

Vladyslav YALOVEHA, Andrii PODOROZHNIAK, Heorhii KUCHUK

*National Technical University "Kharkiv Polytechnic Institute", Ukraine*

## CONVOLUTIONAL NEURAL NETWORK HYPERPARAMETER OPTIMIZATION APPLIED TO LAND COVER CLASSIFICATION

*In recent times, machine learning algorithms have shown great performance in solving problems in different fields of study, including the analysis of remote sensing images, computer vision, natural language processing, medical issues, etc. A well-prepared input dataset can have a huge impact on the result metrics. However, a correctly selected hyperparameter combined with neural network architecture could highly increase the final metrics. Therefore, the hyperparameters optimization problem becomes a key issue in a deep learning algorithm. The process of finding a suitable hyperparameter combination could be performed manually or automatically. Manual search is based on previous research and requires enormous human efforts. However, there are many automated hyperparameter optimization methods have been successfully applied in practice. The automated hyperparameter tuning techniques are divided into two groups: black-box optimization techniques (such as Grid Search, Random Search) and multi-fidelity optimization techniques (HyperBand, BOHB). The most recent and promising among all approaches is BOHB which, which combines both Bayesian optimization and bandit-based methods, outperforms classical approaches, and can run asynchronously with given GPU resources and time budget that plays a vital role in the hyperparameter optimization process. The previous study proposed a convolutional deep learning neural network for solving land cover classification problems in the EuroSAT dataset. It was found that adding spectral indexes NDVI, NDWI, and GNDVI with RGB channels increased the result accuracy (from 64.72% to 84.19%) and F1 (from 63.89 % to 84.05%) score. However, the convolutional neural network architecture and hyperparameter combination were selected manually. The re-search optimizes convolutional neural network architecture and finds suitable hyperparameter combinations applied to land cover classification problems using multispectral images. The obtained results must increase result performance compared with the previous study and given budget constraints.*

**Keywords:** *hyperparameter optimization; EuroSAT; BOHB; convolutional neural network; land cover; remote sensing.*

### Introduction

In recent years machine learning algorithms are widely used in different areas. The deep learning solutions show high performance in robotics and autonomous vehicle control, text recognition [1, 2], automatic license plate recognition [3], natural language processing [4], neuroscience research [5], applications in computer vision [6], remote sensing problems [7, 8], medicine issues [9, 10]. Deep learning (DL) is a part of machine learning algorithms that is applied to various areas, showed high performance solving many types of problems and based on the theory of artificial neural networks [11].

In general, building an effective machine learning model is a complex and time-consuming process that involves determining the appropriate algorithm and obtaining an optimal model architecture by tuning its hyperparameters [12]. In practice, it is needed to search continuously, apply different sets of values, retrain models, analyze, and compare result metrics to get the best model. Therefore, hyperparameters optimization prob-

lem (HPO) becomes a key issue in a machine learning algorithm.

Compared with other machine learning algorithms, deep learning models usually have many hyperparameters that must be tuned. And their optimization plays a vital role in the prediction accuracy of algorithms [13]. Finding good external artificial neural network parameters could highly increase result metrics in a reasonable amount of time. On the other hand, because of the huge number of parameters, the issue becomes more complex, and it is unclear to understand how the parameters interact with each other to affect the final performance. Since all neural network models have an input layer, and an output layer, the complexity of a deep learning model mainly depends on the number of layers and the number of neurons of each layer, which are two main hyperparameters to build deep learning models [14].

The research aims to optimize convolutional neural network (CNN) architecture and find suitable hyperparameters combination applied to land cover classification problem using multispectral images. The obtained results must increase result performance in comparison with the previous study [1] and given budget constraints.

## 1. Hyperparameter Optimization Problem

Hyperparameter optimization methods are used to optimize the architecture and internal parameters of a deep learning model. These methods provide evaluation of the optimal hyperparameter combinations from given search space configuration within the predefined budgets [15].

For given input data  $X$ ,  $n$  hyperparameters  $\{n_1, n_2 \dots n_m\}$  with algorithm  $A$  the hyperparameter problem aims to find an optimal configuration of  $n$  hyperparameters, which maximizes the performance of  $A$  in  $X$  [16]. The optimization problems can be classified as constrained or unconstrained optimization problems based on whether they have constraints for the decision variables or the solution variables [12].

Hyperparameters in deep learning take different values and constraints (for instance, the number of neural network layers and the number of neurons in dense layers usually have different degree order). So, the optimization problem in the research is defined as maximization constraint one and can be expressed by the next equation:

$$n' = \arg \max_{n \in R} f(n),$$

$$R = \{n \in N \mid g_i(n) \leq 0, h_j(n) = 0\},$$

where  $f(n)$  is the objective function,  $n'$  is the hyperparameter configuration that produces the value of  $f(n)$ ,  $R$  is the feasible region that limits possible values from the configuration search space  $S$ ,  $g_i(n), i=1, 2, \dots, k$  are the inequality constraint functions,  $h_j(n)=0, j=1, 2, \dots, l$  are the equality constraint function,  $N$  is the domain of  $n$ . The mathematical expression of the function  $f$  varies, depending on the objective function of the chosen deep learning algorithm and the result metric. On the other hand, in practice, time budgets are an essential constraint for optimizing models and must be considered. It often requires a massive amount of time to optimize the objective function of a model with a reasonable number of hyperparameter configurations [12].

## 2. Hyperparameter Optimization Methods

The process of finding suitable hyperparameters could be done manually or automatically. Manual search is based on previous results, guessing, and data scientist experience. It goes through different neural

network architectures and input setups until some predefined stopping criterion. But manual tuning is ineffective for many problems due to certain factors, including many hyperparameters, complex models, time-consuming model evaluations, and non-linear hyperparameter interactions. These factors have inspired increased research in techniques for automatic optimization of hyperparameters and make it possible for users to apply machine learning models to practical problems effectively [12, 15].

In principle, the automated hyperparameter tuning techniques can be classified into two main categories: black-box optimization techniques and multi-fidelity optimization techniques [15]. The most widely known and used optimization approaches for deep learning issues are considered below.

### 2.1. Black-box Optimization Approaches

#### 2.1.1. Grid Search

The main idea of the Grid Search algorithm is based on setting a grid of search space configuration, evaluating all possible combinations, and selecting the best one. The solution is easy to implement and parallelize. However, a grid experiment with a fine-enough resolution for optimization would be prohibitively expensive [17] and it does not scale well for large configuration spaces, as the number of trails grows exponentially with the number of hyperparameters [18]. And the next experiment doesn't use previous knowledge to configure the search space to find better hyperparameters configuration. This could lead to many redundant combinations. The computational complexity of the algorithm increases exponentially, so it is suitable when we have optimization problem with strict constraints and search space is small.

#### 2.1.2. Random Search

Another widely known alternative to Grid Search is Random Search. As the name implies, the trial configuration in random search is generated by selecting hyperparameters independently and randomly. Random search is also effortless to implement and could be run asynchronously, which is important for big amounts of data and deep learning issues (such as images classification) and provides better solutions than grid search.

In [17] it is shown that random experiments are more efficient than grid experiments for hyperparameter optimization in the case of several learning algorithms on several datasets, and its computational complexity is  $O(n)$  [19]. On the other hand, as function evaluations are very expensive, random search requires a long optimization period.

### 2.1.3. Bayesian Optimization

Bayesian optimization is a state-of-the-art optimization framework for the global optimization of expensive black-box functions, which recently gained traction in HPO by obtaining new results in tuning artificial neural networks for image classification, speech recognition, and language modeling [20]. Bayesian optimization consists of two main components which are surrogate models for modeling the objective function and an acquisition function that measures the value that would be generated by the evaluation of the objective function at a new point [15]. A probabilistic model of the objective function  $f$  is obtained using Bayes' theorem:

$$P(f | D_{1:n}) \propto P(D_{1:n} | f)P(f),$$

where  $D_{1:n}$  – hyperparameters configurations,

$f$  – objective function.

Gaussian processes have become the standard surrogate for modeling the objective function in Bayesian optimization [21]. One of the main limitations of the Gaussian processes is the runtime complexity of  $O(n^3)$  the number of data points which limits their parallelization capability [17].

## 2.2. Multi-fidelity Optimization Approaches

Increasing size of the input data, number of internal and external neural network parameters make black-box performance evaluation harder because it requires more time and computational resources. Training a combination of hyperparameters could easily exceed from hours to even several days [22].

