# Generative Adversarial Optimization (GOA) for Acute Lymphocytic Leukemia Detection

**Milan TUBA*, Eva TUBA**

Singidunum University, Danijelova 29, Belgrade, 11000, Serbia
tuba@ieee.org (**Corresponding author*), etuba@ieee.org

**Abstract:** Computer-aided diagnostic systems represent an active research topic that captures the attention of scientists from various fields. Due to the complexity of human body and the various sources that can be used to generate medical digital images, numerous methods designed for analysing and detecting of different diseases have been developed. In this paper the focus is on detecting acute lymphocytic leukemia in microscopic digital blood images. A method for lymphocyte classification as normal cells or blasts is proposed. For purpose of classification, each cell is described by 11 features, namely 5 shape and 6 texture features. The support vector machine was used as a classifier and its parameters were tuned by means of a novel optimization algorithm: the generative adversarial optimization (GAO) algorithm. No other applications of this optimization algorithm – the latest so far - were found in literature. The proposed method for acute lymphocytic leukemia detection is compared with the Naïve Bayes classificatory, the k-nearest neighbour, the back propagation neural network, and the support vector machine optimized by the bare bones fireworks algorithm. The proposed GAO-based method achieved higher classification accuracy which exhibits the great potential of this new algorithm.

**Keywords:** Generative adversarial optimization, Swarm intelligence, Generative adversarial networks, Acute lymphocytic leukemia, Local binary pattern, Classification, Support vector machine, Parameter tuning, Optimization.

## 1. Introduction

Digital images have brought a revolution in many areas, from everyday life to various scientific fields. Due to various sources that can be used for obtaining digital images (e.g. X-rays, infrared, magnetic resonance, ultrasound, etc.) and relatively fast and easy processing methods, digital images have found their purpose in unexpected areas such as agriculture, astronomy, biology, surveillance, autonomous vehicles, medicine, and many more. Besides finding new fields where images can be useful, digital images have drastically changed the way how the images are used in certain areas. One of the best examples is medicine where images of different body parts and tissues generated by Xrays have been used for more than a century but nowadays, with digital images, the whole new scientific area is created. The analysis of analog medical images was a long, tedious and error prone process performed by human experts which implies that the quality of the final results largely depended on the current psychophysical state of the experts, their visual acuity, concentration, etc. Digital images enable automatic analysis which significantly speeds up the process and the results are completely objective. Besides the speed, good software can also detect the subtle changes in body tissues that can be hardly visible or even completely invisible to the naked human eye, i.e. to detect anomalies in early stages which is the most important factor for successful treatment of the patient.

Since it is undeniable that the usage of digital images in medicine is highly beneficial, various methods for detecting different diseases and anomalies in medical digital images were presented during past decades. It should be mentioned that, besides using digital images from sources that have already been used in medicine, new technologies enabled even more sources of medical images (e.g. wireless endoscopic capsule, elastography, photoacoustic imaging).

In this paper, the focus is on leukemia (blood cancer) detection in microscopic digital images of blood. Leukemia detection is one of the widely studied topics and various methods were proposed in the past (Kumar, Mishra, & Asthana, 2018; Renuga et al., 2016; Singhal, & Singh, 2016). Blood cancer can be diagnosed by analyzing microscopic images of white blood cells. A large number of blasts, or immature white blood cells, characterizes leukemia. Based on the rate of disease progression and the place where the blasts have been developed, four different types of leukemia can be distinguished: acute myeloid leukemia (AML), chronic myeloid leukemia (CML), acute lymphocytic leukemia (ALL) and chronic lymphocytic leukemia (CLL). In this paper, a method for detection of ALL is presented. ALL is manifested by a large number of blasts produced in bone marrow which means that numerous white blood cells are not fully formed hence they are not capable of subduing infections. Early ALL detection is very important since it spreads fast to the brain, liver and other organs.

https://doi.org/10.24846/v28i3y201901

ALL detection methods are usually based on the classification of white blood cells to normal or cancer cells. Two important steps are the extraction of appropriate features from microscopic digital images of blood cells and the classification. The feature extraction step uses standard digital image processing techniques, but it is important to find what are well discriminating features. Besides choosing a good feature set, it is important to have efficient classifier that will provide the best accuracy. The current study makes use of the support vector machine (SVM), which is one of the most efficient classification methods, provided that it has been properly adjusted to the problem under consideration. As the SVM with radial basis function (RBF) was used as a kernel function, two parameters of the SVM had to be tuned. The first parameter is soft margin parameter $C$ and the second one is RBF free parameter $\gamma$. Accuracy of the SVM model depends on the appropriate selection of these two parameters. However, the accuracy function is characterized by very irregular and unpredictable surface with numerous local optima, which makes this SVM parameter selection problem a hard optimization problem that cannot be solved by deterministic methods.

