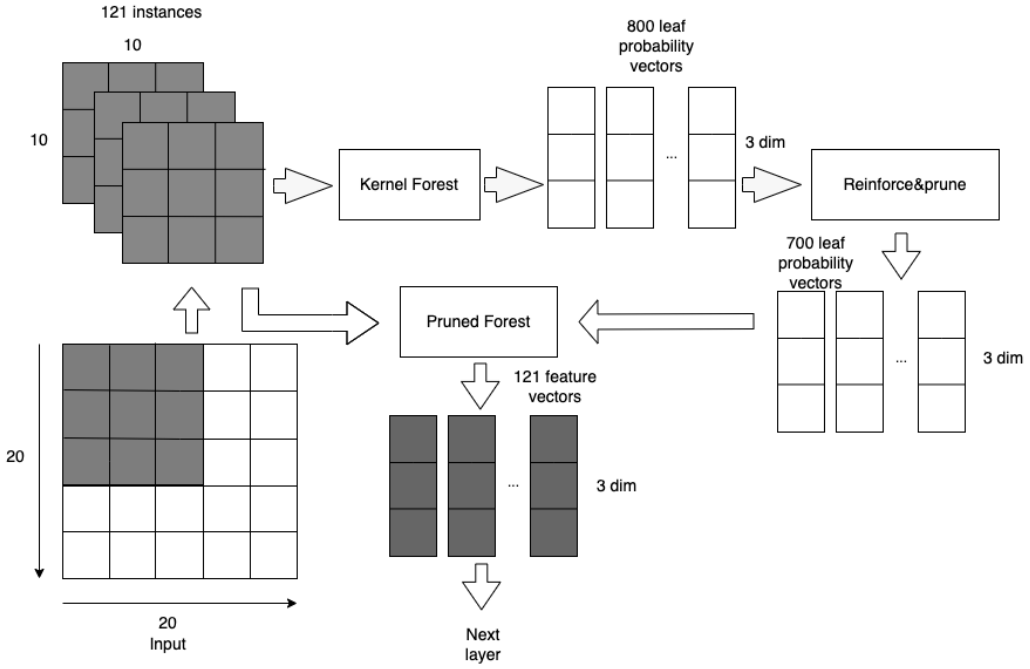

FIGURE 1. Training of the input layer of the proposed algorithm (Deep Kernel Forest/Deep Oblique Forest, DKF/DOF). The input image size is 20 × 20, the sliding window size is 10 × 10, and the number of classes is 3. The Kernel Forest contains 800 leaves and pruning removes 100 of them.

Lets the subtree has n training samples and the assigned real vector $R \in \mathbb{R}^m$. Therefore, the criterion for this subtree is:

$$g(h) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(R_j - y_{ij}\right)^2 \tag{3}$$

D. DEEP OBLIQUE FORESTS AND DEEP KERNEL FORESTS

The method is a modification of the Deep Forest, in which data is processed sequentially on several layers. The input layer has the following structure (Fig. 1). In that layer, the multi-grained scanning [4] generates a set of objects based on each sample from the training set and all those generated objects are labeled with the class of the original sample. For example, let the input image has the size 20 × 20, the sliding window has the size 10 × 10, and the classification problem has 3 classes. Multi-grained scanning generates 121 feature vectors. These vectors are used to train a Random Kernel Forest [26] or CO2 Forest [7], then the trained forest is refined and pruned with the method from [8]. Suppose the forest has 800 leaves, and the pruning procedure left only 700 of them. All the leaves have correspondent 3-dimensional synthetic probability vectors generated with the Ren's method [8]. Finally, those 700 vectors are utilized to generate 121 feature vectors that are used as the input together with the original features in the next layer.

The basic idea of that refinement procedure is to replace the original class empirical probabilities stored in all tree leaves of a pre-trained forest with the synthetic ones generated by explicitly minimizing a global ensemble loss function. Suppose the forest has T trees with Γ leaves on each tree. Let $\Phi : \mathbb{R}^{\mathrm{f}} \rightarrow \{0, 1\}^{T\Gamma}$ be a function that for any sample x returns the binary vector, whose elements are 1 if x goes to the corresponding decision tree leaf and 0 otherwise.

$$\Phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_{T\Gamma}(x)) \tag{4}$$

Let matrix W with $n_{classes} \times T\Gamma$ holds the class probabilities for all the leaves of decision trees in the Forest.

$$W = (w_1, w_2, \ldots, w_{T\Gamma}) \tag{5}$$

Then the refined classifier has the following decision rule:

$$y = W^* \Phi(x) \tag{6}$$

where W^* is the matrix of refined class probabilities. It can be found with the following SVM-like optimization on a training set with size n.

$$W^* = \arg\min_{W} \frac{1}{2}||W||^2 + \frac{C}{n} \sum_{i=1}^{n} l(y_i, \hat{y}_i),$$
$$s.t. y_i = W\Phi(x), \forall i \in [1, n], \tag{7}$$

where C is a regularization term, and $l(y_i, \hat{y}_i)$ is a loss function.

