Hyperparameters Optimization of Ensemble-based Methods for Retina Image Classification

Marina Lukashevich
Postdoctoral Researcher, Electronic Computing Machines Department
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
lukashevich@bsuir.by

Sergei Bairak
Electronic Computing Machines Department
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
baikar@bsuir.by

Valery Starovoitov
Laboratory of the System Identification
United Institute of Informatics Problems
Minsk, Belarus
valerys@newman.bas-net.by

***Abstract*— Diabetic retinopathy causes damage to the eye's retina and leads to visual impairment in diabetic patients worldwide. It affects the retina, begins asymptomatically and can lead to vision loss. It can be diagnosed quite accurately by using machine learning algorithms to analyze retina images. Diagnosis at an early stage is crucial to prevent dangerous consequences such as blindness. This paper presents a comparative analysis of ensemble machine learning algorithms and describes an approach to the selection of hyperparameters to solve the problem of diabetic retinopathy stage classification (from 0 to 4). Special attention is focused on grid search and random search approaches. This study proposed a hyperparameter selection technique for ensemble algorithms based on the combination of grid search and random search approaches. Hyperparameter selection increased retina image classification accuracy. Experimental results shown that hyperparameter selection increased retina image classification accuracy for testing dataset from 0.7460 for best model (GB) with default parameters to 0.7503 for best model (RF). If we consider binary classification (diabetic retinopathy presents or not) it is possible to achieve accuracy of about 0.9304 (RF).**

Keywords—retina images, diabetic retinopathy recognition, machine learning, ensemble methods, hyperparameter, grid search, random search

I. INTRODUCTION

Fundus photography makes it quite easy to capture the retina image. Automation of digital image analysis and interpretation is still very poorly developed. In the field of diabetic retinopathy, one of the most important applications of research is the early prediction and disease diagnosis. Machine learning methods can extract patterns from images and have generalization abilities that allow to build effective models for image classification [1-4]. The objective of ensemble methods is to combine the predictions of a few base estimators built with a given learning algorithm in arrangement to improve generalizability over a single estimator [5-7].

Hyperparameters in machine learning are a model's parameters whose values are predetermined before the training process. They can be parameters of the algorithm itself (for example, tree depth in random forest, number of neighbors in k Nearest Neighbor, weights of neurons in neural networks), as well as methods of feature processing, etc. There are several methods for solving this problem. The traditional way to optimize hyperparameters is grid search, which is the search for a manually defined subset of the hyperparameter learning algorithm's hyperparameter space. Despite its simplicity, this method has serious disadvantages. It is very slow because it is necessary to search for all combinations of all parameters. The search will continue even with obviously unsuccessful combinations. Often it is necessary to increase the search step for time-saving purposes, which may result in the fact that the optimal parameter value will not be found. Random search replaces the exhaustive enumeration of all combinations by their random selection. In most cases, it is faster than grid search, and the parameter values are not limited by the grid. However, it does not always allow us to find the optimum and does not protect from over-selection of obviously unsuccessful combinations [7-9].

This study proposed a hyperparameter selection technique for ensemble algorithms based on the combination of grid search and random search approaches.

II. RESEARCH BACKGROUND

The methodology of this study includes the following stages: data preprocessing, informative feature extraction, machine learning model development and hyperparameter model selection.

A. Dataset

This study used retina images from the Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection (APTOS 2019 BD) dataset [10]. This is a large dataset of retina images captured with a fundus lens under different visual conditions. 3662 images are labeled by experts according to the degree of severity of diabetic retinopathy on a scale of 0 to 4. The set is unbalanced, and the distribution of images by grade is as follows: 0 - 1805 images, 1 - 370 images, 2 - 999 images, 3 - 193 images, 4 - 295 images). Image examples are shown in Fig. 1.