In the past decades, hard optimization problems have been successfully solved by various nature inspired algorithms where swarm intelligence algorithms represent large and widely used subcategory. These algorithms have been also applied to image processing and machine learning problems such as the retinal blood vessel segmentation (Tuba, Mrkela & Tuba, 2017), retinal image registration (Tuba, Tuba & Dolicanin, 2017), design of passive suspension systems by artificial neural networks (Niresh, Archana, & Anand, 2019), support vector machine optimized for erythemato-squamous diseases detection (Tuba et al., 2017), etc.

This paper sets forth the first practical application of new population-based optimization algorithm presented recently (only a month ago), namely a generative adversarial optimization algorithm (GAO). The GAO represents a novel optimization method that was inspired by the generative adversarial network proposed by Goodfellow et al. in 2014. Tan and Shi in 2019 proposed the GAO where generative models were used to generate candidate solutions by an adversarial process. This algorithm represents a novel optimization method where exploration and exploitation techniques are extremely different from all other algorithms proposed so far. Based on the results reported in Tan and Shi, (2019), the GAO is one of the current state-of-the-art optimization methods.

The rest of this paper is organized as follows. Section 2 gives a short literature review of leukemia detection methods based on optimization metaheuristics. Section 3 gives the details about the used dataset, features and short SVM explanation. The generative adversarial network is explained in Section 4, while the generative adversarial optimization algorithm is given in Section 5. Simulation results and comparison with other leukemia detection methods are presented in Section 6. This paper is concluded in Section 7 where plans for future works are also given.

## 2. Previous Work

Acute lymphocytic leukemia detection is a widely researched topic. Numerous methods that use different machine learning and image processing techniques have been proposed in the past. Since the texture of the white blood cells represents a rather important feature, Singhal & Singh (2016) proposed two different texture descriptors to be used, grey level co-occurrence matrix and local binary pattern, and based on the results it was concluded that the local binary pattern is better texture descriptor for differentiation of normal blood cells from blasts.

Mohapatra, Patra & Satpathy (2014) proposed a computer-aided diagnostic system for ALL detection where each lymphocyte was described by 17 shape or size features, 15 texture and 12 color features. Some of the features were areas of lymphocyte elements, their ratios, compactness, roundness, wavelet texture features, mean of different color components, etc. Multiple-classifier system was used for classification and the proposed method achieved better results compared to Naïve Bayesian, knearest neighbor, multilayer perceptron, and radial basis network.

Lymphocyte segmentation by the k-means algorithm was proposed by Amin et al. (2015). Amin et al. (2015) extracted 72 geometrical and statistical features and the number of features was reduced to 15 by feature selection method based on discriminative analysis. The quality of the proposed method was tested for detecting ALL as well as different stages of the ALL.

The classification has a very important role in ALL detection methods. Proper choice of the classification method and tuning of the classifier's parameters can significantly improve the ALL detection accuracy. The support vector machine has been proved to be a good choice for classifier if the parameters are tuned properly. The SVM parameter tuning represents a hard optimization problem and a separate research topic. Numerous optimization metaheuristics have been proposed for the SVM parameter tuning problem. Since it is a hard optimization problem, swarm intelligence algorithms such as fruit fly optimization (Shen et al., 2016), fireworks algorithm (Tuba, Tuba & Beko, 2016), artificial bee colony (Yang et al., 2015), elephant herding optimization algorithm (Tuba & Stanimirovic, 2017), bat algorithm (Tuba, Tuba & Simian, 2016) have been commonly used.

This paper presents the first application of the most recent optimization metaheuristic, the generative adversarial optimization, for solving the SVM parameter tuning problem.

## 3. Feature Extraction and Classification

In this paper a method for ALL detection based on the microscopic images of blood is proposed. ALL can be diagnosed by analysing white blood cells or lymphocytes. In order to classify lymphocyte as normal or blast (leukemia cell), it is necessary to extract them from the image. After detecting cells and their elements, i.e. nuclei and cytoplasm, features should be extracted. Specialists use color, shape and texture features when analyzing the cell so those elements are included in the proposed computer-aided diagnostic system as well.

Color features are used for extracting cell elements. Lymphocyte segmentation was done in the HSI and CIELab color models. These color spaces are used since the nucleus and cytoplasm can be clearly distinguished which is not the case if the RGB model is used. Based on the empirical tests guided by methods from literature, it has been established that the nucleus of a lymphocyte can be extracted in microscopic images by using the saturation component from HIS color space. For nucleus extraction, the thresholding technique was used. Based on the previous work presented in (Tuba et al., 2019) and repeated experiments, it was established that the best threshold value for extracting the nucleus is 0.35 when the saturation range is [0,1].

The whole blood cell, i.e. lymphocyte, can be extracted by applying a threshold to the luminance component in CIELab color space. After normalization of the luminance component to the range [0, 1], the threshold value used for cell extraction was set to 0.5. In order to get better segmentation, after applying the threshold, morphological operation of opening with the window size 10 was applied.

