

RESEARCH ARTICLE

Image Classification of Leukemic Cells Using Invariants of Triangle-Free Graphs as Synthetic Features

DMITRY MUZALEVSKIY¹ AND IVAN TORSHIN²¹Independent Researcher, 10437 Berlin, Germany²FIC "Computer Science and Control," Institution of Russian Academy of Sciences, 119333 Moscow, Russia

Corresponding author: Dmitry Muzalevskiy (muzalevsky.ds@gmail.com)

ABSTRACT Development of efficient methods of the cellular image processing is an important avenue for practical application of modern artificial intelligence techniques. In particular, practical hematology requires automatic classification of images with or without leukemic (blast) cells in peripheral blood smears. This paper presents a new approach to the problem of classification of such cellular images based on graph theory, XGBoost algorithm and convolutional neural networks (CNN). Firstly, each image is transformed into a weighted graph using gradient of intensity. Secondly, a number of graph invariants are computed thus producing a set of synthetic features that is used to train machine learning model based on XGBoost. Combining XGBoost with CNN further increases the accuracy of leukemic cell classification. Sensitivity (TPR) and Specificity (TNR) of the XGBoost-based model were 95% and 97% accordingly; ResNet-50 model showed TPR of 95% and TNR of 98%. Combined use of the XGBoost-based and the ResNet-50 models demonstrated TPR of 99% and TNR of 99%.

INDEX TERMS Adjacency matrices, gradient boosting, graph invariants, image classification, neural networks, triangle-free graphs, weighted graphs.

I. INTRODUCTION

Automation of the processing of cellular images requires a number of image analysis tasks had to be solved. In particular, it is necessary to (1) detect presence of a particular cell type in the image, (2) distinguish between different types of cells in the images, (3) count the numbers of cells of each type and (4) highlight the relevant structures inside the cells.

This work presents a development of image analysis methods applied to classification of blood smear images obtained from patients with acute lymphoblastic leukemia (ALL) with the purpose of detecting leukemic (blast) cells. The practical importance of the methods proposed is obvious since smear analysis is performed only by trained specialists highly qualified in cell morphology.

ALL is a disease of the blood system that results from malignant transformation of B/T lymphocyte precursors [1]. It is the most common leukemia in the pediatric population,

accounting for up to 80% of cases in this group (compared to only 20% in adults). Early and accurate diagnosis of ALL is important for successful therapy [2]. Expert analysis of bone marrow smear images is necessary to establish a diagnosis of ALL. A characteristic feature of the images is the presence of so called "blast cells," a morphologically distinct group of cells including the earliest hematopoietic progenitor cells. Typically, these blast cells comprise 1%- 2% of the cells in bone marrow smears [3]. The malignant cells remain poorly differentiated, but carry markers of lymphoid and/or myeloid cells. Previously, it was believed that malignant cells, while proliferating, displaced normal hematopoietic cells from the bone marrow. Later, it was shown that the number of normal hematopoietic stem cells does not decrease, but they undergo a blockade of differentiation, resulting in the development of cytopenia [4]. Peripheral blood smear may contain more than 20% blast cells [5], [6].

The task of classifying images containing blast cells is complicated by the fact that three different types of lymphoblasts are found in bone marrow biopsy specimens

The associate editor coordinating the review of this manuscript and approving it for publication was Sandra Costanzo¹.

from patients with ALL: (1) small size, high nuclear-cytoplasmic ratio, rounded nuclei, delicate chromatin structure and poorly distinguishable nucleoli; (2) medium-sized with varied nucleus shape, delicate chromatin structure; (3) medium to large size with oval or rounded nucleus shape, delicate chromatin structure, one or more well-defined nucleoli, sharp basophilia and vacuolization of the cytoplasm [7]. A characteristic feature of the blast cells is their difference in size from lymphocytes which, nevertheless, varies widely [8]. These features of images containing blast cells increase the variability of image data, making it difficult to create effective classification algorithms.

This study proposes an approach to classify the images containing the leukemic cells by transforming the images into weighted triangle-free graphs in which the image gradient magnitudes are used as edge weights. The weighted graphs obtained from the images are used to compute several graph invariants which form a feature set that is subsequently used to train a machine-learning ensemble based on the XGBoost model architecture [9]. We also explored how the complementing the graph-based model with the ResNet-50 model [10] can improve the results of classification (in particular, on low-resolution images). Low resolution of the blood smear images represents an actual problem for neural networks of complex architectures and also has important practical ramifications since it is not always possible to obtain biomedical images of high resolution.

The remainder of this paper is organized as follows: In Section II we describe similar approaches and publications related to the subject of the problem to be solved. In Section III, we describe the dataset used in the study and the methods developed for preprocessing the data. In Section IV we describe the algorithms proposed, their implementations and overall experimental setup. In Section V we present the results of the algorithms application on real data, compare performance of our algorithms with the performance of the state-of-the-art neural networks. In Section VI we compare our results with the results of earlier studies that used the ALL_IDB1 dataset. Section VII presents finalizing remarks and highlights some ideas for future studies.

II. RELATED WORK

Several attempts have been made to apply various automated image processing methods to solve the problem of detecting leukemic cells in images of blood smears. In particular, paper [11] presents the idea of applying a CNN to classify images from an ALL-related dataset. The authors also proposed a valuable data augmentation approach and described implementation peculiarities of their CNN architecture. The shortcomings of the study are the lack of technical detail and relatively small number of computational experiments.

Closely related to this paper are the works [12], [13] in which the authors suggested the use of more sophisticated approaches based on segmentation combined with CNN and other types of network architectures. The merits of these

studies include a large number of tested algorithms (which were based on the application of neural networks of different architectures). A major disadvantage of both of these studies is insufficient description of the solutions proposed, including the approaches to the problem of training neural algorithms on small amounts of image data.

In [14], the authors used a combination (ensemble) of complex neural networks (VGG19, MobileNet, and DenseNet) to solve somewhat similar problem of biomedical image classification (breast cancer biopsy images) over a small set of data without using methods of graph theory. The advantages include an effective combination of several neural networks of complex architecture to perform classification while disadvantages are related to the fact that such combinations produce excessive complexity of the algorithms.

The main advantage of the paper [15] is to extract original image features called “Energy,” “Entropy,” “Shannon Entropy,” “Log Energy Entropy”, Mean, Variance, Correlation and to use these synthetic features to predict the image classes. The most prominent disadvantage of this approach is relatively low classification accuracy of the algorithms tested.

In [16] the authors used a gradient calculation method based on weighted graphs combined with a Random Forest algorithm. Paper shows that the image gradients constructed with the method proposed can be used as an alternative to the edge maps methods and described an interesting approach of training Random Forest algorithm on gradient-based information. Limitation of the study is due to the fact that it focuses only on the image segmentation tasks not covering classification use cases. Also, it’s worth mentioning that the algorithm proposed by authors was not tested on the real-world datasets. Other approaches of this kind were proposed in [17] and [18].