This approach helps consider various relationship between ensemble tree leaves and improve discriminative power of the Forest. However, the global optimization (expression 7) can lead to over-fitting of the Forest. To tackle this we use a Ren's global pruning that merges the insignificant leaves to reduce the number of model parameters. Two neighbor leaves are considered insignificant if the norm of their synthetic class probability vectors are close to zero.

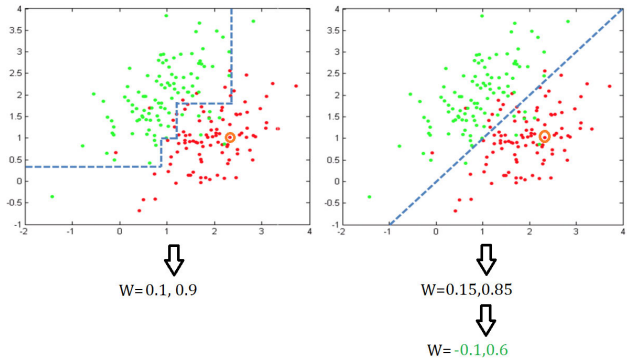

FIGURE 2. Difference in the feature generation in Deep Forest (left) and Deep Kernel (Oblique) Forest (right).

After the pruning procedure, the Random Kernel Forest or CO2 Forest is used to form embeddings of processed samples for the next layer. These embeddings represent the generated synthetic class probabilities from trees of the kernel forest. The embeddings also include the original features. In practice, as in the original study [4] we utilize cross-validation to estimate the embeddings because it reduces the bias of the obtained values. Fig. 2 highlights difference in the feature generation procedure in Deep Forest and Deep Kernel or Oblique Forests. Deep Kernel or Oblique Forests allow obtaining less fragmented regions in the original feature-set. The global refinement procedure leads to forming more helpful embeddings for the next layer.

The next layer has the same structure, except it does not perform multi-grained scanning. The following layers can be added to the model until accuracy scores on cross-validation keep growing. Implementation of the proposed Deep Forest modifications together with experimental code is presented online [28].

E. CONVOLUTION NETWORK-BASED METHODS

In this study, we tested two deep neural networks as a baseline. AlexNet is a deep convolutional neural network [29]. The network contains four convolution layers, three pooling layers, a dropout mechanism for regularization, and a multilayer dense network to produce output (Fig. 3). On the one hand, AlexNet has a few hidden layers, which means it has limited discriminative power. On the other hand it has a relatively small number of parameters, which makes it applicable to analyze small datasets.

Deeper networks allow for more complex relationship detection in various features and classes. Recent experiments show that specialized residual networks like HSCNN+ provide accurate results in multi-spectral data recovery from pure RGB images. In this study, we focused on a simpler problem of NDVI prediction, which can be done with more general residual architectures. Namely, we applied ResNet neural network [30] with 9 convolutional layers, an average pooling layer, dropout mechanism, and an output dense network. ResNet also contains shortcut residual connections between

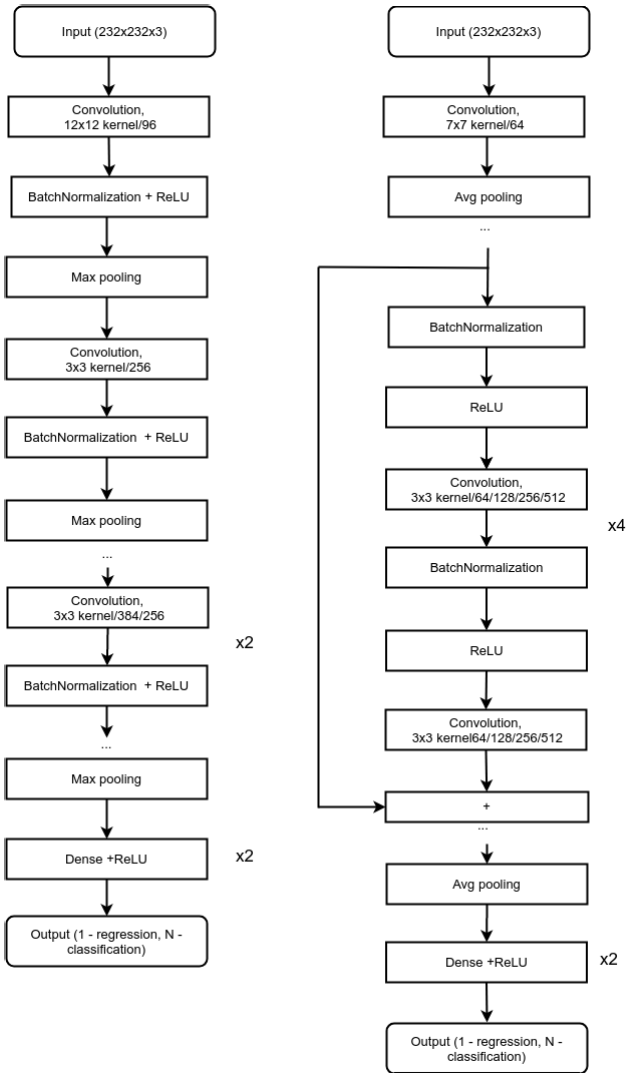

FIGURE 3. Architectures of the AlexNet (left) and ResNet-11 (right) networks.

convolution layers which helps to speed the training up and tackle gradient vanishing in large number of layers [31].