After extracting the nucleus and cell, the cytoplasm can be obtained by the simple subtraction of nucleus from the cell.

Most important features and the main indicator of the blast are the cell's shape features (Mishra et al., 2017). In this paper, the authors used five shape related features based on area, perimeter and circularity of cell, nucleus and cytoplasm. Since blasts represent not fully formed white blood cells, they are usually not rounded as the healthy cell and their nuclei are smaller, i.e. more cytoplasm is present. Object area is defined by the number of pixels that the object contains. The perimeter measures the maximal distance between two disjoint pixels of the object. Circularity can be obtained by using area and perimeter and it is defined as follows:

$$circ = \frac{\frac{4}{\pi}A}{P^2}, \tag{1}$$

where $A$ represents the area of the object and $P$ is its perimeter. Circularity is used to measure the roundness of the object and it this paper it is used since healthy cells are more rounded than blasts.

Five shape-based features that are used to describe a lymphocyte are the area of the cell, perimeter of the nuclei, a circularity of the nuclei, rations of the areas cytoplasm to nuclei and nuclei to cell.

Besides features that are based on the shape of the cell and its elements, texture features are rather important. The current paper proposes the usage of the local binary pattern (LBP) for describing the texture. The LBP is a commonly used texture descriptor in various applications including computer-aided diagnostic systems (Stojak, Tuba & Tuba, 2016; Mishra et al., 2017; Tuba et al., 2019). The LBP uses 8neighborhood and defines a binary pattern for each pixel from the neighbor pixels. The binary pattern is obtained by thresholding pixels inside the window 3x3 with the central pixel so that all pixels around the center are set to 0 or 1. By reading obtained binary values, the binary pattern is determined. In literature, versions with different neighborhood structures can be found. Another modification of the LBP is proposed with the aim to make it rotation invariant. In the case of ALL detection, texture descriptor should be rotation invariant so the uniform LBP is used in this paper instead of the original one. The uniform LBP reduces the number of patterns that are considered different. All patterns obtained by the original LBP that have more than two transitions from 0 to 1 or from 1 to 0 are grouped as single pattern. In this case, from 256 patterns, uniform LBP recognizes only 59 different patterns. After computing LBP for all pixels, a histogram of patterns is made. As features for classification, statistical data such as central moments, energy and entropy are used. In this paper for each cell, six measures of the LBP histogram were used, that is mean and standard deviation, skewness, kurtosis, energy and entropy.

After extracting all features, they were used for training the support vector machine (SVM). SVM represents a binary classifier that separates instances from different classes by a hyperplane. Instances are considered as points in *k*dimensional space where *k* is the number of features that are used for describing the instance. In this paper *k* is equal to 11 because 5 shape features and 6 texture features have been used.

The support vector machine is a widely used classifier since it can achieve high accuracy, but only if it is tuned for the considered dataset. Parameter tuning of the SVM is a hard optimization problem since accuracy is a multimodal function of two parameters, soft margin *C* and kernel function's parameter γ. Soft margin parameter *C* has been introduced in the SVM definition to enable classification of real-world datasets that commonly contain outliers, instances that are significantly different from other instances from the same class. By introducing parameter *C*, it is possible to allow for wrongly classified data and *C* controls the cost of these instances. If *C* is set to large values, SVM model is close to the hard margin model where each instance has to be on the right side of the hyperplane. The opposite case, involving low values of parameter *C* leads to the model where wrongly classified data are almost completely ignored. This aspect results in poor classification accuracy which is similar to the random assignment of classes to instances.

Kernel function has been used in cases where instances are not linearly separable as is usually the case. Kernel function replaces a dot product which results in mapping instances into higher search space where they become linearly separable and the SVM model can be used successfully. One of the most common kernel functions is radial biases function (RBF):

$$K(x,x') = \exp[-\gamma \|x-x'\|^2], \qquad (2)$$

where γ is a free parameter that needs to be tuned. SVM parameters *C* and γ cannot be set separately but the best combination of values for these two parameters has to be found.

## 4. The Generative Adversarial Networks

The generative adversarial networks (GAN) model was presented in 2014 by Goodfellow et al. (Goodfellow et al. 2014). The concept of the GAN is based on the minimax two-player game in the game theory and it contains two neural networks named generator *G* and discriminator *D*. The goal of the generator is to generate samples that will be as realistic as possible so the discriminator cannot recognize the difference, i.e. to decrease the accuracy of the discriminator. On the other hand, the discriminator should distinguish real and fake data. The generator tries to estimate the distribution of the given sample data. It generates new data that define probability distribution $p_g$ and the goal is for $p_g$ to converge to the distribution of the sample data $p_{data}$. The discriminator can be understood as a function that maps given samples to the set {0,1} depending on whether the sample is from the real or generated set. In reality, the discriminator returns the probability that the reviewed sample is from the real dataset. When the generator produces solutions that match the real sample distribution, the discriminator is not able to spot the difference hence all samples will have probability around 0.5 of belonging to real/fake data.