A widely used technique to speed up such optimization is to probe a selected combination of hyperparameters on a subset of the input data and train it only for a few iterations. Multi-fidelity methods cast such manual heuristics into formal algorithms, using so-called low fidelity approximations of the actual loss function to minimize [20]. Bandit-based algorithms have shown strong performance, especially for optimizing deep learning algorithms [23].

### 2.2.1. HyperBand

HyperBand is a bandit-based powerful multi-fidelity algorithm. While recent approaches use Bayesian optimization to adaptively select configurations, Hyperband focuses on speeding up random search through adaptive resource allocation and early stopping. The

technique is formulated as a pure-exploration on stochastic infinite-armed bandit problem where a predefined resource like iterations, data samples, or features is allocated to randomly sampled configurations [23].

The algorithm divides the given total budget  $B$  into  $k$  pieces and allocates them to each configuration  $b = B/k$ . Then, it calls successive halving technique [24] on each random sample configuration. Hyperband shows great success with deep neural networks and performs better than random search and Bayesian optimization [20]. The algorithm complexity is  $O(n \log n)$ .

### 2.2.2. BOHB

Authors of the recent optimization algorithm BOHB [25] propose to combine both Bayesian optimization and bandit-based methods, to achieve strong anytime performance and fast convergence to optimal configurations. BOHB can run asynchronously and uses the resources effectively. It outperforms both Bayesian optimization and Hyperband on a wide range of problem types, including high-dimensional toy functions, support vector machines, feed-forward neural networks, Bayesian neural networks, deep reinforcement learning, and convolutional neural networks [25].

For a given budget, BOHB relies on the Hyperband algorithm to determine how many parameters combinations to evaluate with it. Despite Hyperband at the beginning of the trial, BOHB uses a model-based search instead of the random selection of configurations. Once the desired number of configurations for the iteration is reached, the standard successive halving procedure is carried out using these configurations. Authors keep track of the performance of all function evaluations  $g(x, b) + \epsilon$  of configurations  $x$  on all budgets  $b$  to use as a basis for models in later iterations. BOHB's Bayesian optimization component resembles the Tree-Structured Parzen estimator [26] but differs by using multidimensional kernel density estimators [20]. The computational complexity of BOHB is  $O(n \log n)$ .

The algorithm was successfully applied to hyperparameters optimization of support vector machines, feed-forward neural networks, Bayesian neural networks, deep reinforcement learning agents, and convolutional neural networks. For tuning ConvNet the CIFAR-10 dataset is used and achieved a test error of  $2.78\% \pm 0.09\%$  which is better than similar research [25].

Among all considered optimization approaches above the BOHB algorithm was selected for experiments due to its high performance, comparative computational complexity, and availability of asynchronous execution.

### 3. Experiments Setup

#### 3.1. Previous Work

Modern artificial neural networks are used for solving remote sensing problems. The remotely sensed images usually have more complicated and diverse patterns, thus higher requirements are imposed on the processing methods of them [27]. However, deep learning automates the process of finding valuable feature representation on complex data. It is applied in many aspects, including land cover and classification [28], agriculture yield prediction [29] and analysis [30], crop-types classification [31], detecting fire hazardous forest areas [32].

In [1] was proposed simple convolutional deep learning neural network (see Fig. 1) for solving land cover classification problem on the EuroSAT dataset [33]. It was found that adding spectral indexes (NDVI, NDWI, GNDVI) as additional features with RGB channels could highly increase result accuracy (from 64.72% in baseline approach to 84.19%) and F1 (from 63.89 % to 84.05%) score. But in the research the deep learning model and hyperparameter combination were selected manually. In the current work, we focus on increasing classification metrics by conducting hyperparameters optimization.

The proposed baseline model in Fig. 1 consists of sequences of convolutional and max-pooling layers. For all next experiments, we chose  $\text{ReLU}(x) = \max(0, x)$  activation function among neural network layers. At the final stage, the softmax activation function is defined as:

$$s(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{10} e^{x_j}},$$

where  $\bar{x} = (x_0 \dots x_{10})$  and  $x_i$  represents  $i$ -th class in EuroSAT dataset classes. After each max-pooling layer BatchNormalization layer is added for preventing overfitting problem.

#### 3.2. Search Space Configuration

The hyperparameters in models are divided into two types: structure-related parameters (rough-tuning) and learning-related ones (fine-tuning). These hyperparameters must be set before training a model configuration.

For searching suitable hyperparameters we must define the search space configuration. After we will search through the space of parameters, assessing the performance of each neural network by training it until some stopping criterion (depending on the budget available) [34].

Tables 1 and 2 below highlight the overall search space configuration and hyperparameters range for the convolutional neural network optimization problem considered in the study.

#### 3.3. Optimization Tools

For carrying out experiments, we used Ray Tune [35]. It is a unified framework for model selection and training that provides a narrow-waist interface between training scripts and search algorithms. A trial is defined as a single training run with a fixed initial hyperparameter configuration. An experiment is a collection of trials supervised by Tune using one of its trial scheduling algorithms. During the search, many trials are evaluated in parallel. Hyperparameter search algorithms examine trial results in sequence and make decisions that affect the parallel computation. The Ray framework provides the underlying distributed execution and resource management. Each trial in Tune runs in its Python process and can be allocated a given number of CPU and GPU resources through Ray [33]. The Ray Tune execution pipeline is shown in Fig. 2.

As the integrated development environment (IDE) Google Collaboratory was selected. It is a project that has the objective of disseminating machine learning education and research. It is free of use and based on Jupyter Notebook. The environment provides either Python runtimes with the essential deep learning libraries (such as TensorFlow, Keras etc.) and GPU support for parallel computing that is considered to run the training process in a feasible time [36].

#### 3.4 Research Experiments Pipeline

The process of searching hyperparameters includes an estimator, a search space, an optimization method (BOHB) for hyperparameter combinations, and an evaluation function (validation accuracy) to compare results obtained with different sets of parameters.

For all experiments next resources and constraints were used:

$$\begin{cases} T_b = 10 \cdot 3600, \\ N'_{iter} = 500, \\ N'_s = 1500, \\ Acc'_{val} = 0.99 \end{cases}$$

where  $T_b$ , – time budget (in seconds) per iteration,  $N'_{iter}$ , – max number of training iterations,  $N'_s$  – max number of samples,  $Acc'_{val}$  – max validation accuracy. The experiments pipeline is shown in Fig. 3.

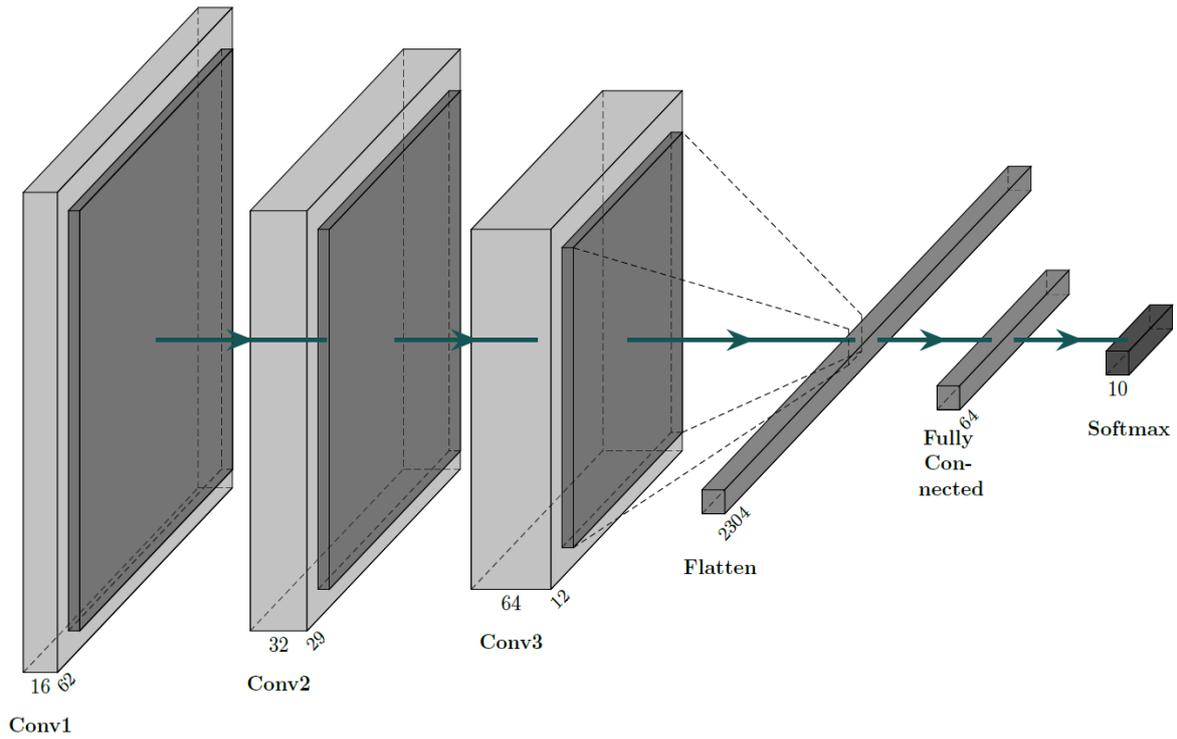


Fig. 1. Baseline convolutional neural network architecture

Table 1

Neural network architecture-related search space configuration

Hyperparameter	Range
Conv filters	{16, 32, 64, 128, 256, 512}
Conv layers	{1, 2, 3, 4, 5}
Dense layers	{1, 2}
Dense units	[128; 2048]
Dropout rate	{0.0, 0.1, ..., 0.5}