For the classification tasks we used Triplet loss [32] because it was reported as one of the most suitable for image classification tasks in several studies [33], [34]. The NDVI prediction is a regression problem, therefore we used l^2-loss, presuming that the error is distributed normally.

In both models, the output dense layer outputs an embedding with class probabilities for the classification setting, and a single real value for the regression.

F. COMPLEXITY ANALYSIS

a: NEURAL NETWORK MODELS

Table 1 shows the training complexity scores for the neural networks and their particular layers. Generally, these models contain a 2 layer full-connected (dense) network, convolutional, and pooling layers. Namely, for AlexNet we have 4 convolutional and 3 pooling layers and one 2 layer

dense network. For the ResNet-11 we have 9 convolutional layers with residual connections, one pooling layer, and one dense network with a hidden layer. As far as the training loss for neural networks is generally non-convex, one cannot guarantee a convergence of the training process to the global optimum; therefore, we just set a fixed number of training epochs p.

The training complexity for convolution and dense networks are well-known from literature [35], [36]. We stick the following ideas to obtain the complexity for the pooling layer. Pooling layers perform a finding of maximum or average value inside the filter with size k for each sample from a dataset with size m and input size $\mathfrak{f}$. In total, it processes $\mathfrak{f}-k+1$ filters, which is no bigger than $\mathfrak{f}$.

Although the training complexity for both networks is generally linear from the number of samples, it is polynomial from the number of features and sizes of layers. Therefore, the number of calculations at each layer $\sim k\mathfrak{f}d$ is pretty large for image classification and regression tasks, it can be compared to the size of datasets. Besides, the number of epochs p to perform training can also be pretty large (30-100).

b: DEEP FORESTS

Table 1 also shows the train complexity for the original Deep Forest and the proposed modifications (Deep Oblique Forest, Deep Kernel Forest). Training cost of Random Forests linearly depends on the size of the dataset m. Besides, Deep Forest performs k-fold cross-validation at each level to obtain class probability estimates.

In general, kernel trees follow the same greedy procedure to train as the univariate ones. However, the cost of decision stump training, in that case, depends on the split type (linear or more complex one) and on the particular optimization method used to train the stump. Therefore, for the Deep Kernel Forest with oblique decision splits the training cost is linear. Table 1 shows the training cost linearly depends on the number of training samples m; therefore, Deep Kernel Forest with oblique decision splits can be trained on large data sets effectively.

For the Deep Kernel Forest with non-linear decision splits there is square dependence between the training cost and the size of the dataset. That means, Deep Kernel Forest with non-linear decision splits can hardly be trained on large-scale datasets. However, the training process of the Random Forest can be parallelized effectively even at the level of decision stumps, which means the training can be performed with several CPUs on a distributed or a virtual cluster in a cloud. Besides, the dependence from the size of feature-set $\mathfrak{f}$ in contrast to neural networks is linear that reduces overall complexity of the training for image processing.

IV. DATASETS

A. STANDARD IMAGE RECOGNITION DATASETS

First, we conducted experiments on three standard UCI multi-class datasets, and the CIFAR-10 image dataset. We used USPS, Letter, and MNIST from the UCI [37]. They are devoted to image recognition problems. For example, the MNIST and USPS datasets contain handwritten images of digits, while the Letter dataset contains Latin letters. The CIFAR-10 dataset is also related to image recognition [29]. It contains 32 by 32 colored images of 10 classes (airplane, horse, bird, etc.) with eight gray levels. We apply a simple preprocessing technique to all the image recognition datasets. Namely, we perform feature-level normalization of the data with ''MinMaxScaler'' and ''Normalize'' tools from Scikit-Learn [38]. No other complex processing is used.

B. RGB-NDVI PREDICTION DATASET

We collected a dataset to evaluate RGB to NDVI models as follows. Generally, we have stuck the procedure described in [39]. First, we obtained several multispectral satellite images of rural areas in Europe from April to October 2018. We chose mostly rural areas with grain crops (wheat) and manually filtered out images with clouds or other artifacts. We got the high-resolution multispectral raster data (RGB and Infrared) of the Sentinel-2 satellite from the Copernicus web platform [40]. Then we applied QGIS Desktop software tool to evaluate *NDVI* based on red and infrared bands. With those bands one can use a simple expression to evaluate that index:

$$NDVI = \frac{I - R}{I + R}, \tag{8}$$

where I is infrared level and R is red level.