Here, we would also like to discuss the specifics of the papers that were using the same ALL_IDB1 dataset [19] for training and validation of the models developed. In the works [11], [20], and [21], different variants of CNNs were used to detect images with leukemic cells. They achieved accuracies of 96.6%, 98.3% and 96.5%, respectively. The authors of [13] used different set of models including the residual-based neural networks (such as ResNet-50); the accuracy reported was 96.2%. Several other works used different variations of SVM [22], [23], and [24] with reported accuracy of 93.7%, 98.1% and 94%, respectively. Both studies [25] and [26] used variations of the YOLO algorithm for the task of blast cell detection with reported accuracy of 97.2% and 96.1%, respectively. In [27], authors used 4-moment statistical features and several neural networks for building and training the model to achieve 97.1% accuracy on the same ALL_IDB1 dataset.

III. DATA AND PREPROCESSING

The ALL_IDB1 dataset used in this study consisted of 108 microscopic images representing approximately

39000 cells. Images were taken at various magnifications (x300-x500). An example of an image is presented in Fig. 1.

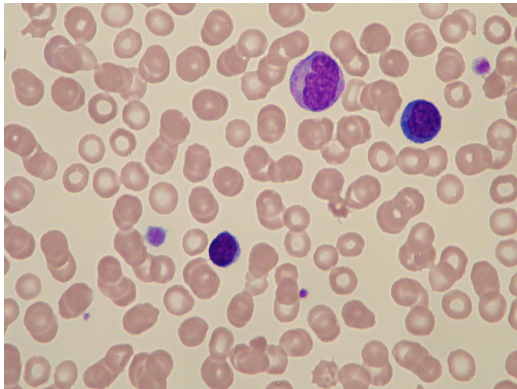


FIGURE 1. Example of a microscopic blood image. Lymphocytes (purple) and erythrocytes (most of the cells in the figure) are shown.

Two approaches, raw image processing and preprocessing of the tabular data, were used to prepare the data for training our models. Data augmentation (increasing the image sample using simple transformations such as image rotation, stretching, changing image brightness and adding reflections) was applied in order to combat overfitting [28]. For this purpose three basic transformations were applied to the original images:

- image rotation by a random number between 0 and 20 degrees;
- brightness alteration is in the range of [0.15;0.3], where 1 indicates the maximum brightness of the image;
- approximation (zooming) of the image.

For each original image from the dataset of 108 images, additional images were generated by applying all of the three transformations which resulted in the formation of a dataset of 3231 images. An example of augmented image is shown in Fig. 2.

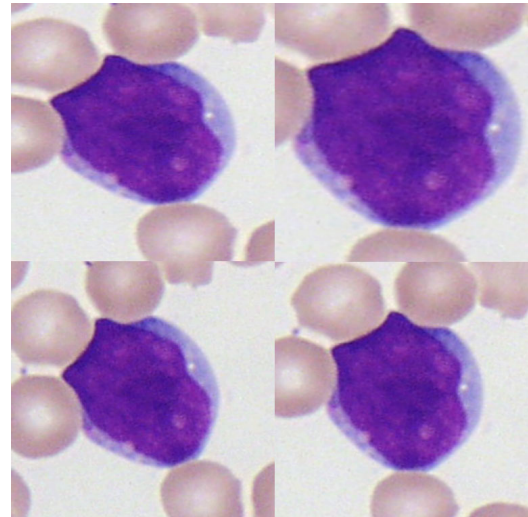


FIGURE 2. Example of an augmented image. Rotation, zooming and brightness adjustment operations were applied to the original image.

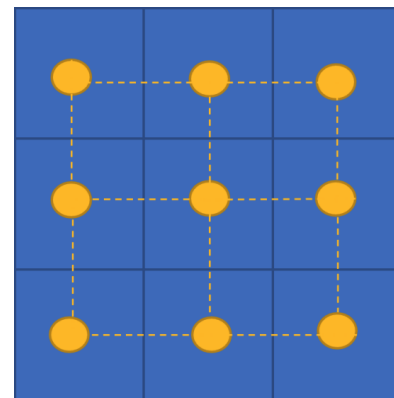


FIGURE 3. Process of forming edges inside the image pixels. Connecting vertices $\pm n$ (in this example ± 1 is used) along the x, y, z axes ensures that there will be no triangles in the newly generated graph.

IV. METHODS

A. GENERATING GRAPHS FROM THE IMAGE

The algorithm takes an RGB image with dimensions $H \times W \times D$ as input, where H is the height of the image in pixels, W is the width, and D is the number of channels (layers) in the image [29]. The elements of this array are intensities with values ranging from 0 to 255, where 0 is the minimum and 255 is the maximum intensity. Each pixel in the image corresponds to a vertex of the graph. Edges of the graph were calculated as the connections between adjacent pixels in each layer of the image by translating an array of indices into a 3D mesh and stacking slices of the mesh around the layers.

We demonstrate this process using an example on an imaginary input image I with dimensions of $(3, 3, 1)$. We select and number the vertices from the first to the last pixel in the image; in total, we obtain $3 \times 3 \times 1 = 9$ vertices. The edges generated by connecting adjacent pixels are shown in Fig. 3.

In this example, the edges were generated by considering only the x and y directions (because $z = 1$), and the case of ± 1 is considered. In cases with $z > 1$ there is also a third direction of edge formation (z -axis). The main stages of the graph generation procedure are shown in Fig. 4.

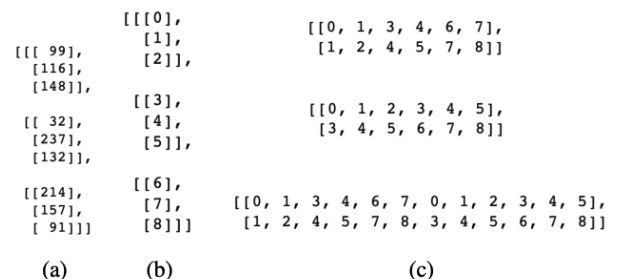


FIGURE 4. Process of the graph generation: a) Input image as image pixel intensities; b) Vertices formed based on each pixel of the input image; c) Edge creation in x -direction, in y -direction and the overall set of edges.

B. CALCULATING THE IMAGE GRADIENTS

The gradient of an image was calculated as the absolute difference in intensity between every two pixels connected by an edge [30].

$$gradient = |img_{x_1,y_1,z_1} - img_{x_2,y_2,z_2}|. \tag{1}$$

Next, the vertex indices were computed as follows:

$$(x_i, y_i, z_i) = \left(\left\lfloor \frac{E_i}{n_y \cdot n_z} \right\rfloor, \left\lfloor \frac{E_i \bmod (n_y \cdot n_z)}{n_z} \right\rfloor, E_i \bmod (n_y \cdot n_z) \bmod (n_z) \right), \tag{2}$$

where E_i is the vertex index of the image edge, n_y is the pixel width of the image, n_z is the number of image channels.

As a result, we obtained an adjacency matrix of a weighted graph, in which the main diagonal contains the pixel values of the original image and the other values in the matrix are either the resulting gradient values or zeros. This adjacency matrix corresponds to weighted graphs with self-loops and without triangles. An example of such a matrix is shown in Fig. 5.

99	17	0	67	0	0	0	0	0
17	116	32	0	121	0	0	0	0
0	32	148	0	0	16	0	0	0
67	0	0	32	205	0	182	0	0
0	121	0	205	237	105	0	80	0
0	0	16	0	105	132	0	0	41
0	0	0	182	0	0	214	57	0
0	0	0	0	80	0	57	157	66
0	0	0	0	0	41	0	66	91

FIGURE 5. Adjacency matrix with image gradient magnitudes. The main diagonal (marked in orange) contains pixel intensities of the original image, while the other values are the magnitudes of the gradients.

