Comparative Study of Data Augmentation Strategies for White Blood Cells Classification

George Kolokolnikov, Andrey Samorodov
Bauman Moscow State Technical University
Moscow, Russia
geokolok5@gmail.com, avs@bmstu.ru

***Abstract*—The paper is devoted to the study of strategies that can be applied to unbalanced data in solving the task of classifying white blood cells. The main goal of the proposed paper is to determine the best approach to combat data imbalance when working with images of blood cells. Description of classical and state-of-the-art methods to deal with imbalance is given. In the course of the research, biomedical data is collected, annotated, and preprocessed, as well as selected strategies are applied to form datasets. Base model of artificial neural network for image classification is selected and built. Also, the dependence of trained models accuracy on the applied strategies is studied. Thus, the best approach to data augmentation for white blood cells classification problem is determined.**

I. Introduction

Deep learning approaches are gaining more popularity in computer vision tasks related to medical problems [1]. The most common application of neural network models in medicine is the analysis of biomedical images in fields like oncopathology [2], hematology, ultrasound, cardiology [3], etc. To solve the widely spread problems of object classification and segmentation, it is necessary to utilize supervised learning with sufficient amount of labeled data. An often-encountered problem is an unbalanced set of training data, in which the number of elements in different classes is not the same. For instance, pathological cases are less common than normal ones. A model trained on an unbalanced dataset, in which examples of the norm are more common, will often classify pathological cases as normal ones, that is, the model will be subject to the type II error. Type II errors are a significant problem in medical research. They give a patient and a doctor a false belief that disease is absent, while in reality it is present. Therefore, an important task is the selection of strategies to reduce the effect of an unbalanced training dataset on the accuracy of trained model. This is the main goal of the proposed paper: to find the best data augmentation approach to build a classification model for white blood cells images. The research takes into account the real conditions of model development, under which computing resources and number of blood microsamples are limited. Lack of labeled data is another common problem [4] that the augmentation method can address.

II. Goal setting and research pipeline

A. Research pipeline

The research is aimed to find the optimal data augmentation method for developing white blood cells classifier. It is assumed that the classifier receives segmented images [5] as input, in which exactly one classifiable object (one of the types of white blood cells, a thrombocyte or an artifact) is surrounded by red blood cells. The whole process of digital analysis of a cytological blood sample [6] is presented in Fig. 1.

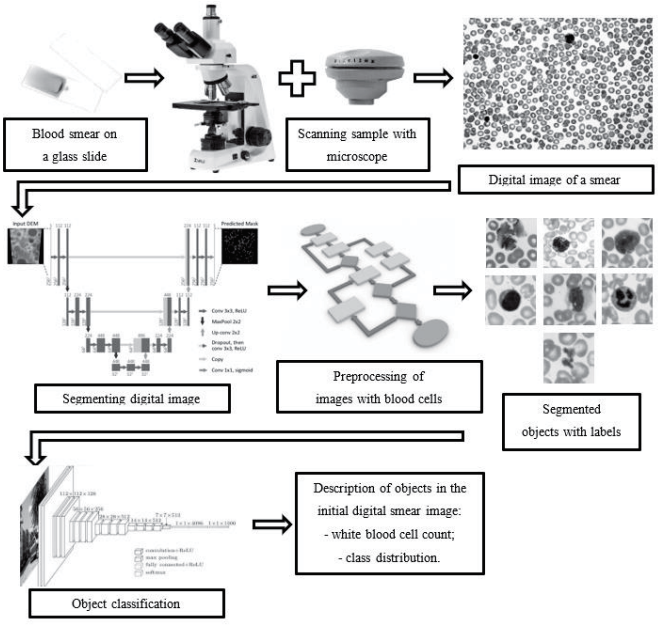


Fig. 1. Stages of digital analysis of cytological blood microsamples

This paper covers the stage of data preprocessing step. Evaluation of preprocessing quality is carried out by validating a neural network model trained on an appropriately prepared training dataset.