The training process consists in the alternated minimization of the generator and discriminator. While the parameters of one of the networks are fixed, the other tries to minimize the error rate. The generator and discriminator try to optimize different and opposing loss functions while using the knowledge of the other network. The GAN structure is presented in Figure 1.

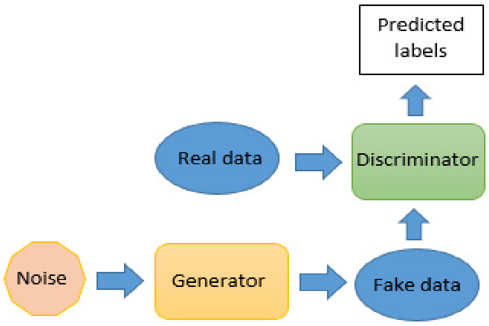

**Figure 1.** The structure of the generative adversarial network

The generative adversarial network represents a recent method but it has already been widely used in image and video processing applications for text to photo-realistic image synthesis (Zhang et al., 2017), the problems of simultaneous color and depth image super-resolution in 3D video (Zhao et al., 2019), real time texture synthesis (Li & Wand, 2016), natural language processing (Saito, Takamichi, & Saruwatari, 2017) and more.

## 5. The Generative Adversarial Optimization Algorithm

The generative adversarial optimization (GAO) algorithm is a novel optimization metaheuristic proposed by Tan and Shi, (2019). Beside the original paper where the GAO algorithm was proposed, tested on benchmark functions for continuous optimization problems CEC2013 and compared with other state-of-the-art methods, no other papers that use this algorithm were found in literature, hence the first application of the GAO algorithm is proposed in this paper. Besides being probably the latest optimization algorithm, the GAO introduced a radically different method for generating new solutions compared to the existing metaheuristics that use random sampling or guiding vectors (Tan & Shi, 2019; Li, Zheng & Tan, 2016). In the GAO, new solutions are generated based on the GAN, i.e. generator is used for finding a guiding vector for new solutions, while the discriminator is trained to predict whether the generated solutions will be better compared to the current ones. Based on the discriminator's output, the guiding vector for generating new and better solutions is calculated. Feedback from the discriminator is used to calculate gradients that are used for training the generator. The GAO algorithm's structure is given in Figure 2.

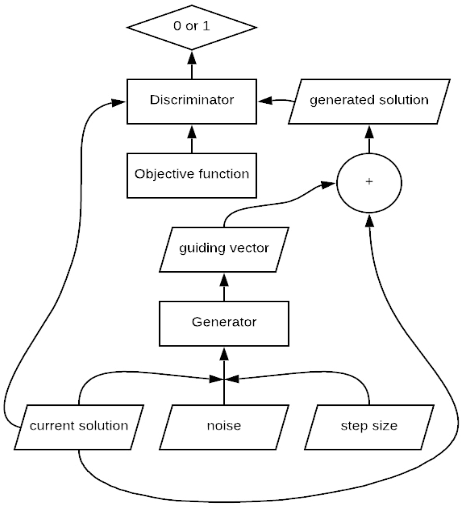

**Figure 2.** The GAO architecture

The generator, discriminator and their connection in the GAO algorithm are as follows. The generator $G$ uses as input the current solution $x_c$, a noise vector $z$ and the step size $l$, while the output is the guiding vector $g$ that is used to generate new solutions based on the current population. Mathematical representation of obtaining the guiding vector is (Tan & Shi, 2019):

$$g = G(x_c, z, l). \tag{3}$$

The architecture of the generator is given in Figure 3.

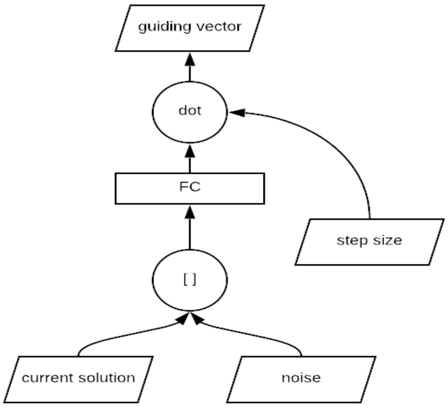

**Figure 3.** The generator in the GAO algorithm

Guiding vector is obtained by the following steps. Firstly, the current solution and the noise are concatenated and the result of the concatenation is used to feed a fully-connected layer ($FC$). The second step is combing the output of the $FC$ with the step size, which is the algorithm's parameter. Guided vector is a dot product of the step size and $FC$ output:

$$g = G\left(x_c, z, l\right) = FC\left(\left[x_c^T, z^T\right]^T\right) \cdot l. \tag{4}$$

The new solution is obtained by adding the guiding vector to the current solution (Tan & Shi, 2019):

$$x_g = x_c + g. \tag{5}$$

The discriminator *D* uses the current solution and the generated one to produce the prediction whether the new solution is better than the current one. Labels needed for training the discriminator are determined by the following equation:

$$y^i = \begin{cases} 1, & if(x_g) < f(x_c) \\ 0, & else \end{cases}. \tag{6}$$

As it can be seen in Figure 4 where the discriminator architecture is presented, the *D* consists of two fully-connected layers, $FC_1$ and $FC_2$.