Table 2

Neural network learning-related search space configuration

Hyperparameter	Range
Learning rate	[1e-10; 0.1]
Batch size	{25, 50, 75, 100}
Validation split	{0.1, 0.15, ..., 0.4}
l2 regularization	[1e-10; 0.1]
Optimizer	{Adam, RMSprop, SGD}

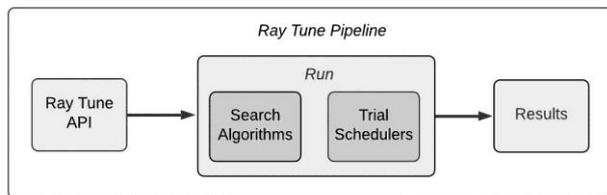


Fig. 2. Ray Tune execution pipeline

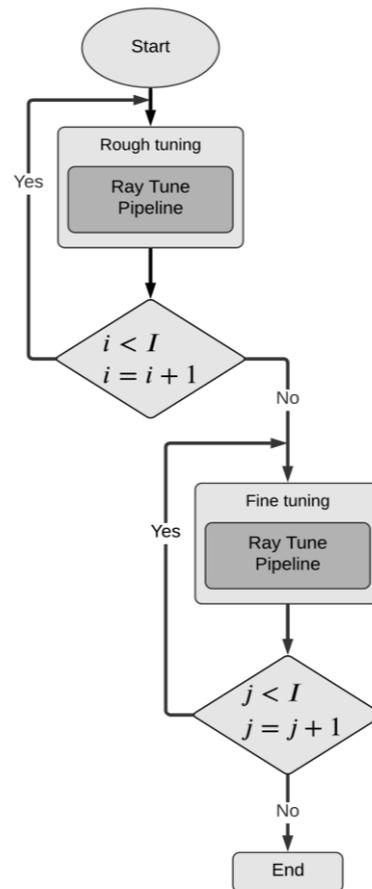


Fig. 3. Research experiments pipeline (i, j = 0, I = 2)

### 4. Results and Discussion

#### 4.1. Rough-tuning stage

On the rough-tuning stage, the search space configuration for convolutional neural network architecture was selected from Table 1, and learning-related parameters were taken and fixed from the baseline model (see Table 3).

Table 3

Fixed learning-related hyperparameters on rough stage

Hyperparameter	Value
Learning rate	$1e-6$
Batch size	50
Validation split	0.2
l2 regularization	Not used
Optimizer	RMSprop

After each iteration we filter hyperparameters configurations by the next criterion:

$$\begin{cases} |Acc_t - Acc_{val}| \leq \epsilon; \\ Acc_t \geq \delta, \end{cases} \quad (1)$$

where  $Acc_t$  – train accuracy of  $k$ -th iteration,  $Acc_{val}$  – validation accuracy of  $k$ -th iteration,  $\epsilon$  and  $\delta$  – thresholds.

Filtering trials is needed to remove unpromising hyperparameters configurations, reduce configuration space and general search complexity.

For first iteration and given time budget  $T_b$  the total number of BOHB training trials were 1016. After applying conditions (1) with  $\epsilon=0.1$  and  $\delta=0.85$  14 best configurations left. The filtered training trials and accuracy values on each epoch are illustrated in Fig. 4. For visualization high-dimensional or multivariate data it is convenient to use parallel plots. For each of  $n$  dimension the parallel lines are drawn that vertically and equally spaced and considered as axes. High-dimensional plot for current iteration is presented in Fig. 5.

As we can see from the high-dimensional plot in Fig. 5 the input range for the 4-th convolutional filter and the hidden neurons range in dense layers could be decreased. The final search space configuration for the next iteration is shown in Table 4. The current iteration didn't decrease the input space significantly.

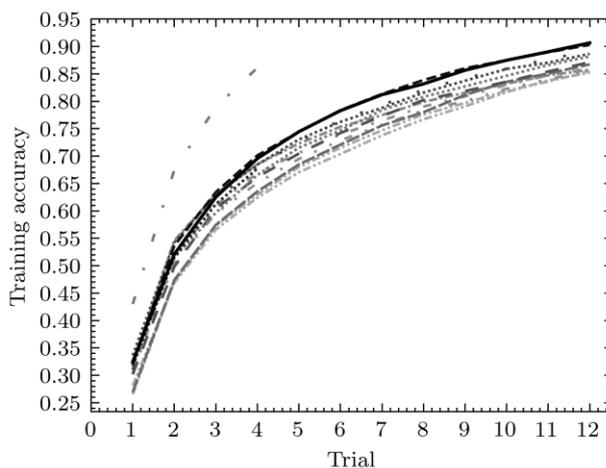


Fig. 4. Best rough-tuning filtered training trials,  $i = 0$

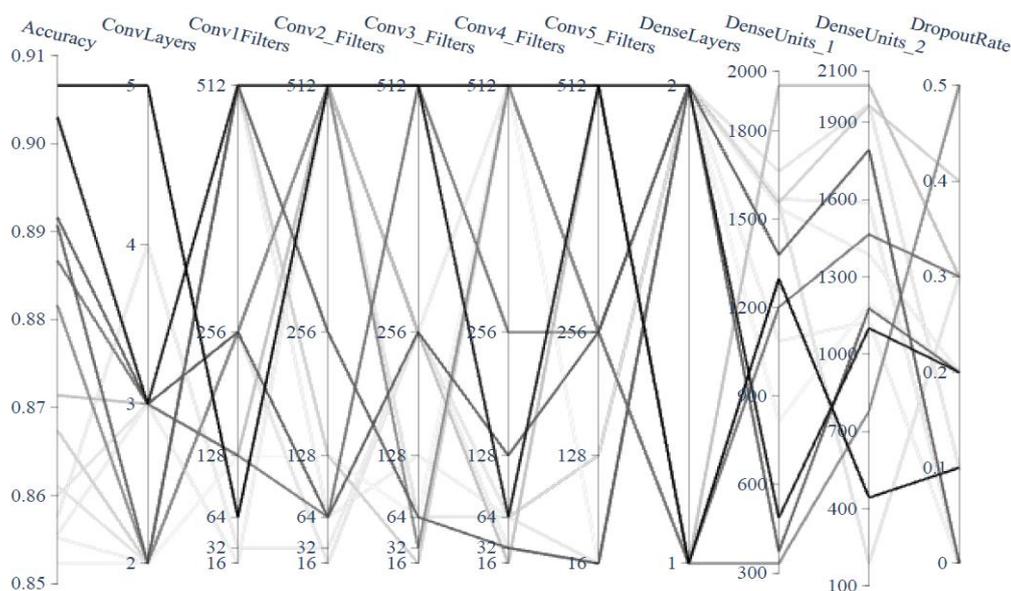


Fig. 5. High-dimensional trials plot,  $\epsilon = 0.1$ ,  $\delta = 0.85$

Table 4

Architecture-related search space configuration,  $i = 1$

Hyperparameter	Value
Conv filters	
• conv 1	• {16, 32, 64, 128, 256, 512}
• conv 2	• {16, 32, 64, 128, 256, 512}
• conv 3	• {16, 32, 64, 128, 256, 512}
• conv 4	• {16, 32, 64, 128, 256, 512}
• conv 5	• {16, 128, 256, 512}
Conv layers	{1, 2, 3, 4, 5}
Dense layers	{1, 2}
Dense units	
• layer 1	• [333; 1953]
• layer 2	• [188; 1969]
Dropout rate	{0.0, 0.1, 0.2, 0.3, 0.4, 0.5}

After the second iteration among 1015 training trials, 5 best configurations were filtered according to (1). Fig. 6 and 7 show the results of the experiment. As we can observe from the high-dimensional plot the number of dense layers decreased from two to one, and for the current problem, the suitable number of convolutional layers is three. Furthermore, we got a single value of 512 for filters in the third convolutional layer. As we reached the stopping criterion ( $i = 1$ ) and the time budget limit the network-related hyperparameters are selected from the best trial. The result training accuracy is 0.9227 (see Fig. 6) and validation accuracy is 0.8373.