Finally, we generated RGB and NDVI images with size 232×232, then we store RGB images as the sample features and NDVI average values as the labels. The size of the obtained dataset is 1000 samples for training and 1000 for validation. We also divided all the dataset into two pieces: the first one covers the data range from April to June (Spring), and the second one covers the range from July to October (Summer/Autumn, Fig. 4). Here we utilize a presumption that the first subset should contain mostly images of immature plants, while the second one contains images of mature ones. Therefore, accuracy scores would be different for those subsets.

As for the classification datasets, we did not perform any complex feature pre-processing, but just process the data with ''MinMaxScaler'' and ''Normalize'' tools from [38]. The dataset is available in the RGB-NDVI repository [41].

V. EXPERIMENT RESULTS

In the experiments we tested the following models:

- Deep Forest: The original Deep Forest (gcForest) with standard Random Forest and Extremely Random Forest classifiers [4].
- Random Kernel Forest: a forest with multivariate decision trees and non-linear decision splits [26].
- Deep Kernel Forest: the Deep Forest, in which basic classifiers are replaced by Random Kernel Forests with Gaussian kernel.

TABLE 1. Training complexity of the considered methods, where k is the filter size, m is the size of the dataset, $\mathfrak{f}$ is the feature set size, d is the output dimension, p is the number of epochs, M is the maximum number of decision stumps in the forest trees (depends on the tree depth), T is the size of the ensemble, K is the number of folds in the cross-validation, and n is the number of layers in the Deep Forest.

Model	Training complexity
Random Forest [26]	$O(m\mathfrak{f}MT)$
Oblique Forest (linear) [26]	$O(m\mathfrak{f}MT)$
Kernel Forest (non-linear) [26]	$O(m^2\mathfrak{f}MT)$
Deep Forest	$O(Knm\mathfrak{f}MT)$
Deep Oblique Forest	$O(Knm\mathfrak{f}MT)$
Deep Kernel Forest	$O(Knm^2\mathfrak{f}MT)$
Convolutional layer [35]	$O(k\mathfrak{f}md^2p)$
Pooling layer	$O(k\mathfrak{f}mp)$
Two layer dense network [36]	$O((\mathfrak{f}+o)mph)$
AlexNet*	$O(((\mathfrak{f}d_1+1)k_1d_1+\sum_{i=2}^{4}((d_{i-1}d_i+1)k_id_i)+(d_5+o)h)mp)$
ResNet-11**	$O(((\mathfrak{f}d_1+1)k_1d_1+\sum_{i=2}^{9}(d_{i-1}(d_i^2k_i+1))+d_9+(d_{10}+o)h)mp)$

*$k_1=12, k_2\ldots k_4=3, d_1=96, d_2=256, d_3=384, d_4=256, d_5=9216, h=4096, o=1,10,26$.
**$k_1=7, k_2\ldots k_9=3, d_1=64, d_2=64, d_3=64, d_4=128, d_5=128, d_6=256, d_7=256, d_8=512, d_9=512, d_{10}=512, h=512, o=1,10,26$.

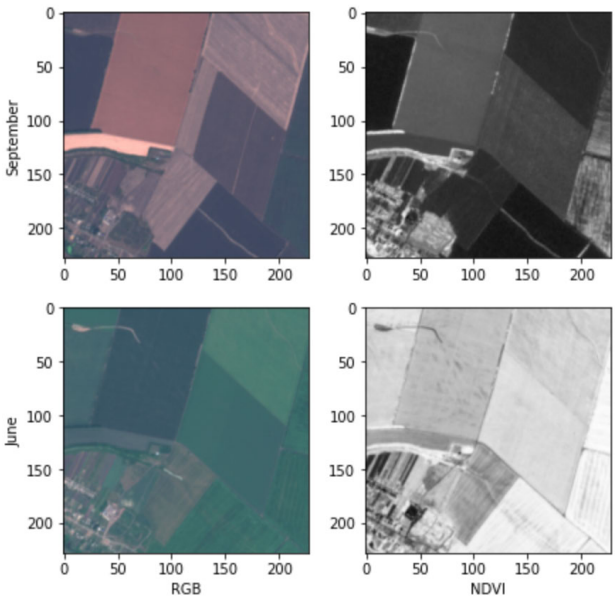

FIGURE 4. Samples of RGB and NDVI images from the dataset.

- Slide window + Deep Oblique Forest + prune: modification of the Deep Forest with CO2 Forests [7] as basic classifiers, pruning and sliding window.
- Slide window + Deep Kernel Forest + prune: modification of the Deep Kernel Forest with pruning and sliding window.
- AlexNet: deep network with 4 convolution, 3 pooling, and 2 dense layers [29].
- ResNet-11: deep network with 9 convolutional, 2 dense layers and residual connections [30].