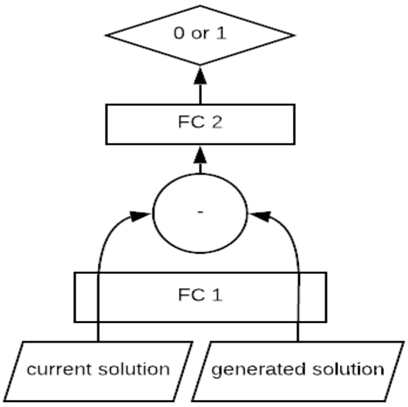

**Figure 4.** The discriminator in GAO algorithm

The current and generated solution, $x_c$ and $x_g$, are given to $FC_1$ and the output of $x_g$ is subtracted from $x_c$. The result of subtraction is used for $FC_2$ that finally produces prediction if the generated solution is better than the current one. Mathematical notation of the prediction is:

$$\begin{aligned} p &= D(x_c, x_g) \\ &= FC_2\left(FC_1(x_c) - FC_1(x_g)\right). \end{aligned} \tag{7}$$

## 5.1. The Training Process in the GAO

The GAO algorithm, as an iterative population-based optimization method, starts with generating solution set $C = \{x_c^i\}$, *i=1,2,…,n,* that contains *n* randomly generated solutions inside the search space. Before iterations start, it is necessary to determine the step size *l* and the stopping criterion. In (Tan & Shi, 2019), the termination criterion was defined by the number of fitness function evaluations and it was set to *maxFES=10.000*dim*, where *dim* represents the problem dimension, i.e. the size of the solution vector. In each iteration, the first step is to generate β new solutions based on the solutions in solution set *C* which is followed by choosing *n* solutions from all solutions, current and newly generated, to be kept for the next iteration. The training process of the GAO algorithm is presented in Algorithm 1.

**Algorithm 1.** Pseudo-code of the training process in the GAO algorithm (Tan & Shi, 2019)

```
Input:
    $n$ – population size,
    β – number of solution that will be generated in each iteration,
    $l_{init}$ – initial value of the step size $l$
Initialization
Randomly sample $n$ solutions in the search space for the solution set $C$.
Evaluate each solution $x_c^i$ in the set $C$ by fitness function $f(x_c^i)$.
Initialize the step size $l=l_{init}$.
repeat
    Generate β solutions and train discriminator $D$.
    Train generator $G$ with fitted $D$.
    Select $n$ solutions for the next iteration among the solutions in the set $C$ and set of β generated solutions.
    Reduce step size $l$.
until the stopping criterion is met.
```

The discriminator *D* is used to evaluate whether the new solution, i.e. a solution $x_g$ generated by *G*, will be better than the current solution $x_c$. The triplet consisting of the current solution, generated solution and the label obtained by Eq. (4), $(x_c^i, x_g^i, y^i)$, is required as an input for the discriminator. For a given input $(x_c^i, x_g^i, y^i)$ the loss function of the discriminator *D* is defined as:

$$\begin{aligned} \max_D loss_D &= y^i \log(D(x_c^i, x_g^i)) \\ &+ (1 - y^i)\log(1 - D(x_c^i, x_g^i)). \end{aligned} \tag{8}$$

Since in the case of the GAO, the discriminator is trained with batches, the overall loss is calculated as the average loss for each triplet.

After training the discriminator, the generator *G* uses the output of the *D* as a guidance for generating the guiding vector for the solutions that are in the current solution set *C*. Based on the feedback from the discriminator, or more precisely, based on the gradients from the feedback of the discriminator, the generator is trained. The loss function of the generator is defined by the following equation:

$$\max_G loss_G = \log(D(x_c^i, x_c^i + G(x_c^i, z, l))), \tag{9}$$

where *z* represents random Gaussian noise while the parameter *l* is the step size. Again, in the case when a batch is used for training, the loss function is equal to the average loss for each triplet in the batch.

Solutions for the next iteration should be among solutions with better fitness function values, but exploration mechanism should also be provided. In (Tan & Shi, 2019) solution selection process is determined by calculating the probability for each solution to be selected, where probability depends on the rank of the solution. The probability of selection for each solution is calculated by the following equation:

$$p(x^i) = \frac{\gamma_{f(x^i)}^{-\alpha}}{\sum_{i=1}^{n+\beta} \gamma_{f(x^i)}^{-\alpha}}, \tag{10}$$

where γ defines the rank of the solution $x^i$ among all solutions based on the fitness function value and α represents the algorithm's parameter that is used to control the shape of the distribution. In practice, larger values of the parameter α mean that the probability of choosing fitter solutions is also larger.