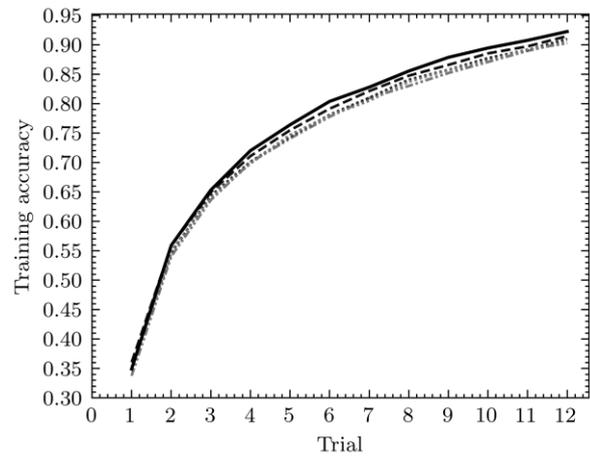


Fig. 6. Best rough-tuning filtered training trials,  $i = 1$

Final optimized convolutional neural network architecture is presented in Fig. 8.

### 4.2. Fine-tuning stage

On fine-tuning stage, we selected the search space configuration from Table 2 and fixed optimized neural network architecture (see Fig. 8). The first fine-tuning experiment has consisted of 23 trials ( $\epsilon = 0.05$  and  $\delta = 0.9$ ). Fig. 9-10 show the training process for 4 best learning-related hyperparameters combinations after filtering.

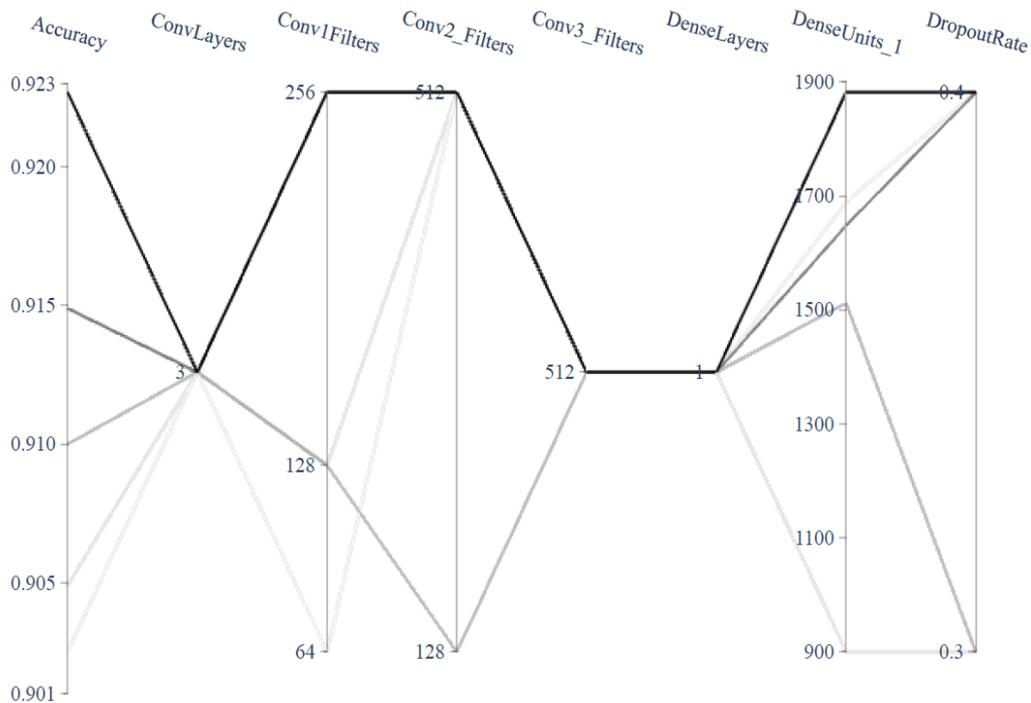


Fig. 7. High-dimensional trials plot ( $\epsilon = 0.1$ ,  $\delta = 0.9$ )

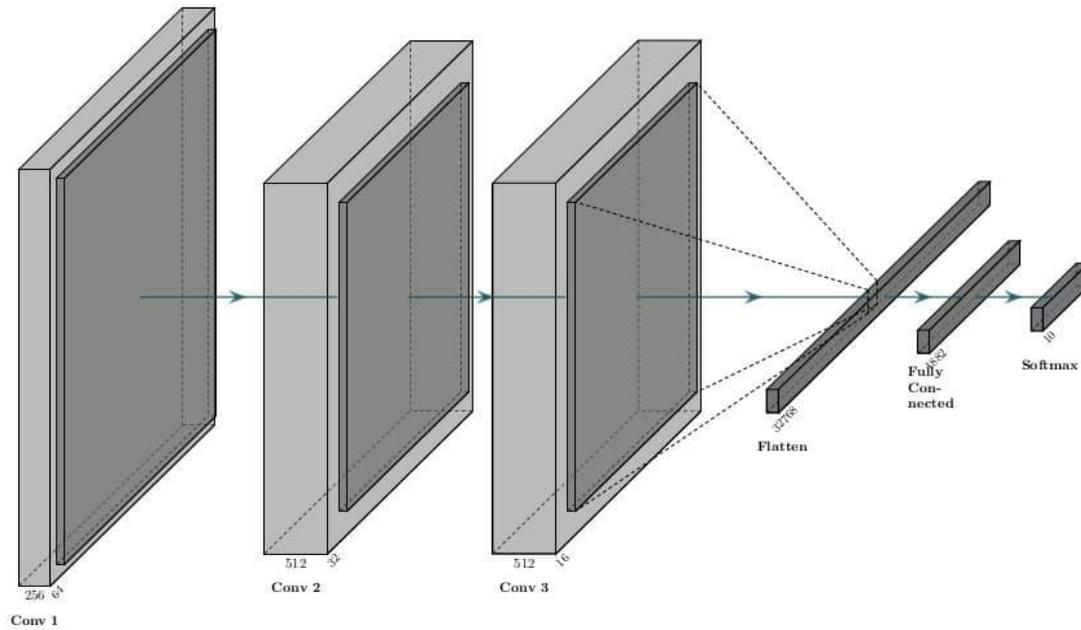


Fig. 8. Optimized convolutional neural network architecture

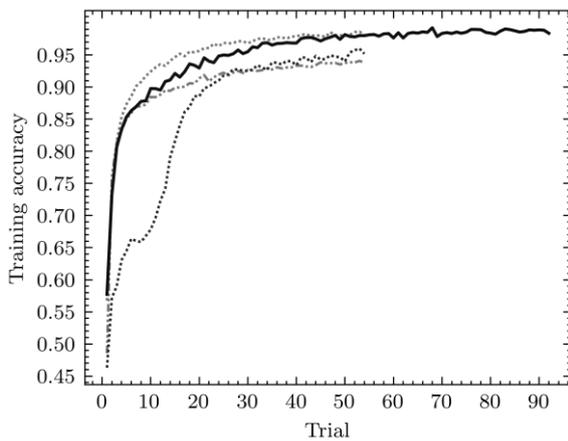


Fig. 9. Best fine-tuning filtered training trials,  $j = 0$

An analysis of the graph in Fig. 10 illustrates the dependencies of hyperparameters values for each trial. The dimension of search space configuration can be decreased due to validation split, batch size, and continuous hyperparameters (learning rate and l2 regularization). Narrowed learning-related search space configuration is shown in Table 5. In comparison with the architecture-related optimization stage, the iteration shows promising results (training accuracy reaches a value of 0.95). During the second iteration ( $j = 1$ ) among 23 executed trials, one is left (see Fig. 11). The result training accuracy is 0.983 and validation accuracy is 0.963. Table 6 highlights final optimized learning-related hyperparameters for the convolutional neural network.