C. TRIANGLE-FREE GRAPHS

After forming the adjacency matrix as described above, the graphs thus obtained were triangle-free by construction [31]. The number of self-loops in such graph G equals to the number of vertices $|V|$. Let $A_G = \{(a_{ij})\}_{1 \leq i \leq j \leq n}$ and $A_G^2 = \{(b_{ij})\}_{1 \leq i \leq j \leq n}$. A triangle in G exists if and only if there are i, j, k so $a_{ij} \cdot a_{ik} \cdot a_{jk} = 1$ [32]. If $a_{ij} \cdot a_{ik} \cdot a_{jk} = 1$, we have $a_{ij} = 1$ and $b_{ij} = \sum_{l=1}^n a_{il} \cdot a_{lj} \neq 0$. In other direction, if $b_{ij} = \sum_{l=1}^n a_{il} \cdot a_{lj} \neq 0$ and $a_{ij} = 1$, there are $k \in \{1, \dots, n\}$ such $a_{ik} \cdot a_{jk} = 1$ so we have then $a_{ij} \cdot a_{ik} \cdot a_{jk} = 1$. The resulting graphs obtained from the images had a diameter equal to $H + W$, where H - is the height of the image in pixels and W - is the width of the image. These graphs had a clique number equal to 2. As the image dimensionality increases, the complexity of the graphs rises (an example is presented in Fig. 6).

D. GRAPH INVARIANTS AS SYNTHETIC FEATURES

After preliminary experiments with various graph invariants using the procedures of feature estimation and selection

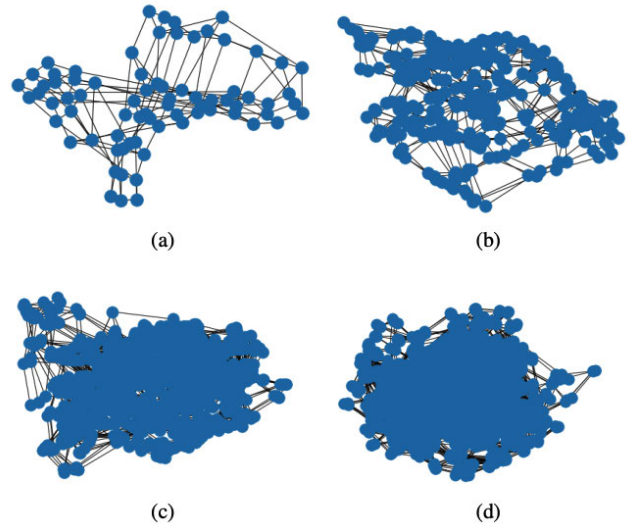


FIGURE 6. Growth of graph structures formed from microscopic images using the ± 1 approach: a) $5 \times 5 \times 3$ image; b) $10 \times 10 \times 3$ image; c) $20 \times 20 \times 3$ image; d) $32 \times 32 \times 3$ image. Self-loops are not displayed.

(see the next subsection), six types of invariants formed the final set of features: (1) Average Edge Weight, (2) Average Degree Connectivity, (3) Degree Assortativity Coefficient, (4) Local Efficiency, (5) Average Edge Weight (across vertices), and (6) Estrada index.

The average edge weight was calculated as the ratio of the total weight of a graph to the number of edges in the graph.

$$W = \frac{\sum_{i=1}^N w_i}{|E|}. \tag{3}$$

The average degree connectivity [33] is the average degree of the nearest neighbor of a vertex with degree k . For weighted graphs, this invariant was computed by calculating the weighted average vertex neighbor degree for vertex i as follows:

$$k_{mn,i}^w = \frac{1}{s_i} \sum_{j \in N(i)} w_{ij} k_j, \tag{4}$$

where s_i is the weighted degree of vertex i , w_{ij} is the weight of the edge connecting vertices i and j , and $N(i)$ are the neighbors of vertex i .

The degree assortativity coefficient [34] estimates the tendency of vertices to connect with other vertices with similar features within the graph and was computed as the Pearson correlation coefficient between vertex degrees:

$$r(i, j) = \frac{\sum_{i,j} (f(i) - \tilde{f}_1) (f(j) - \tilde{f}_2)}{\sqrt{\sum_{i,j} (f(i) - \tilde{f}_1)^2} \sqrt{\sum_{i,j} (f(j) - \tilde{f}_2)^2}}, \tag{5}$$

where $\tilde{f}_1 = \frac{\sum_j f(j)}{|E|}$, $\tilde{f}_2 = \frac{\sum_i f(i)}{|E|}$, $i, j \in E$.

The local efficiency [35] was computed for each individual vertex i in the graph by identifying the subgraphs to which vertex i is directly connected. After removing vertex i from the identified subgraph, the shortest path between all the

vertices in the subgraph was computed. The inverse of the shortest path from each vertex previously connected to vertex i to every other vertex previously connected to vertex i was then summed over all vertices, which was normalized to consider the total possible number of links that may exist between all vertices previously connected to i . Formally, local efficiency calculated as

$$E_{\text{local}} = \frac{1}{N_{G_i} (N_{G_i} - 1)} \sum_{j,k \in G_i} \frac{1}{L_{j,k}}, \quad (6)$$

where N_{G_i} is the number of vertices in subgraphs G_i and $L_{j,k}$ is the average distance (number of steps) between vertices i and j in the graph.

The average edge weight (across vertices) was defined as the ratio of the sum of the edge weights to the number of vertices in a given graph.

$$v = \frac{\sum_{i=1}^n w_i}{|V|}. \quad (7)$$

The Estrada index is the invariant that is used in chemical graph theory for topological operations with molecular structures [36] and was computed as follows:

$$EE(G) = \sum_{j=1}^n e^{\lambda_j}, \quad (8)$$

where $\lambda_1, \dots, \lambda_n$ are eigenvalues of the corresponding adjacency matrix A of the graph G .

During the study, other invariants were also tested. However, they were not included in the final set of synthetic features because of insignificance of their effects on the performance of the algorithm. Among them, we can mention such invariants as the Pagerank [36], Wiener Index [37], Edge Connectivity [38], Node Connectivity [39], Edge Betweenness Centrality [40], and Length of the Sparsest Cut [41].

After converting the augmented image set into graph representations and obtaining the graph-based feature set F , the feature values were standardized to a set of values from a distribution with zero mean and standard deviation equal to 1 [42].

E. APPROACHES TO FEATURE SELECTION

For feature selection, we used the Kolmogorov-Smirnov test statistic ($D_{n,m}$), feature importances from the XGBoost model (weight importances), and Shapley Values (mean SHAP values).

$D_{n,m}$ can be used to estimate the significance of the differences between two discretely presented distributions of the variable, with the null hypothesis that samples are drawn from the same distribution [43]. Let $F_n(x)$ be an empirical distribution function of a variable on $X = (X_1, \dots, X_n)$. Then

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I_{X_i \leq x}, \quad (9)$$

where I_{X_i} defined as following [44]:

$$I_{X_i \leq x} = \begin{cases} 1, & X_i \leq x \\ 0, & X_i > x \end{cases}. \quad (10)$$

Statistics for the function $F_n(x)$ is defined as the Kolmogorov's maximal deviance:

$$D_n = \sup_{x \in \mathbb{R}} |F_n(x) - F(x)|, \quad (11)$$

In case of two-sample statistics, the equation transforms into the following:

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)|. \quad (12)$$

We used the two-sample statistic $D_{n,m}$ to identify features significant for the model (comparing the distributions of each feature to the target class variable) and to determine the degree of confidence in the significance of the feature based on the value of the statistic [45].