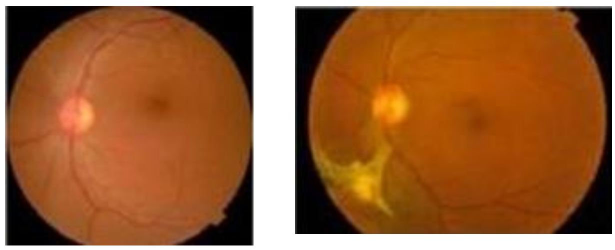

Fig. 1. Image examples

The dataset was divided into training and test sets (80% and 20%, respectively) with class balance preserved.

B. Image Preprocessing and Features Extraction

Preprocessing of retina images includes the following operations. At the image preprocessing stage it is necessary to perform *background cropping*, which sometimes occupies a significant percentage of the total image area and is practically a black and uninformative area. The experiment database contains images of different sizes and aspect ratios. Therefore, it is proposed to perform *image resizing* and make them 512*512 pixels. The following 5 groups of features were selected as features for decision making (classification): Haralick features, Local Binary Patterns (LBP), histogram features, Threshold Adjacency Statistics (TAS), Hu moments. *Standardization* was done for all features by removing the mean and unit variance scaling [11].

In this research we did not consider the stages of informative feature selection, as this could be a separate research branch. But we believe that this step together with the selection of informative features can also improve the classification results.

C. Machine Learning Models

One objective of this paper is to perform a comparative study to evaluate the most effective algorithm for grading diabetic retinopathy stages. Table I presents five investigated ensemble machine learning algorithms [5-7].

TABLE I. MACHINE LEARNING ALGORITHMS SELECTED FOR RESEARCH

#	Algorithm	Base estimator (if it's possible to define different)
1	Bagging Classifier (BG)	Decision tree classifier
2	Random Forest Classifier (RF)	
3	Extra Trees Classifier (ET)	
4	AdaBoost Classifier (AB)	Decision tree classifier
5	Gradient Boosting Classifier (GB)	

Cross-validation or k-fold cross-validation (k-fold cross-validation) with a value of k=10 is used during model development. Python programming language, machine learning library scikit-learn, computer vision and image processing libraries OpenCV and Mahotas were used in the research process [13-15]. Standard metrics for classification were used to evaluate model development: overall model prediction accuracy across all classes (accuracy), model accuracy in identifying positives (precision), completeness (recall), and F-measure (f1-score).

III. EXPERIMENTS

A. Technique for Hyperparameters Optimization

Two hyperparameters, significantly affecting the efficiency of ensemble algorithms, were chosen for the experiments.

'n_estimators' - the number of trees in the forest.

'max_depth' - the maximum depth of the tree.

'n_estimators' parameter is optimized for all five models. 'max_depth' is optimized for RF and ET models.

At the first stage, models were built on the basis of the training dataset with default parameter values defined in scikit-learn library. The accuracy of the models was evaluated on a test dataset. The results are shown in Table II. The confusion matrix and the classification report for GB model with the best score ('accuracy') for the testing set are shown in Fig. 2 and Fig. 3.

Preliminary experiments have proved the limitations of grid search and random search approaches given in the Introduction. Grid search took significant amount of time. The step size increasing partially solved this problem, but there was a risk of missing the optimal value. Random search is faster, but also does not guarantee the results quality.

TABLE II. THE 10-FOLD CROSS-VALIDATION FOR ALL MODELS BEFORE OPTIMIZATION

Model	Mean (std) score ('accuracy') for training set	Score ('accuracy') for testing set
BG	0.706 (0.030)	0.720
RF	0.743 (0.027)	0.742
ET	0.744 (0.028)	0.727
AB	0.679 (0.018)	0.686
GB	0.744 (0.022)	**0.746**