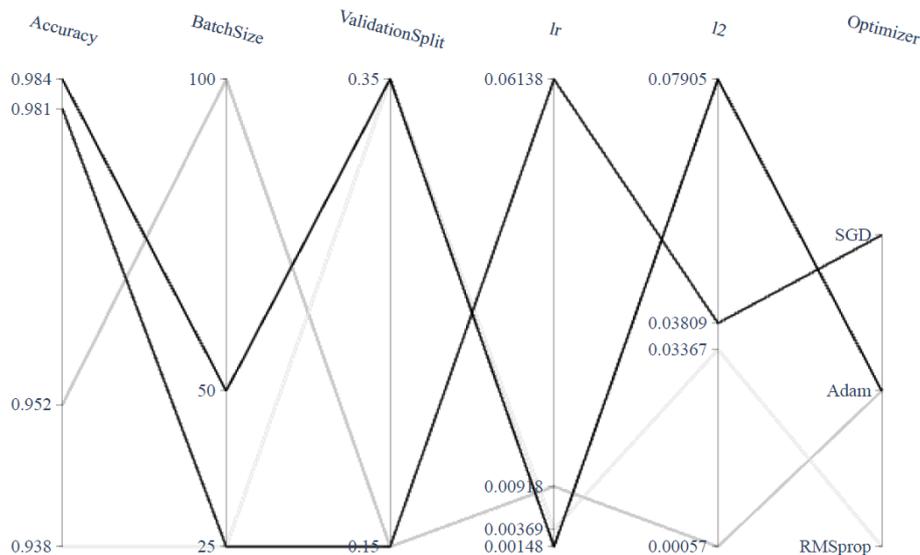


Fig. 10. High-dimensional trials plot,  $\epsilon = 0.05$ ,  $\delta = 0.9$

Table 5

Learning-related search space configuration,  $j = 1$

Hyperparameter	Value
Learning rate	[0.00148; 0.06138]
Batch size	{25, 50, 100}
Validation split	{0.15, 0.35}
$l_2$ regularization	[0.00057; 0.07905]
Optimizer	{Adam, RMSprop, SGD}

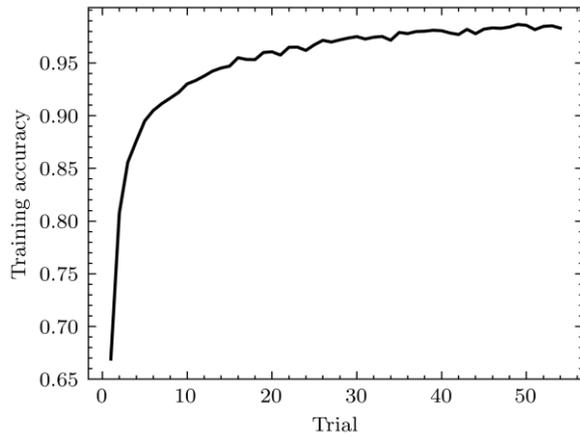


Fig. 11. Best fine-tuning filtered trial,  $j = 1$

Table 6

Optimized learning-related hyperparameters combination

Hyperparameter	Value
Learning rate	0.04590
Batch size	25
Validation split	0.15
$l_2$ regularization	0.05506
Optimizer	SGD

The graphs in Fig. 12, 13 illustrate the training and validation classification accuracies dependence on epochs for final optimized convolutional neural network with architecture defined in Fig. 8 and learning-related hyperparameters from Table 6. Original accuracy and loss curves are distorted by noise, so to capture the trend they had smoothed them out by replacing the actual loss and accuracy values with an exponential moving average with factor  $\alpha = 0.4$ .

As we can observe from graphs 23 epochs are enough to learn the optimized model (to prevent overfitting EarlyStopping [37] with the patience of 5 was used). The total training time was approximately 10 minutes, while the baseline model [1] took more than 3 hours (438 epochs). Such difference between training time and the number of epochs could be explained due to the bigger value of learning rate ( $1e-6$  in baseline model vs 0.04590 in optimized one) that leads to a higher speed of convergence.

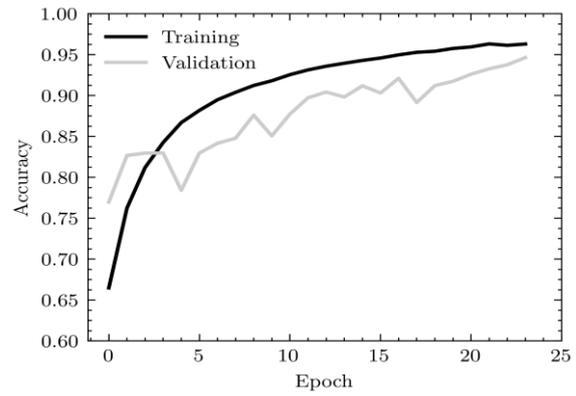


Fig. 12. Classification accuracy of the optimized model

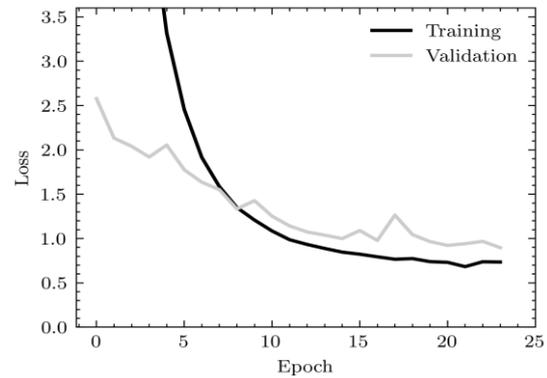


Fig. 13. Classification loss of the optimized model

Table 7 shows the percentage improvement of the F1 score metric for each class in the EuroSAT dataset. The neural network optimization process allowed highly increase the F1 for Herbaceous Vegetation, Permanent Crop, and Highway classes (up to 20%). Only the Forest class among ten showed a slight decrease of the score. The final classification accuracy of the optimized model on the test dataset (15% from all images) increased from  $84.19\%$  to  $95.31 \pm 1.73\%$  and F1 score – from  $84.05\%$  to  $95.33 \pm 1.72\%$ . The obtained results outperformed the selected baseline model from previous research [1].

Table 7

F1 metric comparison table of baseline and optimized models (%)

No	Class	Baseline	Optimized	$\Delta^*$
1	River	88.78	96.90	8.12
2	Highway	73.17	94.18	21.01
3	PermanentCrop	67.52	88.97	21.45
4	Industrial	88.95	97.04	8.09
5	Herbaceous Vegetation	71.34	91.85	20.51
6	AnnualCrop	84.72	93.06	8.34
7	Pasture	77.02	92.50	15.48
8	Forest	93.55	93.33	-0.22
9	Residential	93.46	98.74	5.28
10	SeaLake	96.67	99.07	2.4

\*- Difference between baseline and optimized columns

## Conclusions

Deep learning has made remarkable progress in different fields of study such as natural language processing, computer vision and it shows promising results in classification and analyzing remote sensing images. Despite a well-prepared input dataset shows high performance, optimizing hyperparameters in deep neural networks could increase final metrics. Among all optimization techniques (black-box and multi-fidelity ones) BOHB algorithm is the latest, effective in solving the optimization problem, and could be used by the modern Ray Tune framework.

The current research is devoted to exploring an optimal range of possible hyperparameters and finding the best configurations on a given budget automatically. Final optimized convolutional neural network structure and learning-related hyperparameters values are presented in Fig. 8 and Table 6. It is obtained an overall increase of F1 score (Table 7) for almost all classes in the EuroSAT dataset (Herbaceous Vegetation, Permanent Crop, and Highway – up to 20%). Final classification accuracy and F1 score of the optimized model on the test dataset outperformed baseline results: from 84.19% to  $95.31 \pm 1.73\%$  and from 84.05% to  $95.33 \pm 1.72\%$  respectively. In [38] authors presented a 4-convolution neural network, used all 13 spectral bands, and achieved a result accuracy was 94.90%. In [33] authors got result accuracy 98.57% using ResNet-50 model. But the model contains approximately 23 million trainable parameters and consists of 50 layers. In the current research, we got comparative results, but the proposed neural network is simple (3 convolutional layers and only one dense layer) with a smaller number of trainable parameters and needs less time for training.