Apart from $D_{n,m}$, we used the feature importances obtained within the XGBoost model. Specifically, we tested three estimates of feature importances: "weight", "gain" and "coverage". When using weight, we calculated the number of times a particular feature divided the data across all decision trees in the algorithm. Gain was defined as the average information gain of the tree splits that utilize the features. Cover was defined as the average coverage of the tree splits of the feature, while coverage represents the sample number affected by the tree split [46], [47]. Preliminary experiments showed that "weight" score proved to be the most accurate for feature evaluation in our case.

To evaluate the set of graph invariants we also used SHAP (SHapley Additive exPlanations) - a characteristic based on the results from game theory that can be used for feature importance calculations [48]. The SHAP of a feature was calculated as follows:

$$\phi_j(\text{val}) = \sum_{S \subseteq \{1, \dots, p\} \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} (\text{val}(S \cup \{j\}) - \text{val}(S)), \quad (13)$$

where S is a subset of the features used in the model, val - the value function of S and p is the number of features.

$$\text{val}_x(S) = \int \hat{f}(x_1, \dots, x_p) d\mathbb{P}_{x \notin S} - E_X(\hat{f}(X)), \quad (14)$$

where $\text{val}_x(S)$ is prediction for the values in S .

F. IMPLEMENTATIONS OF XGBOOST AND CNN ALGORITHMS

For training XGBoost model we used a set of features obtained as the result of testing various graph invariants. An early stopping procedure was also used in the model; the algorithm stopped training if the loss parameter did not decrease within five consecutive iterations. The model was trained using 100 estimators, a learning rate of 0.1 and a maximum depth of 3.

We have used the classical version of the pre-trained ResNet-50 with ImageNet weights. For fine-tuning using our

dataset, an Average Pooling layer, a fully-connected layer of four neurons and a dropout layer were added. We have used Binary Cross Entropy as the loss function and RMSProp as the optimizer. The output layer used the softmax activation function for class probability prediction. Softmax function was compared with sigmoid function but we found no evidence that sigmoid function outperforms softmax in terms of evaluation metrics we used or in terms of computation speed. The CNN was trained for 100 epochs; early stopping was added to control overfitting. Training stopped earlier if the loss rate on validation did not decrease for 5 consecutive epochs.

The predictions of the two models were averaged so that the final result was constructed based on the average probability of the classes from the models. Consider an example in which an algorithm produces five results: the first model $M1$ gives a set of predictions p_1, \dots, p_5 , and the second model $M2$: q_1, \dots, q_5 , so that

$$\sum_{i=1}^5 p_i = \sum_{i=1}^5 q_i = 1. \quad (15)$$

By setting any value of weight w in the interval $0 < w < 1$, we denote the combined ensemble prediction as

$$r_i := wp_i + (1 - w)q_i. \quad (16)$$

Then we obtain that

$$\sum_{i=1}^5 r_i = \sum_{i=1}^5 (wp_i + (1 - w)q_i) = w \sum_{i=1}^5 p_i + (1 - w) \sum_{i=1}^5 q_i = 1. \quad (17)$$

The value of parameter $w = 1/2$ was considered to be the optimal value and was further used for comparing approaches and algorithms [49].

G. EVALUATING THE ACCURACY OF THE ALGORITHMS

The developed algorithms were evaluated by dividing the dataset randomly in the ratio of 70% to 30% into training and test subsets, respectively. The common functional indicators for evaluating performance of binary classification (Accuracy, Precision, Recall, F1-Score and Area under ROC curve, AUC) were calculated for the evaluation of the algorithms [50].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (18)$$

$$Recall = \frac{TP}{TP + FN}, \quad (19)$$

$$Precision = \frac{TP}{TP + FP}, \quad (20)$$

$$F1 - Score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}, \quad (21)$$

$$AUC = \int TPRd(FPR), \quad (22)$$

where $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{TN}{TN+FP}$.

The code was written in Python 3.10.12 (using Anaconda of version 23.3.1).

We share all of the results and the code at <https://github.com/ds-muzalevskiy/graph-invariants-leukemia-detection>.

V. EXPERIMENTS

A. OVERALL SETTING

As part of the computational experiment, we solved the binary classification problem of detecting images with the blast cells. First, we augmented the original ALL_IDB1 dataset and obtained a set of 3231 images. Each image in this augmented image set was also compressed to 32×32 pixels. Secondly, after augmentation and rescaling the images were subsequently used in two ways: 1) the images were transformed into weighted graphs and graph invariants were calculated as described in Section IV which, in turn, were used for training the XGBoost model; 2) the images were directly used for training the ResNet-50 CNN algorithm. We also evaluated and compared the results of these two models (graph-invariant XGBoost model and CNN) and their combined performance.

B. INVARIANTS' SELECTION AND EVALUATION

We generated graphs by considering the connection of adjacent pixels as of ± 1 , ± 2 and ± 3 types thus obtaining three sets of graphs and then computing the same set of 11 invariants for the graphs in these three sets. The set of 11 invariants included the Average Edge Weight, Average Degree Connectivity, Local Efficiency, Global Efficiency, Degree Assortativity Coefficient, Average Neighbor Degree, Average Edge Weight (across vertices), Estrada Index, Wiener Index, Pagerank and Degree Centrality. Subsequently, we calculated invariant evaluation functionals using weight importances, mean shap values (also calculated using the XGBoost model) and $D_{n,m}$.

1) ± 1 GENERATED GRAPHS

The set of the most important invariants for the ± 1 graphs was formed by the Local Efficiency, Degree Assortativity, Average Edge Weight, Average Edge Weight (across the vertices), and Average Degree Connectivity. According to the values of SHAP and $D_{n,m}$, the Estrada Index can also be considered to be important for the model (Table 1). Low values of Local Efficiency (Fig. 7) correlated with a higher chance of obtaining the 0 class in the model (images without leukemic cells). On the contrary, higher values for the Degree Assortativity invariant were related to a higher likelihood of the model predicting the 1 class (images with leukemic cells). Additionally, higher values of the Average Edge Weight and Average Edge Weight (across the vertices) are related to 0 class prediction.

2) ± 2 GENERATED GRAPHS

For the ± 2 generated graph invariants, the most significant features were the Average Edge Weight (across the vertices), Average Degree Connectivity, Degree Assortativity and the Estrada Index. We can also notice that Global and

TABLE 1. Results of evaluation of significance of invariants for ± 1 , ± 2 and ± 3 graphs. Abbreviations: xgb-imp is Weight Importances, shap-val is Mean Shap Values and $D_{n,m}$ is Kolmogorov-Smirnov statistics.