The research pipeline shown in Fig. 2 reflects the main stages of work from goal setting to obtaining the best data augmentation strategy. Right after goal setting data acquisition step takes place. The next step is represented by dataset preparation according to selected strategies. Two of the last

data augmentation approaches require training generative models, which is the next step of the research. After that, the selection of the standard convolutional neural network for image classification is carried out. Finally, model performance dependence on the selected strategy is studied and discussed.

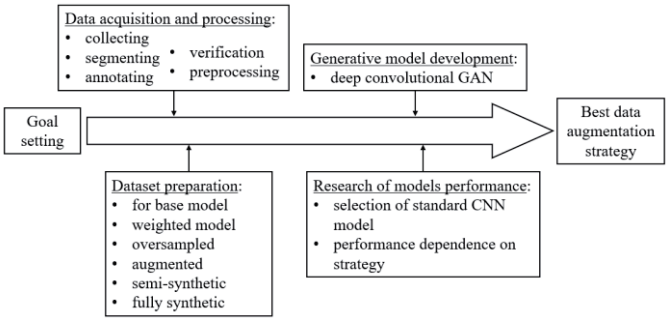


Fig. 2. Research pipeline

The following software technological stack was used:

- Python 3.7 as main programming language.
- Jupyter notebook for coding.
- OpenCV for image management.
- Scikit-learn for dataset preparation.
- Albumenations for data augmentation.
- Keras with TensorFlow as backend for building and training deep learning models.
- And some other auxiliary packages (Numpy, Matplotlib, etc.).

Training of deep learning models is performed both with local computing power (including NVIDIA GeForce GTX 690 and GTX 1050) and cloud service – Google Colaboratory (Tesla K80 GPU).

The blood cells images are registered with automated microscopy system "Granat" [6] by scanning blood smears microsamples with Pixelink digital camera.

B. Imbalanced data problem

The problem of unbalanced data is one of the main issues in machine learning (ML) and data mining, because majority of ML algorithms assume that the data is distributed evenly. Data imbalance is the predominance of certain class instances over the others [7]. Training on such dataset may lead to biased classification results towards the dominant class. Accuracy of trained model will be limited by its ability to predict rare cases, for example, the presence of an oncological disease in a patient. At the same time, the model will classify all observations as instances of the dominant class.

Most real datasets for solving the classification problem do not have exactly the same number of instances in each class: there is always some difference, which often does not matter. However, if the ratio of the number of objects is more than 4:1, problems typical of unbalanced data are possible.

Many image recognition tasks require detection of rare phenomena: illegal penetration into the network, fraud or diseases. In this case, the machine learning model can be subject to either false positives or false negatives [8]. This means that the patient may suffer from a rare disease, but the machine learning model cannot predict this, since most of the data will be obtained from patients without the disease. Obviously, such a mistake much more "costly" (in terms of consequences) than the case when a healthy patient is falsely diagnosed with a disease.

C. Strategies of solving imbalance data problem

To solve the problem of unbalanced data, a number of solutions are proposed (Fig. 3), which can be divided into two groups according to the principles underlying them [7]:

- Methods that equalize the number of instances in different classes.
- Methods that take into account the peculiarity of data imbalance during training.

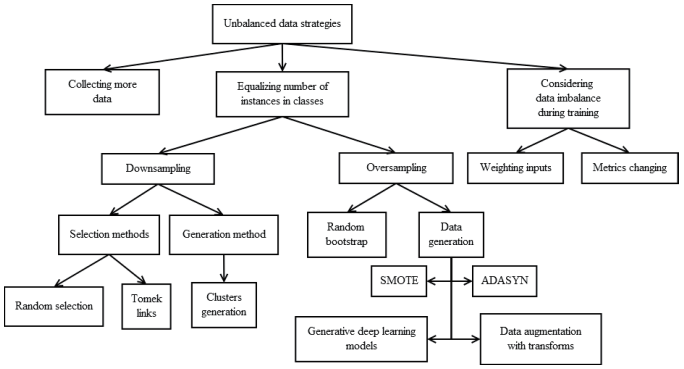


Fig. 3. Strategies of solving imbalance data problem

One of the proposed recommendations is collecting additional data. This method should be used in the first place, since it allows to increase the selection of real non-generated objects. However, if additional data collection is not possible, other strategies should be used.