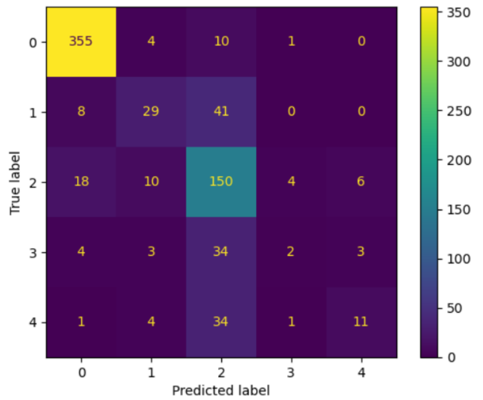

Fig. 2. Confusion matrix for GB with default hyperparameters

```
Classification report:
              precision    recall  f1-score   support

         0.0     0.9197    0.9595    0.9392       370
         1.0     0.5800    0.3718    0.4531        78
         2.0     0.5576    0.7979    0.6565       188
         3.0     0.2500    0.0435    0.0741        46
         4.0     0.5500    0.2157    0.3099        51

    accuracy                         0.7462       733
   macro avg     0.5715    0.4777    0.4865       733
weighted avg     0.7229    0.7462    0.7169       733
```

Fig. 3. Classification report for GB with default hyperparameters

The following hyperparameter search technique was implemented.

For each hyperparameter:

Step 1. Random search of hyperparameters in the specified range.

Step 2. Based on step 1 results (best score/accuracy across all searched params), perform range reduction and grid search with a given step size.

Step 3. Based on step 2 results (best score/accuracy across all searched params), perform range reduction, step reduction and the second grid search iteration.

Step 4. In the case where the best hyperparameter value is chosen at the boundary of the search space, shift the search interval towards the given value and perform the third grid search iteration.

For each combination of hyperparameters (only for RF and ET models):

Step 5. Search for specified hyperparameter values for the estimator.

B. *Step 1. Random Search*

Hyperparameters with search space, default hyperparameter value, best parameter value from the search space and for random search of hyperparameters are presented in Table 3. The 10-fold cross-validation for all models after Random Search Optimization for testing dataset is presented in Table III.

TABLE III. THE MACHINE LEARNING MODELS HYPERPARAMETERS SPACE AND CV-SCORE FOR RANDOM SEARCH

Model	Hyperparameter with search space	Default parameter	Best parameter	The best score ('accuracy') across all searched params for training set
BG	'n_estimators': [50, 200]	100	138	0.7480
RF	'n_estimators': [50, 200] 'max_depth' : [5, 70]	100 *None*	112 57	0.7562
ET	'n_estimators': [50, 200] 'max_depth' : [5, 70]	100 *None*	165 41	0.7528
AB	'n_estimators': [10, 200]	100	33	0.6934
GB	'n_estimators': [10, 200]	100	92	0.7490

TABLE IV. THE CV-SCORE FOR ALL MODELS AFTER RANDOM SEARCH OPTIMIZATION FOR TESTING DATASET

Model	BG	RF	ET	AB	GB
Test score ('accuracy')	0.7285	**0.7435**	0.7340	0.6985	0.7258

C. *Steps 2-5. Grid Search*

Hyperparameter with search space, best parameter, best score ('accuracy') across all searched params for training and test score ('accuracy') are presented in Table V.