In all the experiments we used Deep Forests with three layers. We applied a commonly recognized grid search with cross-validation technique to estimate the ensemble hyperparameters: maximum tree depth $\{4, 5, 6, 7, 8\}$, the proportion of features to be considered at each stump $\{0.08, 0.1, 0.2\}$, pruning (up to 0.9) ratio. Size of the sliding window $[8-128]$, and size of the sliding window step $[2-32]$ depending on a dataset. We also estimated decision stump regularization $C = \{100, 1000, 3000, 5000\}$, kernel parameter $gamma = \{10, 100\}$ for the Deep Kernel Forest and regularization $v = \{0.1, 1, 10, 100\}$ for the Deep Oblique Forest. We applied the grid search on sampled subsets of the original datasets because training time of Deep Forests is really long. Finally, we used the obtained hyperparameter values to perform tests on full datasets. For the neural networks the only parameter we tuned with the grid search is the dropout level $\{0.05, 0.1, 0.2\}$. All other parameters are predefined by the model architectures. Both networks have been trained from scratch on the analyzed datasets without any pre-training.

In the first experiment, we assessed the classification quality on commonly recognized datasets. We used accuracy, precision, and recall with macro-averaging to evaluate the classification quality because most studies on UCI and Cifar-10 datasets utilize those scores, so we can stay comparable with these results. We did not test Cifar-10 with Deep Oblique (CO2) Forest because of hardware limitations.

Table 2 shows that further complexification of the basic estimators in the Deep Forest without any additional regularization does not lead to any significant improvements in quality scores (see "Deep Kernel Forest" column). We believe that means the complexity of Deep Forest is pretty high, and further increasing that complexity leads to overfitting. On the other hand, adding a simple tree refinement and pruning [8] leads to notable accuracy growth. Namely, the results become comparable to the ResNet in low-resolution datasets. It is also worth noting that Deep Forests with oblique (CO2) trees show competitive results on small low-feature datasets like USPS and Letter. We believe, this is because oblique CO2 trees has better generalization than trees with non-linear kernel splitters.

The feature and sample re-generation with the sliding window approach [4] leads to significant improvement for Cifar-10 only. We believe this is because images from this dataset have higher resolution and provide more diversity in

TABLE 2. Classification scores of the tested methods.

Model	MNIST	USPS	Letter	Cifar-10
Accuracy				
AlexNet	99.1	97.4	**98.5**	73.9
ResNet-11	**99.5**	**97.9**	**98.5**	**92.1**
Deep Forest	99.2	95.9	96.3	61.8
Kernel Forest	99.1	95.8	97.4	58.0
Deep Kernel Forest	98.0	93.5	97.2	60.3
Deep Kernel Forest + prune	99.2	97.8	**98.5**	62.9
Slide window + Deep Kernel Forest + prune	99.4	**97.9**	**98.5**	63.2
Slide window + Deep Oblique Forest + prune	98.9	97.5	98.3	-
Precision				
AlexNet	99.3	97.8	98.1	74.1
ResNet-11	99.3	**98.0**	**98.3**	**92.4**
Deep Forest	99.3	96.9	95.8	62.4
Kernel Forest	99.2	97.2	96.0	58.5
Deep Kernel Forest	**99.5**	95.3	96.2	60.9
Deep Kernel Forest + prune	99.3	97.5	96.0	62.9
Slide window + Deep Kernel Forest + prune	**99.5**	97.6	96.1	63.3
Slide window + Deep Oblique Forest + prune	98.7	97.6	98.1	-
Recall				
AlexNet	99.1	95.1	96.2	74.0
ResNet-11	**99.6**	95.0	96.1	**92.1**
Deep Forest	99.3	**95.3**	89.7	61.8
Kernel Forest	99.2	95.0	91.1	58.0
Deep Kernel Forest	99.0	94.1	90.0	60.3
Deep Kernel Forest + prune	99.3	95.2	**96.9**	62.8
Slide window + Deep Kernel Forest + prune	99.4	**95.3**	**96.9**	63.1
Slide window + Deep Oblique Forest + prune	98.9	95.2	96.1	-

TABLE 3. Accuracy of Deep Forests with particular share of univariate trees.

Dataset	0.1	0.3	0.5	0.7	0.9
Deep Oblique Forest + prune					
MNIST	98.6	98.6	98.9	**99.1**	99.0
USPS	97.5	97.6	97.7	97.6	**97.8**
Letter	98.1	98.3	98.2	**98.4**	**98.4**
Deep Kernel Forest + prune					
MNIST	98.9	99.2	**99.3**	98.7	98.6
USPS	**97.9**	**97.9**	97.6	96.1	96.2
Letter	98.3	**98.4**	98.2	97.1	96.9

terms of represented objects, which means the image scaling and transforming can have much effect on classification accuracy. However, on this dataset with a relatively large feature-set, deep networks still significantly outperform Deep Forests.