Methods of the first group are more common. They include:

- Downsampling.
- Oversampling.

In turn, these methods are divided into methods based on arbitrary selection or bootstrap, and methods based on generation of a new dataset.

Methods of the second group are less common, they allow to take into account the imbalance of data during training [8]. In case of rebalancing (or model penalty method), the data is weighted so that the class with the largest number of instances is assigned less weight. In case of metrics changing, special metrics (recall, f-score, precision) are used during training, reflecting the quality of the model.

There are also hybrid methods [9] that combine data-level and algorithm-level approaches. The main point of these methods is to utilize sampling and cost-sensitive learning at the same time in the form of ensemble models.

More detailed description of selected strategies is reported in section IV.

III. Data acquisition and preprocessing

Registration of blood smear images is carried out with the "Granat" complex that consists of the Meiji optical microscope and the Pixelink digital camera. Images were pre-processed so that only one white blood cell, or thrombocyte, or artifact surrounded by red blood cells is present in the frame. Images are manually annotated using a hematological atlas [10] with the following classes: artifact, basophil, eosinophil, lymphocyte, monocyte, neutrophil, thrombocyte (Fig. 4). Images sorted by classes are saved into the appropriate directories. Then, using a developed preprocessing program, the images are read and randomly shuffled. Sub-sampling is performed for each image up to the resolution of 224x224 pixels. This operation allows faster training of neural networks, since the dimension of data significantly affects the required computing power. Also, the specified image size is used to coordinate the dimension of data with the input dimensions of pre-trained neural networks. The data is normalized and the resulting dataset with class labels is saved in a numerical format as a numpy array data structure.

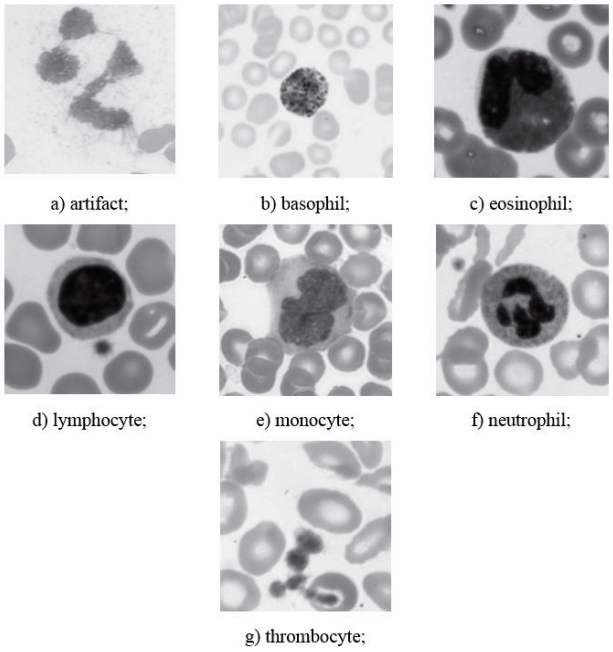


Fig. 4. Classes of blood cells images

IV. Dataset preparation

The first step of dataset preparation is splitting data into train, test, and validation sets. Test and validation set sizes are chosen as 10% and 10% of the initial dataset respectively. The validation set is subject to augmentation. After augmentation, validation set is used only for model performance evaluation. The other 80% of initial dataset forms the training set.

Before applying strategies of balancing dataset, it is necessary to select the appropriate ones taking into account the particularities of the given dataset. The given data is unstructured, that is, images are input to a classifier, and not a set of extracted features. Therefore, such methods of oversampling as SMOTE, ADASYN, and such methods of decreasing selection as Tomek links and generation based on the creation of centroids are not suitable in this case either (due to their purpose for working with structured data). Also, considering the initial distribution of data (Fig. 5), the downsampling approach is not applicable as it can lead to training with extra small dataset. Thus, to solve the problem, the following methods are proposed:

- Random oversampling.
- Weighting input data.
- Image augmentation with transforms.
- Generating synthetic data with deep learning models.