Dataset	± 1 Generated Graphs			± 2 Generated Graphs			± 3 Generated Graphs		
	xgb-imp	$D_{n,m}$	mean-shap	xgb-imp	$D_{n,m}$	mean-shap	xgb-imp	$D_{n,m}$	mean-shap
average edge weight (across vertices)	74	0.61	0.86	88	0.37	0.54	115	0.24	0.42
average edge weight	64	0.56	0.64	62	0.29	0.41	62	0.22	0.13
average degree connectivity	70	0.55	0.21	91	0.11	0.19	77	0.17	0.25
degree assortativity	80	0.21	1.12	124	0.7	2.35	84	0.36	0.65
local efficiency	91	0.67	3.09	52	0.53	0.1	89	0.34	0.77
estrada index	69	0.64	0.52	112	0.6	2.12	39	0.32	0.11
global efficiency	64	0.57	0.43	71	0.45	0.41	44	0.34	0.13
average neighbor degree	19	0.65	0.07	18	0.56	0.09	29	0.34	0.16
degree centrality	10	0.66	0.04	20	0.56	0.1	29	0.33	0.09
pagerank	0	0	0	0	0	0	0	0	0
wiener index	0	0.53	0	0	0	0	0	0	0

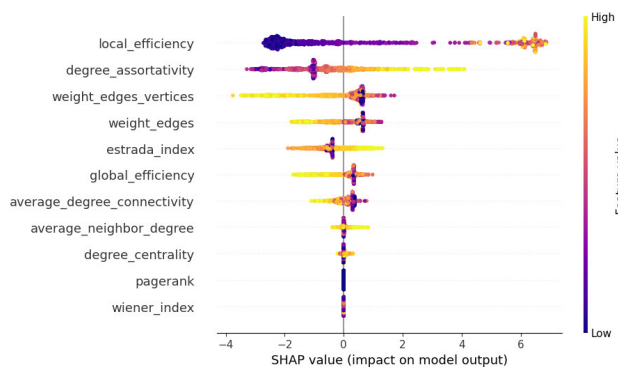


FIGURE 7. Estimates of the significance of the synthetic features for ± 1 graph invariants.

Local Efficiencies also play a role in the overall model’s performance. According to Table 1, Degree Assortativity and Estrada Index have the highest impact on the model output.

3) ± 3 GENERATED GRAPHS AND THE FINAL SET OF INVARIANTS

As n increases from 1 to 3, the graph begins to lose its edges. This explains the differences in feature importances for different n values (Fig. 8): for the ± 3 generated graphs, the most significant features were Local Efficiency, Degree Assortativity, Average Edge Weight (across the vertices) and Average Degree Connectivity (Table 1).

To form the final set of the most significant features we took the six most important features (according to the three functionals we used) of ± 1 graph invariants and added one of the most important features from the sets of ± 2 and ± 3 graph invariants. The complete list of features contained the following features from ± 1 : Local Efficiency, Degree Assortativity, Average Edge Weight, Average Edge Weight (across the vertices), Average Degree Connectivity, Estrada Index, Degree Assortativity from ± 2 and Local Efficiency from ± 3 graphs.

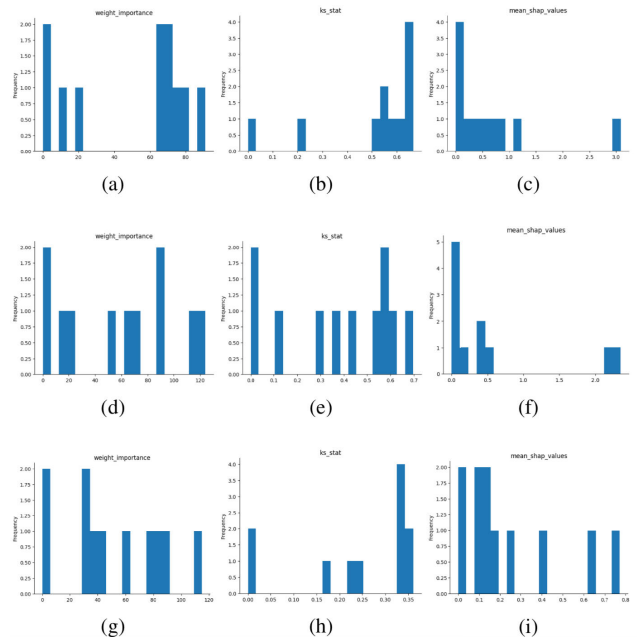


FIGURE 8. Feature significance distributions based on the evaluation methods used. a), b), c) are ± 1 graph invariants; d), e), f) are ± 2 graph invariants and g), h), i) are ± 3 graph invariants.

C. RESULTS OF THE IMAGE CLASSIFICATION EXPERIMENTS

In addition to ResNet-50 and graph-invariant XGboost models, that formed the basis of the study, a series of experiments were conducted with a set of different boosting algorithms for the tabular data: AdaBoost (ADA) [51], Gradient Boosting (GB) [52], LightGBM model [53], and NGBoost [54]. The Table 2 shows the results of standalone graph-invariant XGBoost (GXGB), ResNet-50 (RES-50), and the combined ensemble of the two models (COMB). The results of the evaluation of the models are also presented in the Figure 9.

After performing a standalone comparison of the performance of the graph-invariant XGBoost and Resnet-

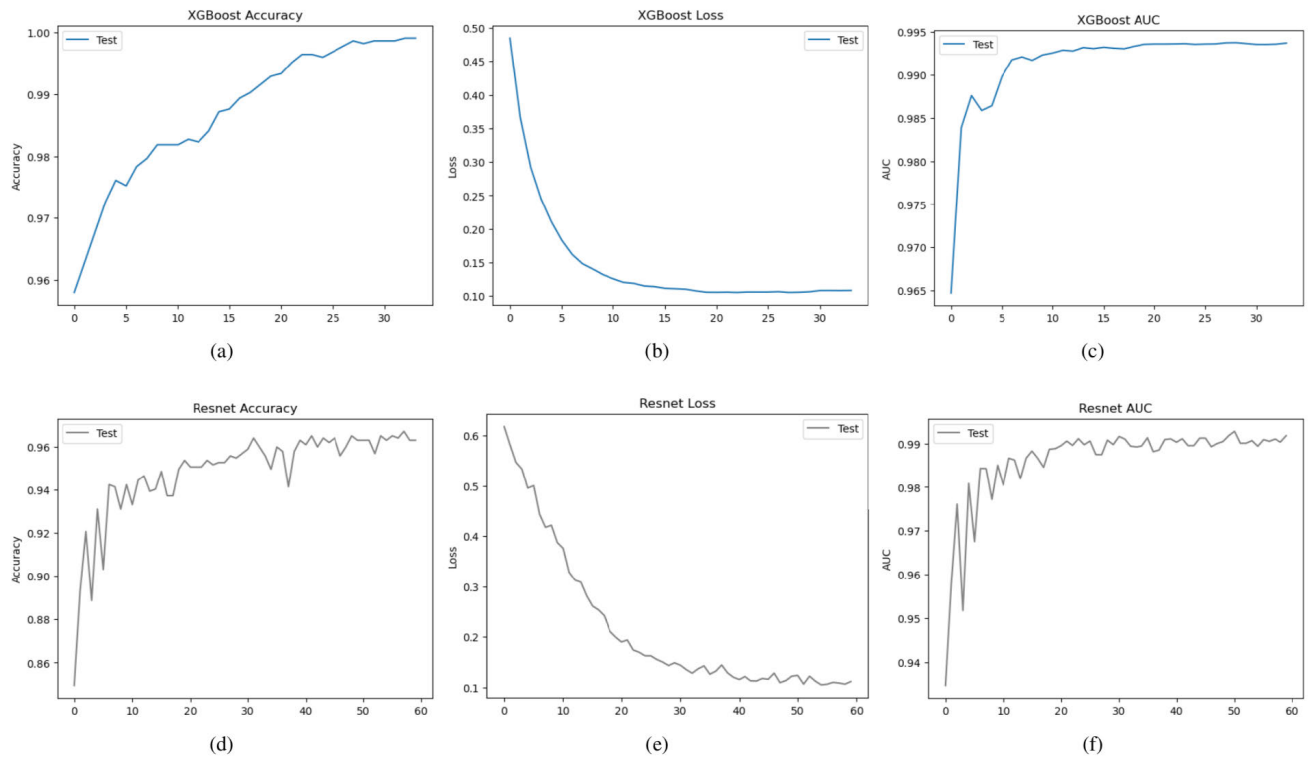


FIGURE 9. Evaluation results for the Graph Invariants XGBoost and Resnet-50 models. a), b), c) are representing results of XGBoost and d), e), f) are demonstrating Resnet-50 performance.