In the second experiment we tested how particular shares of univariate trees in layers of Deep Kernel and Oblique Forests affect the generalization and classification accuracy. We consider a particular share of such trees as a probability that a tree from an ensemble is univariate. Therefore, when we train the ensemble tree we randomly decide if it is multivariate or parallel-axis. The univariate trees had the same maximum depth, impurity criterion, and the number of features to consider as the multivariate ones. Table 3 shows that the use of parallel-axis trees helps to slightly increase the accuracy of Deep Oblique Forests. The best results in this case are achieved if the share of those trees lies between 70-90 percent. However, in the case of forests with non-linear (gaussian) kernels we detected an improvement for MNIST dataset only. Optimal share of univariate trees for Deep Kernel Forests in that case lies between 0 and 50 percent depending on the dataset.

TABLE 4. MSE scores of the NDVI prediction.

Model	Spring	Summer/Autumn	All
Deep Forest	0.006	0.007	0.007
Deep Oblique Forest + prune	0.004	0.005	0.005
Deep Kernel Forest + prune	0.004	**0.004**	**0.004**
AlexNet	0.006	0.007	0.006
ResNet-11	**0.003**	0.006	0.005

In the third experiment we assessed the quality of NDVI prediction with the Deep Forest, Deep Kernel and Oblique Forests (with pruing and sliding window), AlexNet and ResNet neural networks [29]. In the experiments we evaluated mean squared error (MSE) of the predictions. Results from Table 4 show that Deep Kernel Forest can predict the NDVI level more accurately than the original Deep Forest or AlexNet network model. Also, we did not detect any dramatic differences for NDVI evaluation in "Spring" and "Summer/Autumn" subsets with AlexNet, but in general the obtained scores are worse than for the Deep Forests. We believe this is because first of all, the season-based division we used to separate immature plants from senescent ones is pretty fuzzy. In [1] they also use larger scaled UAV images, when non-linear dependencies between RGB levels and vegetation are easier to reveal. In contrast, a more complex ResNet shows the best results for the "Spring" subset, but the MSE for the "Summer/Autumn" subset is significantly lower, as in [1]. In general, the obtained NDVI prediction error is pretty low, although we did not perform any complex image pre-processing. Therefore, the proposed modified models can be applied to assess NDVI in practical software applications for precise farming.

VI. TRAINING TIME

We assessed the time needed to train the original and modified Deep Forests and AlexNet neural network. We set the ensemble layer number to 3, forest size to 30 trees and tree depth to 4 for all the forest-based approaches. All the experiments were performed on a computer with the following hardware: 12-Core CPU AMD Ryzen Threadripper 1920, 256 GB RAM, Nvidia GeForce RTX 3090Ti.

We randomly sampled batches of fixed size (from 1K to 10K) from MNIST dataset and estimated the training time (Fig. 5). We used the stochastic gradient descent method from Scikit-Learn to train Deep Oblique Forest, dual LibSVM solver to train Deep Kernel Forest (gaussian kernel) on CPU, and ThunderSVM solver to train Deep Kernel Forest on GPU. The obtained scores agree with the theoretical results of complexity analysis. We revealed that most of the considered approaches show linear dependence from the dataset size for a fixed tree depth. However, Deep Forests with non-linear splits

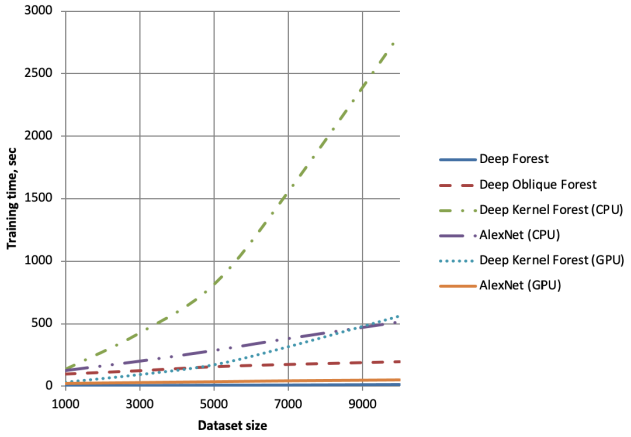

FIGURE 5. Training time of Deep Forests and AlexNet on subsamples of MNIST dataset.

show a polynomial trend both in GPU and CPU. The results show that further research in dual SVM solvers is required to make Deep Kernel Forests with non-linear kernels trainable on large datasets.

VII. DISCUSSION

The experiments show that the proposed modifications of Deep Forests improve quality for both classification and regression tasks. Besides, they can model non-smooth dependencies more accurately than considered neural networks. On the one hand, accurate and informative feature representation generation is a cornerstone of cascade models. Each layer of Deep Forest for each data sample encodes a feature subspace related to this sample. Kernel Forest detects more homogeneous subspaces than Random Forest and considers complex feature relationships (Fig. 2), while Ren's refinement approach helps directly improve those feature representations via optimization of a global loss [8]. On the other hand, algorithms to generate multivariate tree ensembles have significantly low training speed [26], which remains an open problem. Despite the use of more complex basic classifiers, Deep Forests still shows lower classification accuracy for data with high-dimensional feature sets like Cifar-10 than deep networks. Therefore, further research related to enforcing Deep Forests without introducing additional complexity like [20] is required.