TABLE V. THE MACHINE LEARNING MODELS HYPERPARAMETERS SPACE AND BEST CV-SCORE FOR GRID SEARCH

Model	Hyperparameter with search space	Best parameter	Best score ('accuracy') across all searched params for training	Test score ('accuracy')
BG	'n_estimators': [118, 128, 138, 148, 158]	138	0.7480	0.7480
	'n_estimators': [134, 136, 138, 140, 142]	140	0.7442	0.7367
RF	'n_estimators': [92, 102, 112, 122, 132]	112	0.7545	0.7562
	'n_estimators': [108, 110, 112, 114, 116]	116	0.7507	0.7353
	'n_estimators': [112, 114, 116, 118, 120]	114	0.7545	**0.7503**
	'max_depth': [47, 52, 57, 62, 67]	57	0.7521	0.7562
	'max_depth': [53, 55, 57, 59, 61]	57	0.7528	0.7562
	'n_estimators': [112, 114, 116, 118, 120] 'max_depth': [53, 55, 57, 59, 61]	112 53	0.7521	0.7340
ET	'n_estimators': [145, 155, 165, 175, 185]	185	0.7538	0.7299
	'n_estimators': [181, 183, 185, 187, 189]	189	0.7490	0.7285
	'max_depth': [31, 36, 41, 46, 51]	51	0.7483	0.7285
	'max_depth': [47, 49, 51, 53, 55]	51	0.7463	0.7285
	'n_estimators': [145, 155, 165, 175, 185] 'max_depth': [31, 36, 41, 46, 51]	155 46	0.7517	0.7271
AB	'n_estimators': [13, 23, 33, 43, 53]	33	0.6934	0.6985
	'n_estimators': [29, 31, 33, 35, 37]	31	0.6937	0.6903
GB	'n_estimators': [72, 82, 92, 102, 112]	112	0.7483	0.7435
	'n_estimators': [108, 110, 112, 114, 116]	116	0.7473	0.7435

Grid search process of 'n_estimators' hyperparameter for BG model is visualized in Fig. 4. Grid search process of 'n_estimators' hyperparameter for RF model is visualized in Fig. 5. Grid search process of 'max_depth' hyperparameter for RF model is visualized in Fig. 6. Grid search process of 'n_estimators' and 'max_depth' hyperparameters for RF model is visualized in Fig. 7. Grid search process of 'n_estimators' hyperparameter for ET model is visualized in Fig. 8. Grid search process of 'max_depth' hyperparameter for ET model is visualized in Fig. 9. Grid search process of 'n_estimators' and 'max_depth' hyperparameters for ET model is visualized in Fig. 10. Grid search process of 'n_estimators' hyperparameter for AB model is visualized in Fig. 11. Grid search process of 'n_estimators' hyperparameter for GB model is visualized in Fig. 12.