The last step in each iteration is the step size reduction. The idea of generating a guiding vector comes from the algorithm proposed by one of the GAO authors, guided fireworks algorithm (GFWA) (Li, Zheng & Tan, 2016). In the GFWA, the guided vector is used to control the radius of the search in each iteration. In order to find a good solution for multimodal optimization problems, it is necessary that the search at the beginning of the algorithm covers a whole search space so as to find promising areas that will help to avoid being stuck in local optimum. At later stages of the algorithm, it can be assumed that the promising areas have been found and that these parts of the search space need to be examined more carefully, i.e. exploitation method is needed. It is necessary to have a good balance between exploration and exploitation which means that the initial step size should be large enough to cover the whole search space at the beginning and it should be gradually decreased as the iteration number grows. In the original paper where the GAO was presented (Tan & Shi, 2019), it has been tested with several methods of decreasing the step size. In (Tan & Shi, 2019) it was recommended that the initial step size should be equal to half of the search space radius, $l_{init}$*=search space radius/2*. In order to ensure gradual step size reduction, the iteration count is mapped to [ε, $l_{init}$] by a monotone function where ϵ is a small constant, in practical implementation set to be ε=$10^{-20}$. In (Tan & Shi, 2019) it was tested with exponential and power functions with various powers and it has been empirically determined that the best results were achieved when the power function with the power 4.5 was used. Step size in each iteration is defined by the following equation:

$$l = (l^{\frac{1}{pow}} - \Delta)^{pow}, \tag{11}$$

where *pow=4.5* and Δ is defined as:

$$\Delta = l_{init}{}^{1/pow} - \frac{l_{init}{}^{1/pow}}{currentIter}. \tag{12}$$

This paper puts forward the first application of the generative adversarial optimization algorithm. The GAO is used for tuning of the support vector machine parameters used for the classification of blood cells to normal and leukemia cells.

## 6. Simulation Results

The proposed computer-aided diagnostic system was implemented in Python 3.6. The proposed framework was adapted for execution on the graphical processing unit (GPU) using NVIDIA™ CUDA parallel computing in Python with Tensorflow 1.14.0 library. and all tests were performed on NVIDIA RTX 2060 GPU and machine with Intel ® Core™ i5-9600K CPU at 5GHz, 16GB RAM, Windows 10 Professional OS.

According to the literature, SVM parameters should be searched in log space, i.e. C in [$2^5$, $2^{15}$] and γ in the range [$2^{-15}$, $2^5$] (Chang & Lin, 2011). For the GAO algorithm, solutions are 2dimensional vectors where the first dimension has lower bound -5 and upper bound 15, while the second dimension is limited by -15 and 5. The other parameters for the GAO algorithm are set as follows. Population size, the number of solutions that are kept for the next generation, was $n$=5. The number of solutions that are generated in each generation is β=30. Control parameter for the selection process α=2 as it was recommended by Tan and Shi (2019). The discriminator and generator settings are given in Table 1.

**Table 1.** Structure of the generator and discriminator

| Parameter | *G* | *D* | |
|---|---|---|---|
| | FC | $FC_1$ | $FC_2$ |
| No. of hidden layers | 1 | 2 | 1 |
| No. of hidden layer nodes | 64 | 64, 64 | 10 |
| No. of output nodes | *dim* | 10 | 1 |
| Activation function of hidden layer | relu | relu,<br>relu | relu |
| Activation function of output layer | tanh | relu | sigmoid |

Additionally, to avoid overfitting of the discriminator *D* and generator *G*, two regularization techniques have been used: batch normalization on hidden layers and dropout with rate 0.5 is applied to the output layer.

The fitness function used for the considered problem was classification accuracy while the results were analyzed by standard metrics used in machine learning for classification problems. The metrics were chosen to ensure a fair comparison with other methods from literature (Mishra et al., 2017; Tuba et al., 2019). For measuring the quality of the SVM model, sensitivity, specificity and accuracy were used. In order to calculate these metrics, it is necessary to build a confusion matrix that counts the number of true positive, true negative, false positive and false negative instances in the final classification. True positive/negative instances are instances that were correctly classified as normal/leukemia cells. False positive instances are leukemia cells classified as normal while false negative are normal cells wrongly classified as leukemia. After building the confusion matrix, sensitivity and specificity are defined as:

$$sensitivity = \frac{TP}{TP+FN}, \tag{13}$$

$$specificity = \frac{TN}{TN+FP}. \tag{14}$$

Finally, classification accuracy is defined as:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \tag{15}$$