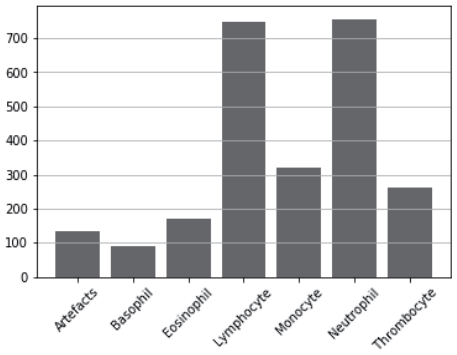


Fig. 5. Initial train set distribution

A. Dataset for base model

The base model is considered as an algorithm trained on the initial dataset without changes. The initial train set is an array of images with a dimension of 224x224x3 with a total number of elements of 2484. The distribution of objects in classes is shown in Fig. 5. The dominant class is a group of neutrophils, containing about 757 instances. The second most common class is lymphocytes with 749 objects. Basophils are a rare class; the number of samples is 90 objects. The remaining classes have from 134 to 321 elements. The number of objects of the dominant class exceeds the size of the rare class by approximately 8.4 times.

B. Dataset for weighted model

Dataset for weighted model is the same as for base model, but the main difference is in weight assignment for input data. An oversampling effect can be achieved by weighting data. Many classification algorithms take an argument, which allows to perform an increase or decrease in the weight of the data. As a result, the error is discounted for records with low weights in favor of records with higher weights [8].

To calculate class weights appropriate function from Scikit-learn package is used. The resulting weight vector is saved for further use in model training.

C. Oversampled dataset

An oversampling method is based on extracting elements from the original set and adding them back. Random oversampling involves supplementing the original data set with copies of objects of rare classes. This is one of the earliest proposed methods, which also proved its reliability [11]. Instead of duplicating each element in a rare class, some of them can be randomly selected with a replacement from the

initial set. Random oversampling is applied to the training set to equalize class distribution. As a result, the size of each class is 757 objects.

D. Augmented dataset

Machine learning algorithms, especially convolutional neural networks, find the most obvious features of objects that distinguish one class from another. For example, if in the training set of images the objects of the first class are oriented from right to left, and the objects of the second class from left to right, then the classifier will divide the images of objects primarily in spatial orientation, which is not really a defining feature of classes. To eliminate this effect, augmentation of the dataset is used. It is performed as simple image transformations. Also, this technique can be applied to objects of a rare class to balance the class distribution. This method of generating objects is mainly applied to images (image data augmentation).

Albumentations library [12] is used in this research to effectively perform data augmentation. The following set of transformations (Fig. 6) is randomly applied to images: horizontal flip; shift, scale, and rotate; random rotate by 90 degrees; grid distortion; blur.

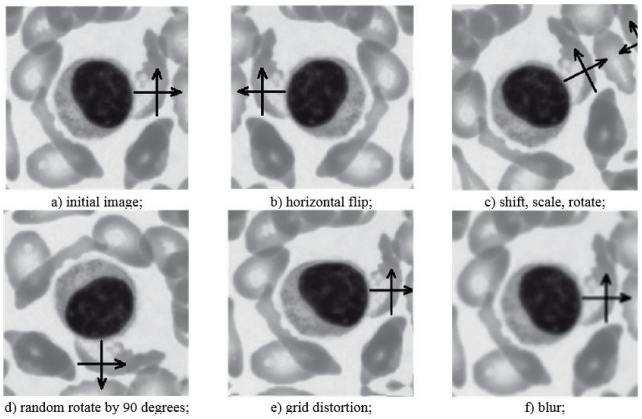


Fig. 6. Image transformations (arrows overlaid for better visualization)

The augmentation is applied to instances of all classes (except “Lymphocyte” and “Neutrophil”) N times, where N is equal to the ratio of number of “Neutrophil” class objects to the size of current class. The resulting train set distribution is shown in Fig. 7.