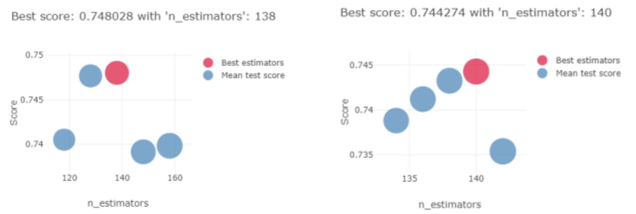

Fig. 4. Grid search process of 'n_estimators' hyperparameter for BG

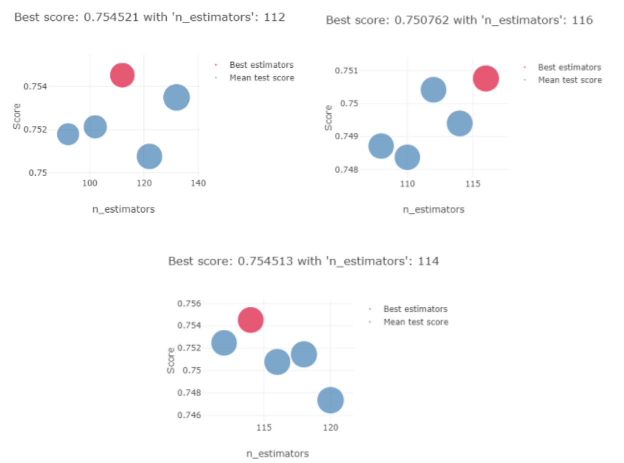

Fig. 5. Grid search process of 'n_estimators' hyperparameter for RF

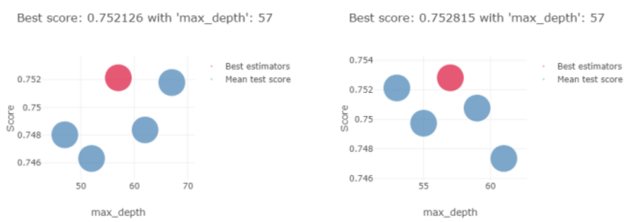

Fig. 6. Grid search process of 'max_depth' hyperparameter for RF

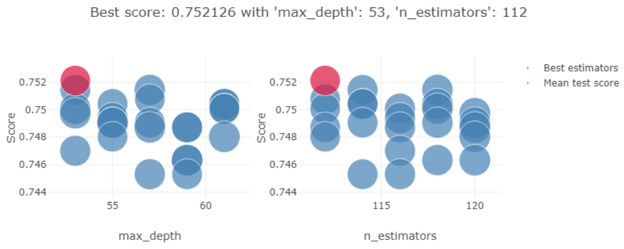

Fig. 7. Grid search process of 'n_estimators' and 'max_depth' hyperparameters for RF

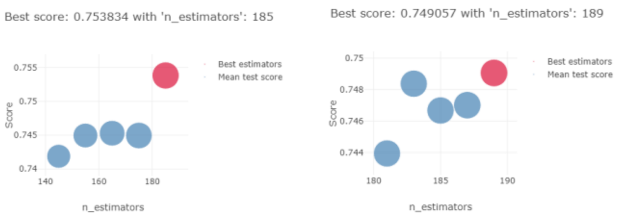

Fig. 8. Grid search process of 'n_estimators' hyperparameter for ET

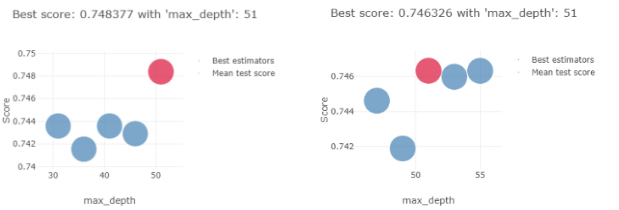

Fig. 9. Grid search process of 'max_depth' hyperparameter for ET

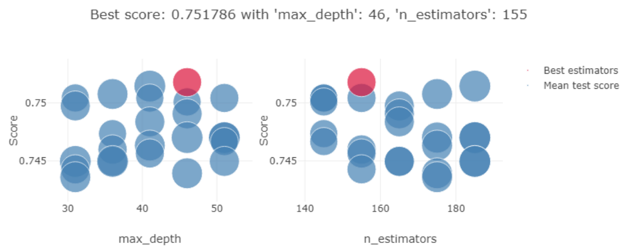

Fig. 10. Grid search process of 'n_estimators' and 'max_depth' hyperparameters for RF

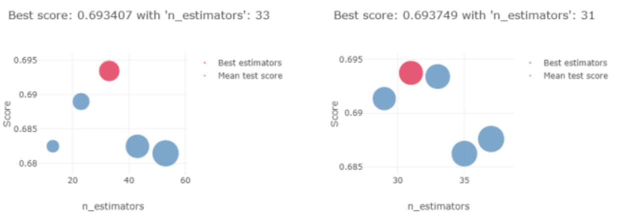

Fig. 11. Grid search process of 'n_estimators' hyperparameter for AB

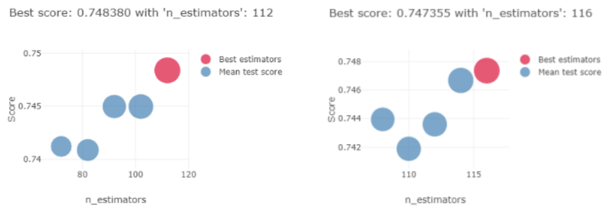

Fig. 12. Grid search process of 'n_estimators' hyperparameter for GB

D. Results Discussion

After applying random search, we see the decrease in score ('accuracy') for the testing dataset (0.7460 for GB to 0.7435 for RF). This is the negative side of this approach that we discussed before. The next stage with grid search allowed us to improve the result of random search and the results of initial models with default hyperparameters. For the testing dataset we received an improvement from 0.7435 to 0.7503 for RF. Two stages of technology showed better results. Fig. 13 and Fig 14. RF are shown the best score ('accuracy') for the testing dataset with 'n_estimators' : 114 and default 'max_depth' : None.