In order to test the quality of the proposed GAO based method for ALL detection, the obtained results were compared with the method proposed by Mishra et al. (2017) and with the previous work of (Tuba et al. 2019). Mishra et al. (2017) proposed a method for ALL detection based on shape features combined with DCT coefficients. Their feature extraction process included image enhancement by applying Weiner filter for noise removal followed by a watershed segmentation process. Non-optimized support vector machine was used for classification and the results were compared to classification by Naive Bayes classifier (NC), knearest neighbors (KNN) and backpropagation neural network (BPNN). All the results reported by Mishra et al. (2017) are also presented in this paper. Tuba et al. (2019) used the same set of features, but the support vector machine was optimized by the bare bones fireworks algorithm (Li & Tan, 2018). The image dataset that was used is ALL-IDB2 dataset (Labati, Piuri & Scotti, 2011) that contains 130 blast and 130 normal lymphocyte images. Table 2 presents the confusion matrix obtained by the SVM optimized by the GAO algorithm.

**Table 2.** Confusion matrix obtained by the proposed GAO method

| Predicted/Actual | Normal | Blasts |
|---|---|---|
| Normal | 125 | 5 |
| Blast | 11 | 119 |

The confusion matrix was used to compute the sensitivity, the specificity and the accuracy of the model. These results, achieved by the proposed algorithm, are presented in the last column in Table 3, along with the results presented by Mishra et al., (2017) and Tuba et al., (2019). The best results are printed in bold.

**Table 3.** Comparison of the results

| | NB | KNN | BPNN | DCT | BBFWA | GAO |
|---|---|---|---|---|---|---|
| Sen. | 88.70 | 98.38 | 80.64 | 84.67 | 94.21 | **96.15** |
| Spec. | 75.38 | 69.23 | 37.67 | **94.61** | 89.37 | 91.15 |
| Acc. | 81.66 | 83.46 | 58.70 | 89.76 | 91.84 | **93.84** |

Based on the results presented in Table 3, it can be concluded that the proposed GAO-based method achieved better accuracy compared to all other methods presented by Mishra et al. (2017) and Tuba et al. (2019). Sensitivity was also the best for the case where the proposed GAO-based classification method was used. Same as in Tuba et al. (2019), specificity was lower compared with the Mishra et al. (2017) method which can be the consequence of the chosen features. Lower specificity means more blasts classified as normal lymphocytes.

The previous table demonstrates improvements introduced by the GAO-based method with the proposed set of features. To examine the contribution of the GAO algorithm alone, the simulation was repeated by the proposed GAO algorithm, but using the same set of feature as in (Tuba et al., 2019), where the BBFWA was used. The proposed GAO method, with the same features used in Tuba et al. (2019), achieved a sensitivity 94.61%, a specificity 90.00% and a classification accuracy 92.31%, which improved the results presented in (Tuba et al., 2019): sensitivity for 0.40%, specificity for 0.63% and classification accuracy for 0.47%. This demonstrates a great potential of the novel GAO algorithm since in its first practical application, without any modifications, it outperformed well established state-of-the-art BBFWA.

## 7. Conclusion

This paper proposes the first application of the recent generative adversarial optimization algorithm. The GAO was used for finding the optimal values for the support vector machine parameters used for the classification of the lymphocytes. For the purpose of classification, five shape features and 6 texture features have been used and the proposed GAO-based method achieved classification accuracy of 93.84% which is better compared to other methods presented in scientific literature. Future work could include improvements of the classification accuracy by adding more features, including feature selection and by modifying the generative adversarial optimization algorithm. Also, deep learning methods such as convolutional neural networks could be used directly on images with the GAO algorithm for architecture and parameter selection.

## Acknowledgements

Mr. M. Tuba’s research work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, Grant No. III-44006.

## REFERENCES

1. Amin, M. M., Kermani, S., Talebi, A. & Oghli, M. G. (2015). Recognition of Acute Lymphoblastic Leukemia Cells in Microscopic Images Using k-Means Clustering and Support Vector Machine Classifier, *Journal of Medical Signals and Sensors*, *5*(1), 49-58.

2. Chang, C. C. & Lin, C. J. (2011). LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), p. 27.

3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. & Bengio, Y. (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* (*NIPS 2014*) (pp. 2672-2680).

4. Kumar, S., Mishra, S. & Asthana, P. (2018). Automated Detection of Acute Leukemia Using k-mean Clustering Algorithm, *Advances in Computer and Computational Sciences*, 655-670. Springer.

5. Labati, R. D., Piuri, V. & Scotti, F. (2011). All-IDB: The Acute Lymphoblastic Leukemia Image Database for Image Processing. In *18th International Conference on Image Processing* (pp. 20452048). IEEE.

6. Li, C. & Wand, M. (2016). Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. In *European Conference on Computer Vision* (pp. 702-716). Springer.

7. Li, J. & Tan, Y. (2018). The Bare Bones Fireworks Algorithm: A Minimalist Global Optimizer, *Applied Soft Computing*, *62*, 454-462.