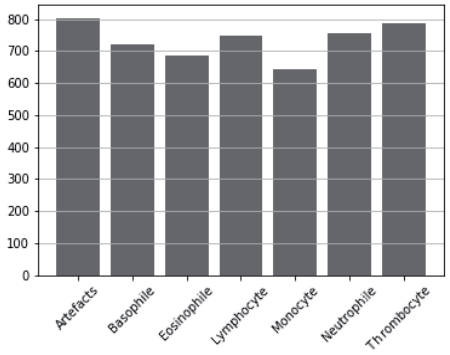


Fig. 7. Augmented train set distribution

E. Synthetic dataset

In the real world, data can exist in various forms and conditions that cannot be accounted for or modeled by the listed above simple methods. To create new objects of an unstructured data type, for example, images, it is proposed to use generative models (Fig. 8). There are three main types of generative models for working with images: generative adversarial networks (GAN) [13], variational autoencoders (VAE) [14], and pixel recurrent neural networks (PixelRNN) [15].

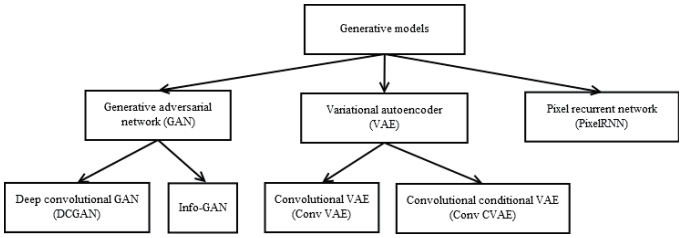


Fig. 8. Generative models for image generation

The learning process of GAN models is based on competition between two separate networks [13]: a generator network and a discriminator network. The generator creates synthetic images, and the discriminator tries to determine whether input objects are samples from the real distribution or synthetically generated distribution. Each time the discriminator notices the difference between the two distributions, the generator adjusts its parameters to make the difference less noticeable. As a result, the generator will learn to accurately reproduce the real distribution of data, and the discriminator, by checking random samples from two distributions, will not be able to distinguish them.

Deep convolutional generative adversarial neural network is a type of GAN with convolutional layers instead of fully connected. The DCGAN architecture is more suitable to generate images than simple GAN.

InfoGAN [16] is one more architecture of GAN that allows to generate objects of different classes with single trained model. DCGAN trained on images of different classes can generate images randomly: the process of generating specific instances is uncontrolled. InfoGAN can accept the label a of class, at the generation step, and synthesize the required images.

Autoencoder is a type of neural network that reconstructs input signal at the output layers with data compression in the middle layers (bottleneck). Variational autoencoder [14] is a type of autoencoders that learns to map input data into a latent space and to generate instances from it. As well as GAN, simple VAE can generate only random samples, however Conditional VAE (as InfoGAN) can create images of a required class.

One more option of generative models is the autoregressive model, such as PixelRNN, which learns to simulate the conditional distribution of each individual pixel, in accordance with previous neighbor pixels.

Due to computing power restrictions and insufficient dataset size (less than 1000 samples in each class), the simplest version of generative model is implemented. Seven DCGAN models are trained separately for each blood cells class.

The architecture of the DCGAN is shown in Fig 9. It consists of a generator neural network and a discriminator neural network.

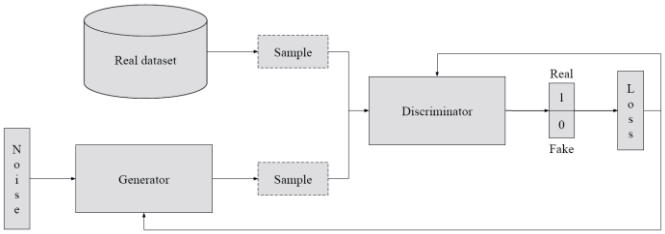


Fig. 9. Deep convolutional GAN architecture

The generator network (Fig. 10) accepts a vector of random values taken from gaussian distribution and passes it through a fully connected layer and a set of transpose convolutional layers. The number of layers and filters in each layer is selected to get 256x256x3 image as output.