TABLE 2. Comparison of the Accuracy, Precision, Recall and F1-Score for the different models used in the experiment.

Model	Accuracy	Precision	Recall	F1-Score
GB	95%	96%	94%	95%
ADA	95%	95%	94%	95%
LightGBM	96%	97%	94%	96%
NGBoost	95%	96%	92%	94%
GXGB	96%	96%	95%	96%
RES-50	96%	97%	95%	96%
COMB	99%	99%	99%	99%

50, we tested a hybrid method, an ensemble constructed from these two models. After obtaining the augmented dataset and dividing the data into training and test sets, the algorithms were trained in parallel - XGBoost on the computed graph invariants and ResNet-50 on the augmented images. After each model made class predictions, the probabilities were averaged and used as a final estimate of model performance. The results of the evaluation of this ensemble model are shown in Figure 10; the validation curve for the graph-invariant XGBoost model is displayed in Figure 11.

VI. DISCUSSION

In this section, we compare our proposed algorithms with the results of algorithms applied in earlier studies which used the same ALL_IDB1 dataset. We summarized the results of these

TABLE 3. Comparison of the model performances from the different papers based on ALL_IDB1 dataset.

Authors	Model	Accuracy
T.T.P. Thanh, et al. [11]	CNN	96.6%
K.K. Anilkumar, et al. [13]	Resnet-50	96.2%
M. Bukhari, et al. [20]	CNN	98.3%
K. Talukdar, et al. [21]	FR-CNN	96.5%
S. Shafique, et al. [22]	SVM	93.7%
I. Abunadi, et al. [23]	SVM/CNN	98.1%
S. Agaian, et al. [24]	SVM	94%
E. Chen, et al. [25]	YOLOv5s	97.2%
R. Khandekar, et al. [26]	YOLOv4	96.1%
S. S. Al-jaboriy, et al. [27]	ANN	97.1%
Proposed model	COMB	99.1%

comparisons in the Table 3. We used Accuracy metric for comparison for two reasons. Firstly, in ALL_IDB1 dataset we operated with balanced classes, so in this case we can use Accuracy as an evaluation metric similar to F1-Score. Secondly, not all papers used all of the evaluation metrics that we applied (F1-Score, Precision, Recall, TPR and FPR). Its apparent from the data in the Table 3 that the combined use of XGBoost and ResNet-50 might improve the quality of the image cells classification. It is also worth noting that all of the ALL_IDB1-based studies cited used the original (high resolution) images (up to 2592×1944).

For XGBoost training in the present work we used highly compressed images of size 32×32 as inputs, i.e.

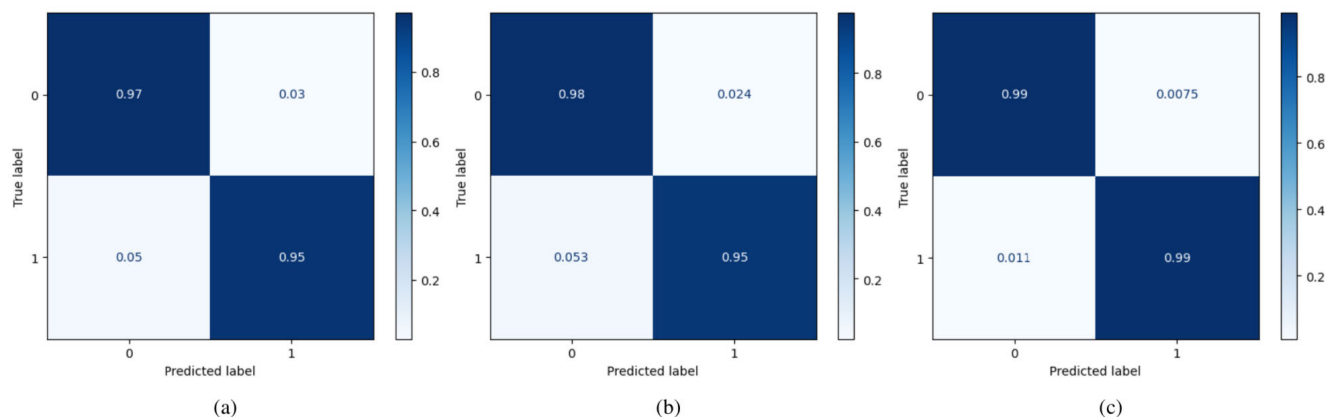


FIGURE 10. Confusion Matrices for both standalone models and the ensemble approach. a) Graph invariants XGBoost model; b) Resnet-50 model and c) Ensemble model.

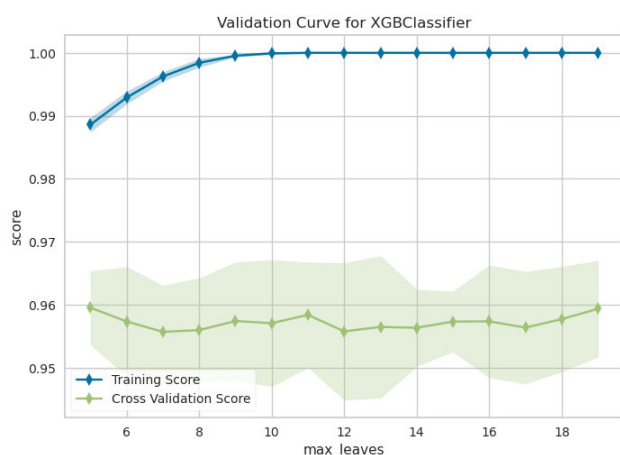


FIGURE 11. Validation Curve for the graph-invariant XGBoost model. F1-Score was used as evaluation score metric. Training and cross validation scores were displayed.

very low resolution images. It’s common knowledge that low resolution of input images significantly aggravates the performance of neural networks of complex architecture. The main advantage of our proposed method is that using the graph invariants we managed to increase the amount of useful information extracted during training on images of low quality and low dimensionality. In addition, our combined (ensemble) method might outperform similar approaches that use exclusively high-resolution images thereby showing the promise of further in-depth application of graph invariants in the task of medical image classification. To the best of our knowledge, this work is the first to use this type of weighted triangle-free graph construction to generate specific invariants based on edge weights, shortest paths and degrees of vertices and their neighbors for further training of machine learning models and subsequent enrichment and improvement of neural networks of complex architecture.