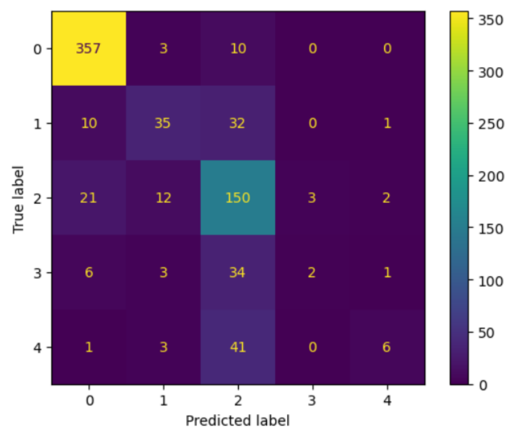

Fig. 13. Confusion matrix for RF with default hyperparameters

```
Classification report:
              precision    recall  f1-score   support

         0.0     0.9038    0.9649    0.9333       370
         1.0     0.6250    0.4487    0.5224        78
         2.0     0.5618    0.7979    0.6593       188
         3.0     0.4000    0.0435    0.0784        46
         4.0     0.6000    0.1176    0.1967        51

    accuracy                         0.7503       733
   macro avg     0.6181    0.4745    0.4780       733
weighted avg     0.7337    0.7503    0.7144       733
```

Fig. 14. Classification report for RF with default hyperparameters

We received good score for 0 grading stage of diabetic retinopathy. Some errors are presented when we classify 1-4 stages of diabetic retinopathy. It is important to note that experimental dataset is very difficult and includes images with diffident size, quality. Images collected in wide range of environment with different conditions ang using different equipment. It means that the input data and its quality influence greatly to the final results of diabetic retinopathy classification based on images. The most important thing is to detect the present of retinopathy automatically. It is two class problem for image classification (binary classification). It could be the first, preliminary stage or additional instrument for treatment. After screening and diabetic retinopathy detection the next stage could be 'manual' specify process which includes doctor's consultation. In this case experimental results are promising, Fig. 15 and Table VI.

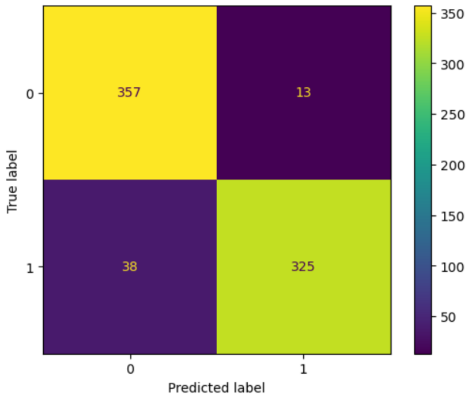

Fig. 15. Confusion matrix for binary classification

TABLE VI. CLASSIFICATION METRICS FOR BINARY CLASSIFICATION

Metric	Score
Accuracy	0.9304
Precision	0.9648
Recall	0.9037
F1-Score	0.9332

Results of binary classification show high accuracy. Presents of false positive error doesn't influence greatly into results. It causes only double check for doctors or addition signal for checking. False negative is more important. It is risk to skip diabetic retinopathy using image analysis

ACKNOWLEDGEMENT

The study was supported in part by the Belarusian Republican Foundation for Fundamental Research (project F22V-010).