In further research, the comprehension of black-box optimization algorithms and BOHB will be provided. Also, augmentation technics that increase result model performance will be used in the next works. While the process of hyperparameters optimization requires a big number of resources using low-dimensional data could speed it up [39].

**Contribution of authors:** devised the ideas of the research, the main conceptual approaches of modern approaches in deep learning optimization techniques. The author carried out the experiments, performed the numerical calculations, formulated of conclusions and was involved in designing experiments pipelines – **V. Yaloveha**; processed the experimental data, performed the analysis, drafted, and reviewed the manuscript, analyzed of references – **A. Podorozhniak**; were involved in planning and supervised the work, formulation of the purpose and tasks of research, development of mathematical approaches – **H. Kuchuk**. All authors

have read and agreed to the published version of the manuscript.

## References (GOST 7.1:2006)

1. Yaloveha, V. *Spectral Indexes Evaluation for Satellite Images Classification using CNN [Text]* / V. Yaloveha, D. Hlavcheva, A. Podorozhniak // *Journal of Information and Organizational Sciences*. – 2021. – Vol. 46, No. 2. – P. 95-113. DOI: 10.31341/jios.45.2.5.
2. *Data pre-processing to increase the quality of optical text recognition systems [Text]* / K. Dergachov, L. Krasnov, V. Bilozerskyi, A. Zymovin // *Radioelectronic and Computer Systems*. – 2021. – No. 4. – P. 183-198. DOI: 10.32620/reks.2021.4.15.
3. *System of license plate recognition considering large camera shooting angles [Text]* / H. Kuchuk, A. Podorozhniak, N. Liubchenko, D. Onischenko // *Radioelectronic and Computer Systems*. – 2021. – No. 4. – P. 82-91. DOI: 10.32620/reks.2021.4.07.
4. Liubchenko, N. *Research of antispam bot algorithms for social networks [Text]* / N. Liubchenko, A. Podorozhniak, V. Oliinyk // *CEUR Workshop Proceedings*. – 2021. – Vol. 2870. – P. 822-831.
5. Pal, A. *A review of deep learning methods for MRI reconstruction [Text]* / A. Pal, Y. Rathi // *Preprint arXiv:2109.08618*. – 2021.
6. Jordan, M. I. *Machine learning: Trends, perspectives, and prospects [Text]* / M. I. Jordan, T. M. Mitchell // *Science*. – 2015. – Vol. 349, No. 6245. – P. 255-260. DOI: 10.1126/science.aaa8415.
7. *Neural network approach for multispectral image processing [Text]* / A. Podorozhniak, N. Lubchenko, O. Balenko, D. Zhuikov // *14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*. – 2018. – P. 978-981. DOI: 10.1109/TCSET.2018.8336357.
8. Ieremeiev, O. *Combined visual quality metric of remote sensing images based on neural network [Text]* / O. Ieremeiev, V. Lukin, K. Okarma // *Radioelectronic and computer systems*. – 2020. – No. 4. – P. 4-15. DOI: 10.32620/reks.2020.4.01.
9. Hlavcheva, D. *Application of convolutional neural network for histopathological analysis [Text]* / D. Hlavcheva, V. Yaloveha, A. Podorozhniak // *Advanced Information Systems*. – 2019. – Vol. 3, No. 4. – P. 69-73. DOI: 10.20998/2522-9052.2019.4.10.
10. *Comparison of CNNs for Lung Biopsy Images Classification [Text]* / D. Hlavcheva, V. Yaloveha, A. Podorozhniak, H. Kuchuk // *2021 IEEE 3rd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. – IEEE, 2021. – P. 1-5. DOI: 10.1109/UKRCON53503.2021.9575305.
11. Wang, S. C. *Interdisciplinary computing in Java programming [Text]* / S. C. Wang // *Springer Science & Business Media*. – 2003. – P. 81-100.
12. Yang, L. *On hyperparameter optimization of machine learning algorithms: Theory and practice [Text]* / L. Yang, A. Shami // *Neurocomputing*. – 2020. – Vol.

415. – P. 295-316. DOI: 10.1016/j.neucom.2020.07.061.

13. Hyperparameter optimization for machine learning models based on Bayesian optimization [Text] / J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, L. D. Lei, S. H. Deng // *Journal of Electronic Science and Technology*. – 2019. – Vol. 17, No. 1. – P. 26-40. DOI: 10.11989/JEST.1674-862X.80904120.

14. Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data [Text] / A. Koutsoukas, K. J. Monaghan, X. Li, J. Huan // *Journal of cheminformatics*. – 2017. – Vol. 9, No. 1. – P. 1-13. DOI: 10.1186/s13321-017-0226-y.

15. Elshawi, R. Automated machine learning: State-of-the-art and open challenges [Text] / R. Elshawi, M. Maher, S. Sakr // *Preprint arXiv arXiv:1906.02287*. – 2019.

16. Experiencethinking: Hyperparameter optimization with budget constraints [Text] / C. Wang, H. Wang, C. Zhou, H. Chen, J. Li, H. Gao // *Preprint arXiv arXiv:1912.00602*. – 2019.

17. Bergstra, J. Random search for hyperparameter optimization [Text] / J. Bergstra, Y. Bengio // *Journal of machine learning research*. – 2012. – Vol. 13, No. 2. – P. 281–305.

18. Bellman, R. E. Adaptive control processes: A Guided Tour [Text] / R. E. Bellman. – Princeton: Princeton university press, 2015. DOI: 10.1515/9781400874668.

19. Witt, C. Worst-case and average-case approximations by simple randomized search heuristics [Text] / C. Witt // *Annual Symposium on Theoretical Aspects of Computer Science*. – Springer. – 2005. – P. 44-56. DOI: 10.1007/978-3-540-31856-9\_4.

20. Hutter, F. Automated machine learning: methods, systems, challenges [Text] / F. Hutter, L. Kotthoff, J. Vanschoren. – Springer Nature, 2019. – 219 p. DOI: 10.1007/978-3-030-05318-5.

21. Zöller, M. A. Benchmark and survey of automated machine learning frameworks [Text] / M. A. Zöller, M. F. Huber // *Journal of artificial intelligence research*. – 2021. – Vol. 70. – P. 409-472. DOI: 10.1613/jair.1.11854.

22. Hyperband: A novel bandit-based approach to hyperparameter optimization [Text] / L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar // *The Journal of Machine Learning Research*. – 2017. – Vol. 18, No. 1. – P. 6765-6816.

23. Krizhevsky, A. Imagenet classification with deep convolutional neural networks [Text] / A. Krizhevsky, I. Sutskever, G. Hinton // *Advances in neural information processing systems*. – 2012. – Vol. 25. – P. 1097–1105.

24. Jamieson, K. Non-stochastic best arm identification and hyperparameter optimization [Text] / K. Jamieson, A. Talwalkar // *Artificial intelligence and statistics*, PMLR. – 2016. – P. 240-248.

25. Falkner, S. BOHB: Robust and efficient hyperparameter optimization at scale [Text] / S. Falkner,

A. Klein, F. Hutter // *International Conference on Machine Learning*, PMLR. – 2018. – P. 1437-1446.

26. Algorithms for hyper-parameter optimization [Text] / J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl // *Advances in neural information processing systems*. – 2011. – Vol. 24. – P. 2546–2554.

27. Deep learning in environmental remote sensing: Achievements and challenges [Text] / Q. Yuan, H. Shen, T. Li, Z. Li, et al. // *Remote Sensing of Environment*. – 2020. – Vol. 241. DOI: 10.1016/j.rse.2020.111716.

28. Joint Deep Learning for land cover and land use classification [Text] / C. Zhang, I. Sargent, X. Pan, H. Li // *Remote sensing of environment*. – 2019. – Vol. 221. – P. 173-187. DOI: 10.1016/j.rse.2018.11.014.

29. Chlingaryan, A. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review [Text] / A. Chlingaryan, S. Sukkariéh, B. Whelan // *Computers and electronics in agriculture*. – 2018. – Vol. 151. – P. 61-69. DOI: 10.1016/j.compag.2018.05.012.

30. Mazzia, V. Improvement in land cover and crop classification based on temporal features learning from Sentinel-2 data using recurrent-convolutional neural network (R-CNN) [Text] / V. Mazzia, A. Khaliq, M. Chiaberge // *Applied Sciences*. – 2020. – Vol. 10, No. 1. – Article no. 238. DOI: 10.3390/app10010238.