The major limitation of the proposed method is its relatively high computational complexity. The invariants related with the Maximum flow problem (such as Node Connectivity and Edge Connectivity) as well as the invariants based on the calculation of all-pairs shortest paths in the graph such as Edge Betweenness Centrality are the most computation-consuming with computational complexity exceeding $O(n^2 \log n)$. However, possibility of using the low-resolution images for generating synthetic features based on these invariants does alleviate the computation costs.

VII. CONCLUSION

In this study, we considered the problem of converting images into weighted graphs, calculating invariants, and trained several image classification algorithms. During the computational experiments on a real-world image dataset it was shown that the method proposed can improve the performance of the network architecture ResNet-50. Further steps of research may include analyzing different approaches to edge extraction in images and obtaining graphs from them (such as the use of Sobel filter [55] or Canny detector [56]), which might further improve the results along with more accurate detection of the edges and boundaries in the image [57]. Mathematical problems related to changes in graph structure as the image size increases (e.g., estimating the growth of connected components) seem to be a promising research direction and an interesting theoretical problem [58].

ACKNOWLEDGMENT

The authors would like to thank Natalia Petinati and Alexander Dainiak for their insightful discussions and valuable advices.

Author contributions: Dmitry Muzalevskiy: Initial idea, model and experiment design, conducting experiments, code, and initial paper preparation. Ivan Torshin: Research supervision, research ideas, paper editing, and revision.

REFERENCES

- [1] T. Terwilliger and M. Abdul-Hay, "Acute lymphoblastic leukemia: A comprehensive review and 2017 update," *Blood Cancer J.*, vol. 7, p. e577, Jun. 2017.
- [2] J. Hallbach, "Hamöblastosen," in *Klinische Chemie Und Hämatologie*, 3rd ed. New York, NY, USA: Thieme, 2011.
- [3] R. D. Brunning, "Classification of acute leukemias," *Semin Diagn Pathol*, vol. 3, pp. 142–153, Aug. 2003.
- [4] F. Miraki-Moud, F. Anjos-Afonso, K. A. Hodby, E. Griessinger, G. Rosignoli, D. Lillington, L. Jia, J. K. Davies, J. Cavenagh, M. Smith, H. Oakervee, S. Agrawal, J. G. Gribben, D. Bonnet, and D. C. Taussig, "Acute myeloid leukemia does not deplete normal hematopoietic stem cells but induces cytopenias by impeding their differentiation," *Proc. Nat. Acad. Sci. USA*, vol. 110, no. 33, pp. 13576–13581, Aug. 2013.
- [5] F.-L. Huang, E.-C. Liao, and C.-L. Li, "Pathogenesis of pediatric B-cell acute lymphoblastic leukemia: Molecular pathways and disease treatments," *Oncol Lett.*, vol. 1, pp. 448–454, Jul. 2020.
- [6] N. Alkhayat, Y. Elborai, O. Al Sharif, M. Al Shahrani, O. Alsuhaibani, M. Awad, H. Elghezal, I. B.-A. Bouhajar, M. Alfaraj, E. Al Mussaed, F. Alabbas, and G. Elyamany, "Cytogenetic profile and FLT3 gene mutations of childhood acute lymphoblastic leukemia," *Clin. Med. Insights, Oncol.*, vol. 11, Jan. 2017, Art. no. 117955491772171.
- [7] R. E. Raskin and J. B. Messick, "Bone marrow cytologic and histologic biopsies: Indications, technique, and evaluation," *Veterinary Clinics North Amer., Small Animal Pract.*, vol. 42, no. 1, pp. 23–42, Jan. 2012.
- [8] G. Rose, H. Reinhard, and S. B. Kahwash, "Is this a blast? An illustrated practical review on peripheral blood smear examination in the paediatric patient," *Malays J. Pathol.*, vol. 42, no. 1, pp. 37–49, Apr. 2020.
- [9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [11] T. T. P. Thanh, "Leukemia blood cell image classification using convolutional neural network," *Int. J. Comput. Theory Eng.*, vol. 10, no. 2, pp. 54–59, Apr. 2018.
- [12] S. M. Abas, A. M. Abdulazeez, and D. Q. Zeebaree, "A YOLO and convolutional neural network for the detection and classification of leukocytes in leukemia," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 25, no. 1, p. 200, Jan. 2022.
- [13] K. K. Anilkumar, V. J. Manoj, and T. M. Sagi, "Automated detection of leukemia by pretrained deep neural networks and transfer learning: A comparison," *Med. Eng. Phys.*, vol. 98, pp. 8–19, Dec. 2021.
- [14] S. H. Kassani, "Classification of histopathological biopsy images using ensemble of deep learning networks," in *Proc. Conf. CASCON Conf. Centre Adv. Stud. Collaborative Res.*, 2019, pp. 92–99.
- [15] A. Muntasa and M. Yusuf, "Modeling of the acute lymphoblastic leukemia detection based on the principal object characteristics of the color image," *Proc. Comput. Sci.*, vol. 157, pp. 87–98, 2019.
- [16] R. Almeida, E. Kijak, S. Malinowski, Z. K. G. Patrocínio Jr., and S. J. F. Guimarães, "Graph-based image gradients aggregated with random forests," *Pattern Recognit. Lett.*, vol. 166, pp. 182–189, Feb. 2023.
- [17] Y. Li, R. Zhu, L. Mi, Y. Cao, and D. Yao, "Segmentation of white blood cell from acute lymphoblastic leukemia images using dual-threshold method," *Comput. Math. Methods Med.*, vol. 2016, pp. 1–12, 2016.
- [18] H. Inbarani, A. T. Azar, and J. G, "Leukemia image segmentation using a hybrid histogram-based soft covering rough K-means clustering algorithm," *Electronics*, vol. 9, no. 1, p. 188, Jan. 2020.
- [19] R. D. Labati, V. Piuri, and F. Scotti, "All-IDB: The acute lymphoblastic leukemia image database for image processing," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011.
- [20] M. Bukhari, S. Yasmin, S. Sammad, and A. A. Abd El-Latif, "A deep learning framework for leukemia cancer detection in microscopic blood samples using squeeze and excitation learning," *Math. Problems Eng.*, vol. 2022, pp. 1–18, Jan. 2022.
- [21] K. Talukdar, K. Bora, L. B. Mahanta, and A. K. Das, "A comparative assessment of deep object detection models for blood smear analysis," *Tissue Cell*, vol. 76, Jun. 2022, Art. no. 101761.
- [22] S. Shafique, S. Tehsin, S. Anas, and F. Masud, "Computer-assisted acute lymphoblastic leukemia detection and diagnosis," in *Proc. 2nd Int. Conf. Commun., Comput. Digit. Syst. (C-CODE)*, Mar. 2019, pp. 184–189.
- [23] I. Abunadi and E. M. Senan, "Multi-method diagnosis of blood microscopic sample for early detection of acute lymphoblastic leukemia based on deep learning and hybrid techniques," *Sensors*, vol. 22, no. 4, p. 1629, Feb. 2022.
- [24] S. Agaian, M. Madhukar, and A. T. Chronopoulos, "A new acute leukaemia-automated classification system," *Comput. Methods Biomechanics Biomed. Eng., Imag. Visualizat.*, vol. 6, no. 3, pp. 303–314, May 2018.
- [25] E. Chen and M. Y. Shalaginov. (2022). *Real-time Detection of Acute Lymphoblastic Leukemia Cells Using Deep Learning*. BioRxiv. [Online]. Available: <http://dx.doi.org/10.1101/2022.10.22.513362>
- [26] R. Khandekar, P. Shastry, S. Jaishankar, O. Faust, and N. Sampathila, "Automated blast cell detection for acute lymphoblastic leukemia diagnosis," *Biomed. Signal Process. Control*, vol. 68, Jul. 2021, Art. no. 102690.
- [27] S. S. Al-jaboriy, N. N. A. Sjarif, S. Chuprat, and W. M. Abdualлах, "Acute lymphoblastic leukemia segmentation using local pixel information," *Pattern Recognit. Lett.*, vol. 125, pp. 85–90, Jul. 2019.
- [28] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019.
- [29] C. Connolly and T. Fleiss, "A study of efficiency and accuracy in the transformation from RGB to CIELAB color space," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 1046–1048, Jul. 1997.
- [30] G. Qu, S. L. Wood, and C. Teh, "Wafer defect detection using directional morphological gradient techniques," *EURASIP J. Adv. Signal Process.*, vol. 2002, no. 7, pp. 686–703, Dec. 2002.
- [31] T. Zhang, "Triangle free graphs and their chromatic numbers," Dept. Math., Univ. Chicago, Chicago, IL, USA, Tech. Rep., 2008. [Online]. Available: <https://www.math.uchicago.edu/~may/VIGRE/VIGRE2008/REUPapers/Zhang.pdf>
- [32] M. E. J. Newman, "Paths," in *Networks: An Introduction*. Oxford, U.K.: Oxford Univ. Press, 2010, pp. 137–138.
- [33] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 11, pp. 3747–3752, Mar. 2004.
- [34] A. Raigorodskii, "Models of random graphs and their applications to the web-graph analysis," in *Proc. RuSSIR Inf. Retr.*, 2015, pp. 101–118.
- [35] M. L. Stanley, S. L. Simpson, D. Dagenbach, R. G. Lyday, J. H. Burdette, and P. J. Laurienti, "Changes in brain network efficiency and working memory performance in aging," *PLoS ONE*, vol. 10, no. 4, Apr. 2015, Art. no. e0123950.
- [36] J. A. de la Peña, I. Gutman, and J. Rada, "Estimating the Estrada index," *Linear Algebra Appl.*, vol. 427, no. 1, pp. 70–76, Nov. 2007.
- [37] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford Digit. Library Technol. Project, Stanford Univ., Stanford, CA, USA, Tech. Rep. SIDL-WP-1999-0120, 1999.
- [38] M. Eliahi, G. Raeisi, and B. Taeri, "Wiener index of some graph operations," *Discrete Appl. Math.*, vol. 160, no. 9, pp. 1333–1344, Jun. 2012.
- [39] H. N. Gabow, "A matroid approach to finding edge connectivity and packing arborescences," *J. Comput. Syst. Sci.*, vol. 50, no. 2, pp. 259–273, Apr. 1995.
- [40] Z. Galil, "Finding the vertex connectivity of graphs," *SIAM J. Comput.*, vol. 9, no. 1, pp. 197–199, Feb. 1980.
- [41] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.
- [42] J. Matousek, "Cutting cheaply using eigenvectors," in *Thirty Three Miniatures Mathematical and Algorithmic Applications of Linear Algebra*. Providence, RI, USA: American Mathematical Society, 2010, pp. 153–163.
- [43] D. U. Ozsahin, M. T. Mustapha, A. S. Mubarak, Z. S. Ameen, and B. Uzun, "Impact of feature scaling on machine learning models for the diagnosis of diabetes," in *Proc. Int. Conf. Artif. Intell. Everything (AIE)*, Aug. 2022, pp. 87–94.
- [44] D. A. Darling, "The Kolmogorov–Smirnov, cramer-von Mises tests," *Ann. Math. Statist.*, vol. 28, no. 4, pp. 823–838, Dec. 1957.
- [45] A. Ivanov and G. Riccardi, "Kolmogorov–Smirnov test for feature selection in emotion recognition from speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 5125–5128.
- [46] A. Alsahaf, N. Petkov, V. Shenoy, and G. Azzopardi, "A framework for feature selection through boosting," *Expert Syst. Appl.*, vol. 187, Jan. 2022, Art. no. 115895.