8. Li, J., Zheng, S. & Tan, Y. (2016). The Effect of Information Utilization: Introducing a Novel Guiding Spark in the Fireworks Algorithm, *IEEE Transactions on Evolutionary Computation*, *21*(1), 153166.

9. Mishra, S., Sharma, L., Majhi, B. & Sa, P. K. (2017). Microscopic Image Classification Using DCT for the Detection of Acute Lymphoblastic Leukemia (ALL). In *Proceedings of International Conference on Computer Vision and Image Processing* (pp. 171-180). Springer, Singapore.

10. Mohapatra, S., Patra, D. & Satpathy, S. (2014). An Ensemble Classifier System for Early Diagnosis of Acute Lymphoblastic Leukemia in Blood Microscopic Images, *Neural Computing and Applications*, *24*(78), 1887-1904.

11. Niresh, J., Archana, N. & Anand, R. (2019). Optimisation of Linear Passive Suspension System Using MOPSO and Design of Predictive Tool with Artificial Neural Network, *Studies in Informatics and Control*, *28*(1), 105-110. DOI: 10.24846/v28i1y201911

12. Renuga, V., Sivaraman, J., Kumar, S. V., Sathish, S., Padmapriya, P. & Reethu, R. (2016). Blood Microscopic Image Analysis for Acute Leukemia Detection, *International Journal of Control Theory and Applications*, *9*(9), 3731-3735.

13. Saito, Y., Takamichi, S. & Saruwatari, H. (2017). Statistical Parametric Speech Synthesis Incorporating Generative Adversarial Networks, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(1), 84-96.

14. Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H. & Liu, D. (2016). Evolving support vector machines using fruit fly optimization for medical data classification, *Knowledge-Based Systems*, *96*, 61-75.

15. Singhal, V. & Singh, P. (2016). Texture features for the detection of acute lymphoblastic leukemia. In *Advances in Intelligent Systems and Computing: International Conference on ICT for Sustainable Development* (pp. 535-543). Springer.

16. Stojak, A., Tuba, E. & Tuba, M. (2016, November). Framework for Abnormality Detection in Magnetic Resonance Brain Images. In *2016 24th Telecommunications Forum (TELFOR)* (pp. 1-4). IEEE.

17. Tan, Y. & Shi, B. (2019, July). Generative Adversarial Optimization. In *Lecture Notes in Computer Science on Advances in Swarm Intelligence*, *LCNS 11655* (pp. 3-17). Springer, Cham.

18. Tuba, E., Mrkela, L. & Tuba, M. (2017, April). Retinal Blood Vessel Segmentation by Support Vector Machine Classification. In *2017 27th International Conference Radioelektronika* (pp. 1-6). IEEE.

19. Tuba, E., Ribic, I., Capor-Hrosik, R. & Tuba, M. (2017). Support Vector Machine Optimized by Elephant Herding Algorithm for Erythemato-Squamous Diseases Detection, *Procedia Computer Science*, *122*, 916923. Elsevier.

20. Tuba, E. & Stanimirovic, Z. (2017, June). Elephant Herding Optimization Algorithm for Support Vector Machine Parameters Tuning. In *9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)* (pp. 1-4). IEEE.

21. Tuba, E., Strumberger, I., Bacanin, N., Zivkovic, D. & Tuba, M. (2019). Acute Lymphoblastic Leukemia Cell Detection in Microscopic Digital Images Based on Shape and Texture Features. In *Lecture Notes in Computer Science on Advances in Swarm Intelligence, LNCS* (pp. 142-151). Springer, Cham.

22. Tuba, E., Tuba, M. & Beko, M. (2016). Support Vector Machine Parameters Optimization by Enhanced Fireworks Algorithm. In *Lecture notes on Computer Science on Advances in Swarm Intelligence, LCNS 9712* (pp. 526-534). Springer, Cham.

23. Tuba, E., Tuba, M. & Dolicanin, E. (2017). Adjusted Fireworks Algorithm Applied to Retinal Image Registration, *Studies in Informatics and Control*, *26*(1), 33-42. DOI: 10.24846/v26i1y201704

24. Tuba, E., Tuba, M. & Simian, D. (2016). Adjusted Bat Algorithm for Tuning of Support Vector Machine Parameters. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2225-2232).

25. Yang, D., Liu, Y., Li, S., Li, X. & Ma, L. (2015). Gear Fault Diagnosis Based on Support Vector Machine Optimized by Artificial Bee Colony Algorithm, *Mechanism and Machine Theory*, *90*, 219-229.

26. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. & Metaxas, D. N. (2017). Stackgan: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 5907-5915).

27. Zhao, L., Bai, H., Liang, J., Zeng, B., Wang, A. & Zhao, Y. (2019). Simultaneous Color-Depth Super-Resolution with Conditional Generative Adversarial Networks, *Pattern Recognition*, *88*, 356-369.