31. Deep learning classification of land cover and crop types using remote sensing data [Text] / N. Kussul, M. Lavreniuk, S. Skakun, A. Shelestov // *IEEE Geoscience and Remote Sensing Letters*. – 2017. – Vol. 14, No. 5. – P. 778-782. DOI: 10.1109/LGRS.2017.2681128.

32. Fire hazard research of forest areas based on the use of convolutional and capsule neural networks [Text] / V. Yaloveha, D. Hlavcheva, A. Podorozhniak, H. Kuchuk // *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. – 2019. – P. 828-832. DOI: 10.1109/UKRCON.2019.8879867.

33. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification [Text] / P. Helber, B. Bischke, A. Dengel, D. Borth // *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. – 2019. – Vol. 12, No. 7. – P. 2217-2226. DOI: 10.1109/JSTARS.2019.2918242.

34. Abreu, S. Automated architecture design for deep neural networks [Text] / S. Abreu // *arXiv preprint arXiv:1908.10714*. – 2019.

35. Liaw, R. Tune: A research platform for distributed model selection and training [Text] / R. Liaw, E. Liang, R. Nishihara, P. Moritz et al. // *Preprint arXiv arXiv:1807.05118*. – 2018.

36. Carneiro, T. Performance analysis of google colab as a tool for accelerating deep learning applications [Text] / T. Carneiro, R. Nóbrega, T. Nepomuceno // *IEEE Access*. – 2018. – Vol. 6. – P. 61677-61685. DOI: 10.1109/ACCESS.2018.2874767.

37. Prechelt, L. Automatic early stopping using cross validation: quantifying the criteria [Text] /

L. Prechelt // *Neural Networks*. – 1998. – Vol. 11, No. 4. – P. 761-767. DOI: 10.1016/S0893-6080(98)00010-0.

38. Chong, E. *EuroSAT Land Use and Land Cover Classification using Deep Learning [Online] / E. Chong*. – Available at: <https://github.com/echong/Remote-Sensing>. – 20.07.2021

39. *Speeding up the hyperparameter optimization of deep convolutional neural networks / T. Hinz, N. Navarro-Guerrero, S. Magg, S. Wermter // International Journal of Computational Intelligence and Applications*. – 2018. – Vol. 17, No. 2. – P. 1-15.

## References (BSI)

1. Yaloveha, V., Hlavcheva, D., Podorozhniak A., Kuchuk, H. Spectral Indexes Evaluation for Satellite Images Classification using CNN. *Journal of Information and Organizational Sciences*, 2021, vol. 46, no. 2, pp. 95-113. DOI: 10.31341/jios.45.2.5.

2. Dergachov, K., Krasnov, L., Bilozerskyi, V., Zymovin, A. Data pre-processing to increase the quality of optical text recognition systems. *Radioelectronic and Computer Systems*, 2021, no. 4, pp. 183-198. DOI: 10.32620/reks.2021.4.15.

3. Kuchuk, H., Podorozhniak, A., Liubchenko, N., Onischenko, D. System of license plate recognition considering large camera shooting angles. *Radioelectronic and Computer Systems*, 2021, no. 4, pp. 82-91. DOI: 10.32620/reks.2021.4.07.

4. Liubchenko, N., Podorozhniak, A., Oliinyk, V. Research of antispam bot algorithms for social networks. *CEUR Workshop Proceedings, 2021*, vol. 2870, pp. 822-831.

5. Pal, A. Rathi, Y. A review of deep learning methods for MRI reconstruction. *Preprint arXiv: 2109.08618*, 2021.

6. Jordan, M. I., Mitchell, M. I. Machine learning: Trends, perspectives, and prospects. *Science*, 2015, vol. 349, no. 6245, pp. 255-260. DOI: 10.1126/science.aaa8415.

7. Podorozhniak, A., Lubchenko, N., Balenko, O., Zhuikov, D. Neural network approach for multispectral image processing. *14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, 2018, pp. 978-981. DOI: 10.1109/TCSET.2018.8336357.

8. Ieremeiev, O., Lukin, V., Okarma, K. Combined visual quality metric of remote sensing images based on neural network. *Radioelectronic and computer systems*, 2020, no. 4, pp. 4-15. DOI: 10.32620/reks.2020.4.01.

9. Hlavcheva, D., Yaloveha, V., Podorozhniak A., Kuchuk, H. Application of convolutional neural network for histopathological analysis. *Advanced Information Systems*, 2019, vol. 3, no. 4, pp. 69-73. DOI: 10.20998/2522-9052.2019.4.10.

10. Hlavcheva, D. Yaloveha, V., Podorozhniak A., Kuchuk, H. Comparison of CNNs for Lung Biopsy Images Classification. *2021 IEEE 3rd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*,

2021, pp. 1-5. DOI: 10.1109/UKRCON53503.2021.9575305.

11. Wang, S. C. Interdisciplinary computing in Java programming. *Springer Science & Business Media*, 2003, pp. 81-100.

12. Yang, L., Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 2020, vol. 415, pp. 295-316. DOI: 10.1016/j.neucom.2020.07.061.

13. Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, L. D., Deng S. H. Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, 2019, vol. 17, no. 1, pp. 26-40. DOI: 10.11989/JEST.1674-862X.80904120.

14. Koutsoukas, A., Monaghan, K. J., Li, X., Huan, J. Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of cheminformatics*, 2017, vol. 9, no. 1, pp. 1-13. DOI: 10.1186/s13321-017-0226-y.

15. Elshawi, R., Maher, M., Sakr, S. Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287*, 2019.

16. Wang, C., Wang, H., Zhou, C., Chen, H., Li, J., Gao, H. Experiencethinking: Hyperparameter optimization with budget constraints. *Preprint arXiv arXiv: 1912.00602*, 2019.

17. Bergstra, J., Bengio, Y. Random search for hyper-parameter optimization. *Journal of machine learning research*, 2012, vol. 13, no. 2, pp. 281-305.

18. Bellman, R. E. *Adaptive control processes: A Guided Tour*. Princeton, Princeton university press, 2015. DOI: 10.1515/9781400874668.

19. Witt, C. Worst-case and average-case approximations by simple randomized search heuristics. *Annual Symposium on Theoretical Aspects of Computer Science, Springer*, 2005, pp. 44-56. DOI: 10.1007/978-3-540-31856-9\_4.

20. Hutter, F. Kotthoff, L., Vanschoren, J. Automated machine learning: methods, systems, challenges. *Springer Nature*, 2019. 219 p. DOI: 10.1007/978-3-030-05318-5.

21. Zöllner, M. A., Huber, M. F. Benchmark and survey of automated machine learning frameworks *Journal of artificial intelligence research*, 2021, vol. 70, pp. 409-472. DOI: 10.1613/jair.1.11854.

22. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 2017, vol. 18, no. 1, pp. 6765-6816.

23. Krizhevsky, A., Sutskever, I., Hinton, G. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012, vol. 25, pp. 1097-1105.

24. Jamieson, K., Talwalkar, A. Non-stochastic best arm identification and hyperparameter optimization. *Artificial intelligence and statistics, PMLR*, 2016, pp. 240-248.

25. Falkner, S., Klein, A., Hutter, F. BOHB: Robust and efficient hyperparameter optimization at scale. *International Conference on Machine Learning, PMLR*, 2018, pp. 1437-1446.

26. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 2011, vol. 24, pp. 2546–2554.

27. Yuan, Q., Shen, H., Li, T., Li, Z. et al. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, 2020, vol. 241. DOI: 10.1016/j.rse.2020.111716.

28. Zhang, C., Sargent, I., Pan, X., Li, H. Joint Deep Learning for land cover and land use classification. *Remote sensing of environment*, 2019, vol. 221, pp. 173-187. DOI: 10.1016/j.rse.2018.11.014.

29. Chlingaryan, A., Sukkariéh, S., Whelan, B. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and electronics in agriculture*, 2018, vol. 151, pp. 61-69. DOI: 10.1016/j.compag.2018.05.012.

30. Mazzia, V., Khaliq, A., Chiaberge, M. Improvement in land cover and crop classification based on temporal features learning from Sentinel-2 data using recurrent-convolutional neural network (R-CNN). *Applied Sciences*, 2020, vol. 10, no. 1, article no. 238. DOI: 10.3390/app10010238.

31. Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 2017, vol. 14, no. 5, pp. 778-782. DOI: 10.1109/LGRS.2017.2681128.

32. Yaloveha, V., Hlavcheva, D., Podorozhniak A., Kuchuk, H. Fire hazard research of forest areas based

on the use of convolutional and capsule neural networks. *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2019, pp. 828-832. DOI: 10.1109/UKRCON.2019.8879867.

33. Helber, P., Bischke, B., Dengel, A., Borth, D. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019, vol. 12, no. 7, pp. 2217-2226. DOI: 10.1109/JSTARS.2019.2918242.

34. Abreu, S. Automated architecture design for deep neural networks. *Preprint arXiv, arXiv: 1908.10714*, 2019.

35. Liaw, R., Liang, E., Nishihara, R., Moritz, P. et al. Tune: A research platform for distributed model selection and training. *Preprint arXiv, arXiv: 1807.05118*, 2018.

36. Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T. Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*, 2018, vol. 6, pp. 61677-61685. DOI: 10.1109/ACCESS.2018.2874767.

37. Prechelt, L. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 1998, vol. 11, no. 4, pp. 761-767. DOI: 10.1016/S0893-6080(98)00010-0.

38. Chong, E. *EuroSAT Land Use and Land Cover Classification using Deep Learning*. Available online: <https://github.com/echong/Remote-Sensing>. (Accessed: 20.07.2021).

39. Hinz, T., Navarro-Guerrero, N., Magg, S., Wermter, S. Speeding up the hyperparameter optimization of deep convolutional neural networks. *International Journal of Computational Intelligence and Applications*, 2018, vol. 17, no. 02, pp. 1-15.

Надійшла до редакції 04.10.2021, розглянута на редколегії 16.02.2022

## ОПТИМІЗАЦІЯ ГІПЕРПАРАМЕТРІВ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ КЛАСИФІКАЦІЇ ЗЕМЕЛЬНОГО ПОКРИТТЯ

В. А. Яловега, А. О. Подорожняк, Г. А. Кучук

Останнім часом алгоритми машинного навчання продемонстрували високу ефективність під час вирішення задач у різних областях, включаючи аналіз зображень дистанційного зондування, комп'ютерного зору, обробку природної мови, медичних проблем тощо. Відомо, що добре підготовлений набір вхідних даних може мати величезний вплив на результуючі метрики. Однак і правильно підібрана комбінація гіперпараметрів архітектури нейронної мережі може значно підвищити кінцеві результати. Тому проблема оптимального вибору гіперпараметрів стає ключовою в алгоритмі глибокого навчання. Це дослідження має на меті знайти архітектуру згорткової нейронної мережі з відповідною комбінацією гіперпараметрів для задачі класифікації земельного покриття за допомогою мультиспектральних зображень та підвищити результуючі метрики із заданими бюджетними обмеженнями. Процес пошуку відповідної комбінації гіперпараметрів може здійснюватися вручну або автоматично. Ручний пошук заснований на попередніх дослідженнях і потребує величезних людських зусиль. З іншого боку, існує багато автоматизованих методів оптимізації гіперпараметрів, які успішно застосовуються на практиці. Такі методи налаштування гіперпараметрів поділяються на дві групи: black-box методи оптимізації (такі як Grid Search, Random Search) і multi-fidelity методи (HyperBand, BOHB). Найновішим і перспективним серед усіх підходів є BOHB, який поєднує як байєсівську оптимізацію, так і bandit-based методи. Він може виконуватись асинхронно з заданими GPU ресурсами та бюджетом часу, що відіграє важливу роль у процесі оптимізації гіперпараметрів. У попередньому дослідженні запропоновано згорткову нейронну мережу глибокого навчання для вирішення проблеми класифікації земельного покриття на основі EuroSAT. Додавання спектральних індексів NDVI, NDWI, GNDVI разом із каналами RGB дало змогу підвищити точність класифікації (з 64,72% до 84,19%) і F1 метрику (з 63,89% до 84,05%). Проте архітектура згорткової ней-

ронної мережі та комбінація гіперпараметрів були обрані вручну. Це дослідження має на меті знайти архітектуру згорткової нейронної мережі з відповідною комбінацією гіперпараметрів для задачі класифікації земельного покриття за допомогою мультиспектральних зображень та підвищити результуючі метрики із заданими бюджетними обмеженнями. Проте архітектура згорткової нейронної мережі та комбінація гіперпараметрів були обрані вручну. Це дослідження має на меті знайти архітектуру згорткової нейронної мережі з відповідною комбінацією гіперпараметрів для задачі класифікації земельного покриття за допомогою мультиспектральних зображень та підвищити результуючі метрики із заданими бюджетними обмеженнями.

**Ключові слова:** оптимізація гіперпараметрів; EuroSAT; ВОНВ; згорткова нейронна мережа; земельне покриття; дистанційне зондування.

## ОПТИМИЗАЦИЯ ГИПЕРПАРАМЕТРОВ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ ДЛЯ КЛАССИФИКАЦИИ ЗЕМЕЛЬНОГО ПОКРЫТИЯ

*В. А. Яловега, А. О. Подорожняк, Г. А. Кучук*

В последнее время алгоритмы машинного обучения продемонстрировали высокую эффективность при решении задач в различных областях, включая анализ изображений дистанционного зондирования, компьютерного зрения, обработку природного языка, медицинских проблем и т. д. Известно, что хорошо подготовленный набор входных данных может сильно повлиять на результирующие метрики. Однако и правильно подобранная комбинация гиперпараметров архитектуры нейронной сети может значительно повысить результаты. Поэтому проблема оптимального выбора гиперпараметров становится ключевой в алгоритме глубокого обучения. Целью данного исследования является поиск архитектуры сверточной нейронной сети с соответствующей комбинацией гиперпараметров для задачи классификации земельного покрытия с помощью мультиспектральных изображений и повышение значений результирующих метрик с заданными бюджетными ограничениями. Процесс поиска соответствующей комбинации гиперпараметров может производиться вручную или автоматически. Ручной поиск основан на предыдущих исследованиях и нуждается в больших человеческих усилиях. С другой стороны, существует много автоматизированных методов оптимизации гиперпараметров, успешно применяемых на практике. Такие методы настройки гиперпараметров делятся на две группы: black-box методы оптимизации (такие как Grid Search, Random Search) и multi-fidelity методы (HyperBand, ВОНВ). Самым новым и перспективным среди всех подходов является ВОНВ, объединяющий как байесовскую оптимизацию, так и bandit-based методы. Он может выполняться асинхронно с заданными GPU ресурсами и бюджетом времени, что играет важную роль в процессе оптимизации гиперпараметров. В предварительном исследовании предложена сверточная нейронная сеть глубокого обучения для решения проблемы классификации земельного покрытия на основе EuroSAT. Добавление спектральных индексов NDVI, NDWI, GNDVI вместе с каналами RGB позволило повысить точность классификации (с 64,72% до 84,19%) и F1 метрику (с 63,89% до 84,05%). Однако архитектура сверточной нейронной сети и комбинация гиперпараметров была выбрана вручную. Целью данного исследования является поиск архитектуры сверточной нейронной сети с соответствующей комбинацией гиперпараметров для задачи классификации земельного покрытия с помощью мультиспектральных изображений и повышение значений результирующих метрик с заданными бюджетными ограничениями.

**Ключевые слова:** оптимизация гиперпараметров; EuroSAT; ВОНВ; сверточная нейронная сеть; земельное покрытие; дистанционное зондирование.

**Яловега Владислав Анатолійович** – аспірант кафедри обчислювальної техніки та програмування, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна

**Подорожняк Андрій Олексійович** – канд. техн. наук, доц., доц. каф. обчислювальної техніки та програмування, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна

**Кучук Георгій Анатолійович** – д-р техн. наук, проф., проф. каф. обчислювальної техніки та програмування, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна

**Vladyslav Yaloveha** – PhD Student of the Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, e-mail: vladyslavyaloveha@gmail.com, ORCID: 0000-0001-7109-9405.

**Andrii Podorozhniak** – PhD (Engineering Sciences), Associate Professor, Associate Professor of the Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, e-mail: andriipodorozhniak@gmail.com, ORCID: 0000-0002-6688-8407.

**Heorhii Kuchuk** – Doctor of Technical Sciences, Professor, Professor of the Department of Computer Engineering and Programming, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine, e-mail: kuchuk56@ukr.net, ORCID: 0000-0002-2862-438X.