CONCLUSION

This paper presents a comparative analysis of ensemble machine learning algorithms and describes an approach for hyperparameter selection to solve the problem of diabetic retinopathy stage classification. This study proposed a hyperparameter selection technique for ensemble algorithms based on the combination of grid search and random search approaches. Experimental results showed that hyperparameter selection increased retina image classification accuracy for the testing dataset from 0.7460 for the best model (GB) with default parameters to 0.7503 for best model (RF). If we consider binary classification (diabetic retinopathy presents or not) it is possible to achieve accuracy of about 0.9304.

REFERENCES

[1] R. K. Kumar and K. Arunabhaskar, "A Hybrid Machine Learning Strategy Assisted Diabetic Retinopathy Detection based on Retinal Images," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2021, pp. 1-6, doi: 10.1109/ICSES52305.2021.9633875.

[2] Dagliati, A, Marini, S, Sacchi, L, Cogni, G, Teliti, M, Tibollo, V, De Cata, P, Chiovato, L & Bellazzi, R 2018, 'Machine Learning Methods to Predict Diabetes Complications', *Journal of Diabetes Science and Technology*, vol. 12, no. 2, pp. 295-302. https://doi.org/10.1177/1932296817706375

[3] Ioannis Kavakiotis, Olga Tsave, Athanasios Salifoglou, Nicos Maglaveras, Ioannis Vlahavas, Ioanna Chouvarda, Machine Learning and Data Mining Methods in Diabetes Research, Computational and Structural Biotechnology Journal, Volume 15, 2017,Pages 104-116, ISSN 2001-0370,

[4] Dagliati A, Marini S, Sacchi L, Cogni G, Teliti M, Tibollo V, De Cata P, Chiovato L, Bellazzi R. Machine Learning Methods to Predict Diabetes Complications. J Diabetes Sci Technol. 2018 Mar;12(2):295-302. doi: 10.1177/1932296817706375. Epub 2017 May 12. PMID: 28494618; PMCID: PMC5851210.

[5] Polikar, R. (2006) 'Ensemble based systems in decision making', *IEEE Circuits and Systems Magazine*, 6(3), pp. 21–45. doi:10.1109/mcas.2006.1688199.

[6] Rokach, L. (no date) 'Ensemble Methods for Classifiers', *Data Mining and Knowledge Discovery Handbook*, pp. 957–980. doi:10.1007/0-387-25465-x_45.

[7] Kuncheva, L.I. and Whitaker, C.J. (2003) *Machine Learning*, 51(2), pp. 181–207. doi:10.1023/a:1022859003006.

[8] Chin-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin (2010). A practical guide to support vector classification. Technical Report, National Taiwan University.

[9] Chicco D (December 2017). "Ten quick tips for machine learning in computational biology". BioData Mining. **10** (35): 35. doi:10.1186/s13040-017-0155-3. PMC 5721660. PMID 29234465

[10] *Aptos 2019 blindness detection* (no date) *Kaggle*. Available at: https://www.kaggle.com/competitions/aptos2019-blindness-detection/data (Accessed: 22 September 2023).

[11] M.M. Lukashevich, Y.I. Golub, V.V. Starovoitov Classification Of Diabetic Retinopathy Stadies Based on Machine Learning Algorithms and Characteristic Set // Proceedings of the International Scientific Congress on Informatics. In 3 parts. Ch. 2, Republic of Belarus, Minsk, October 27-28, 2022. pp. 169-176.

[12] Starovoitov V.V., Golub Y.I., Lukashevich M.M. Digital fundus image quality assessment. «*System analysis and applied information science*». 2021;(4):25-38. (In Russ.) https://doi.org/10.21122/2309-4923-2021-4-25-38

[13] *Learn* (no date) *scikit*. Available at: https://scikit-learn.org/stable/ (Accessed: 22 September 2023).

[14] *Home* (2023) *OpenCV*. Available at: https://opencv.org/ (Accessed: 22 September 2023).

[15] *Computer vision in python*¶ (no date) *Mahotas*. Available at: https://mahotas.readthedocs.io/en/latest/ (Accessed: 22 September 2023).