VIII. CONCLUSION

The paper presents a modified Deep Forest that combines the Kernel and Oblique (CO2) Forest model and random forest refinement technique. In this modification, we use forests of multivariate trees to enforce discriminative power of the model at each level, and apply the refinement technique to generate informative embeddings. Experiments on commonly recognized image classification datasets show that the proposed method significantly outperforms the original Deep Forest. It also provides results similar to ResNet-11 in the data with a small feature-set. Tests on the RGB-NDVI datasets confirm that the proposed methods outperform neural network-based models and form accurate predictions for immature and senescent plants. In terms of training complexity, the proposed models are more effective than neural-network based models in data with a small feature-set. Besides, the training process of the Deep Forest can be effectively paralleled at the level of decision stamps. Therefore, the training can be performed on distributed heterogeneous cloud clusters temporarily built to run particular task, which is cheaper and more approachable for small farms or companies than the use of GPU cards in case of neural networks. In practice, this possibility becomes more important in case of modern GPU deficiency and high cost.

We believe the proposed combination of multi-layer Deep Forests and refined multivariate forests can be helpful in many practical tasks where frequency bias effect in the training process of neural networks prevent them from revealing required non-smooth dependencies without complex feature pre-processing [2]. Those tasks vary from obtaining vegetation indices and yield forecasting to MRI reconstruction and inverse 3D rendering [3].

The remaining issue of Deep Kernel Forests is that the multi-grained scanning procedure leads to an exponential growth of the training dataset. In the future, we will try to develop an online modification of the proposed method to tackle that problem. We are also going to manually verify the immature/senescent split in our RGB-NDVI dataset, so it can be used as gold data in future research.

REFERENCES

[1] Z. Khan, V. Rahimi-Eichi, S. Haefele, T. Garnett, and S. J. Miklavcic, "Estimation of vegetation indices for high-throughput phenotyping of wheat using aerial imaging," *Plant Methods*, vol. 14, no. 1, pp. 1–11, Dec. 2018, doi: 10.1186/s13007-018-0287-6.

[2] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, and A. Courville, "On the spectral bias of neural networks," in *Proc. ICML*, Long Beach, CA, USA, 2019, pp. 5301–5310.

[3] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, and U. Singhal, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proc. NeurIPS*, Vancouver, BC, Canada, vol. 33, 2020, pp. 7537–7547.

[4] Z. H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," in *Proc. IJCAI*, Melbourne, VIC, Australia, 2017, pp. 3553–3559.

[5] Y.-H. Chen, S.-H. Lyu, and Y. Jiang, "Improving deep forest by exploiting high-order interactions," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Auckland, New Zealand, Dec. 2021, pp. 1030–1035.

[6] B. B. Yang, S. Q. Shen, and W. Gao, "Weighted oblique decision trees," in *Proc. AAAI*, Honolulu, HI, USA, 2019, pp. 5621–5627.

[7] M. Norouzi, M. D. Collins, D. J. Fleet, and P. Kohli, "CO_2 forest: Improved random forest by continuous optimization of oblique splits," 2015, *arXiv:1506.06155*.

[8] S. Ren, X. Cao, Y. Wei, and J. Sun, "Global refinement of random forest," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 723–730.

[9] Y. Wang, X. Wang, and J. Jian, "Remote sensing landslide recognition based on convolutional neural network," *Math. Problems Eng.*, vol. 2019, pp. 1–12, Sep. 2019, doi: 10.1155/2019/8389368.

[10] J. Zhao, A. Kumar, B. N. Banoth, B. Marathi, P. Rajalakshmi, B. Rewald, and W. Guo, "Deep-learning-based multispectral image reconstruction from single natural color RGB image—Enhancing UAV-based phenotyping," *Remote Sens.*, vol. 14, no. 5, p. 1272, 2022, doi: 10.3390/rs14051272.

[11] Z. Shi, C. Chen, Z. Xiong, D. Liu, and F. Wu, "HSCNN+: Advanced CNN-based hyperspectral recovery from RGB images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Long Beach, CA, USA, Jun. 2018, pp. 939–947.

[12] L. Wang, Y. Duan, L. Zhang, T. U. Rehman, D. Ma, and J. Jin, "Precise estimation of NDVI with a simple NIR sensitive RGB camera and machine learning methods for corn plants," *Sensors*, vol. 20, no. 11, p. 3208, Jun. 2020, doi: 10.3390/s20113208.

[13] L. El Hoummaidi, A. Larabi, and K. Alam, "Using unmanned aerial systems and deep learning for agriculture mapping in Dubai," *Heliyon*, vol. 7, no. 10, Oct. 2021, Art. no. e08154, doi: 10.1016/j.heliyon.2021.e08154.