- [47] Y. Wang, Z. Pan, J. Zheng, L. Qian, and M. Li, "A hybrid ensemble method for pulsar candidate classification," *Astrophys. Space Sci.*, vol. 364, no. 8, pp. 1–14, Aug. 2019, Art. no. 139.
- [48] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 4768–4777.
- [49] G. Claeskens, J. R. Magnus, A. L. Vasnev, and W. Wang, "The forecast combination puzzle: A simple theoretical explanation," *Int. J. Forecasting*, vol. 32, no. 3, pp. 754–762, Jul. 2016.
- [50] Ž. Đ. Vujovic, "Classification model evaluation metrics," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 599–606, 2021.
- [51] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class AdaBoost," *Statist. Its Interface*, vol. 2, no. 3, pp. 349–360, 2009.
- [52] T. Hastie, R. Tibshirani, and J. Friedman, "Boosing and additive trees," in *The Elements of Statistical Learning*, 2nd ed. Cham, Switzerland: Springer, 2008, pp. 360–361.
- [53] K. Guolin, Q. Meng, and T. Finley, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1–9.
- [54] T. Duan, A. Avati, and D. Y. Ding, "NGBoost: Natural gradient boosting for probabilistic prediction," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, vol. 119, pp. 1–11.
- [55] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE J. Solid-State Circuits*, vol. SSC-23, no. 2, pp. 358–367, Apr. 1988.
- [56] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [57] R. Sun, T. Lei, Q. Chen, Z. Wang, X. Du, W. Zhao, and A. K. Nandi, "Survey of image edge detection," *Frontiers Signal Process.*, vol. 2, pp. 1–13, Mar. 2022, Art. no. 826967.
- [58] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognit.*, vol. 70, pp. 25–43, Oct. 2017.



DMITRY MUZALEVSKIY was born in 1989. He received the bachelor's degree in applied informatics from Bauman Moscow State Technical University, and the master's degree in discrete mathematics from Moscow Institute of Physics and Technology. He is an Independent Researcher, Berlin, Germany. His research interests include medical image processing and analysis, algorithms on graphs and its applications for bioinformatics, and statistical learning theory.



IVAN TORSHIN was born in 1972. He received the Ph.D. degree in physics and mathematics, and the Ph.D. degree in chemistry. He is currently an Associate Professor with Moscow Institute of Physics and Technology; a Lecturer with the Faculty of Computational Mathematics and Cybernetics, Moscow State University; a Senior Researcher with the Federal Research Center "Computer Science and Control," Russian Academy of Sciences; and a Researcher with the Big Data Storage and Analysis Center, Moscow State University. He is the author of over 700 research articles published in journals on applied math, informatics, medicine, chemistry, and biology. He is also the author of ten monographies, of which six are in Russian and four in English (in the series "Bioinformatics in the Post-Genomic Era," Nova Biomedical Publishers, New York, 2006–2009).

...