[14] Y.-S. Lee, S. Lee, and H.-S. Jung, "Mapping forest vertical structure in Gong-Ju, Korea using Sentinel-2 satellite images and artificial neural networks," *Appl. Sci.*, vol. 10, no. 5, p. 1666, Mar. 2020.

[15] Z. Dai, "A new method of LncRNA classification based on ensemble learning," in *Proc. J. Phys., Conf.*, 2021, vol. 1994, no. 1, Art. no. 012002, doi: 10.1088/1742-6596/1994/1/012002.

[16] J. Tang, H. Xia, J. Zhang, J. Qiao, and W. Yu, "Deep forest regression based on cross-layer full connection," *Neural Comput. Appl.*, vol. 33, no. 15, pp. 9307–9328, Aug. 2021, doi: 10.1007/s00521-021-05691-7.

[17] S. H. Lyu, L. Yang, and Z.-H. Zhou, "A refined margin distribution analysis for forest representation learning," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2019, pp. 5530–5540.

[18] J. Ren, B. Hou, and Y. Jiang, "Deep forest for multiple instance learning," *J. Comput. Res. Develop.*, vol. 56, no. 8, pp. 1670–1676, 2019.

[19] L. Yang, X.-Z. Wu, Y. Jiang, and Z.-H. Zhou, "Multi-label learning with deep forest," 2019, *arXiv:1911.06557*.

[20] L. V. Utkin, A. V. Konstantinov, V. S. Chukanov, and A. A. Meldo, "A new adaptive weighted deep forest and its modifications," *Int. J. Inf. Technol. Decis. Making*, vol. 19, no. 4, pp. 963–986, Jul. 2020, doi: 10.1142/S0219622020500236.

[21] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[22] W. Liu, H. Lin, L. Huang, L. Peng, T. Tang, Q. Zhao, and L. Yang, "Identification of miRNA–disease associations via deep forest ensemble learning based on autoencoder," *Briefings Bioinf.*, vol. 23, no. 3, pp. 1–12, 2022, doi: 10.1093/bib/bbac104.

[23] L. V. Utkin and K. D. Zhuk, "Improvement of the deep forest classifier by a set of neural networks," *Informatica*, vol. 44, no. 1, pp. 1–13, Mar. 2020, doi: 10.31449/inf.v44i1.2740.

[24] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, no. 9, pp. 15–64, 2005.

[25] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.

[26] D. A. Devyatkin and O. G. Grigoriev, "Random kernel forests," *IEEE Access*, vol. 10, pp. 77962–77979, 2022, doi: 10.1109/ACCESS.2022.3193385.

[27] L. Breiman, J. H. Friedman, and R. A. Olshen, *Classification and Regression Trees*. Belmont, CA, USA: Wadsworth, 1984.

[28] *Deep Kernel/Oblique Forests Implementation*. Accessed: Mar. 23, 2023. [Online]. Available: https://github.com/masterdoors/kernel_trees/

[29] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV*, Amsterdam, The Netherlands, 2016, pp. 630–645.

[32] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2003, pp. 1–8.

[33] Z. Ming, J. Chazalon, M. M. Luqman, M. Visani, and J.-C. Burie, "Simple triplet loss based on intra/inter-class metric learning for face verification," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 1656–1664.

[34] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, Long Beach, CA, USA, 2017, pp. 1–11.

[36] A. Sagheer, M. Zidan, and M. M. Abdelsamea, "A novel autonomous perceptron model for pattern classification applications," *Entropy*, vol. 21, no. 8, p. 763, Aug. 2019, doi: 10.3390/e21080763.

[37] P. M. Murphy and D. W. Aha, "UCI repository of machine learning databases," Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 1991.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[39] *Plants vs CO_2*. Accessed: Mar. 23, 2023. [Online]. Available: https://github.com/GrHalbgott/Plants-vs-CO2

[40] *Copernicus Web Platform*. EU Space Agency/EU Commission Copernicus. Accessed: Mar. 23, 2023. [Online]. Available: https://scihub.copernicus.eu

[41] *RGB NDVI Image Dataset*. Accessed: Mar. 23, 2023. [Online]. Available: http://keen.isa.ru/ndvi

DMITRY A. DEVYATKIN received the B.S. and M.S. degrees in computer science from Rybinsk State Aviation Technology University, Rybinsk, Russia, in 2011. He is currently pursuing the Ph.D. degree in computer science with the Federal Research Center "Computer Science and Control," RAS, Moscow, Russia.

Since 2011, he has been a Researcher with the Russian Artificial Intelligence Research Institute, Federal Research Center "Computer Science and Control," RAS. His research interests include machine learning, randomized ensembles, natural language processing, information extraction, and information retrieval.

Mr. Devyatkin received several awards and honors, including the Best Paper Award from the IEEE 8th International Conference on Intelligent Systems (IS).