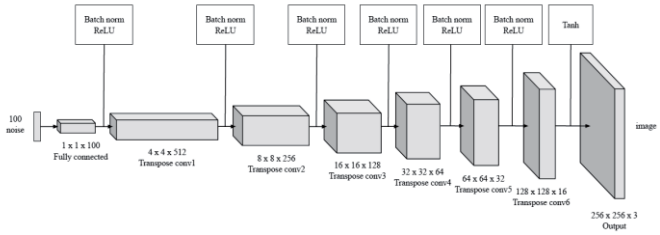


Fig. 10. Architecture of generator neural network

The discriminator network (Fig. 11) works as a standard convolutional neural network for classification: it accepts an image, passes it through a set of convolutional layers, and after fully connected layer with sigmoid activation function decides whether the image is real or synthetic.

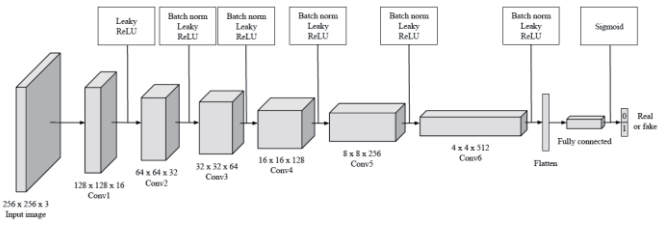


Fig. 11. Architecture of discriminator neural network

Each DCGAN is trained for 5000 epochs. Fig. 12 shows the improvement of generated images with increase of the number of iterations. The trained DCGAN models for each class are used to generate:

- Semi-synthetic dataset.
- Fully synthetic dataset.

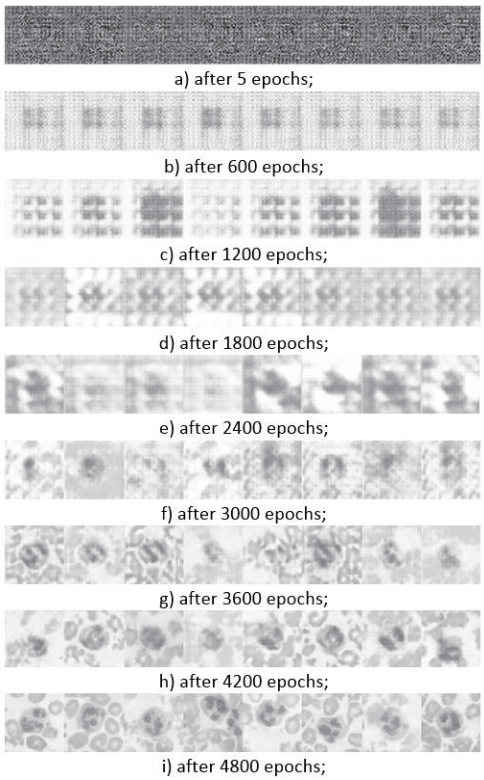


Fig. 12. Improvement of generated images with increase of epoch number

In case of the semi-synthetic dataset, DCGAN is utilized to generate N images, where N is equal to difference between "Neutrophil" class size and current class size. Generated images set is appended to initial images set. Thus, in total each class contains 757 instances (Fig. 13).

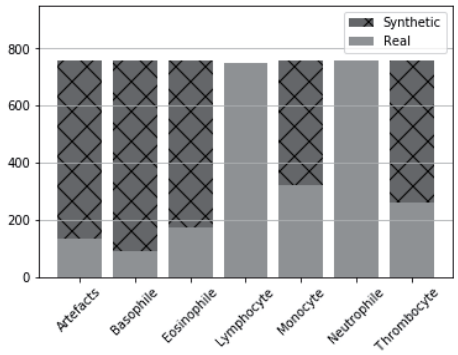


Fig. 13. Semi-synthetic train set distribution

In case of the fully synthetic dataset, all instances of each class are generated. The purpose of this dataset is to evaluate ability of classifier to be trained only on synthetic data but work with real data.

V. Study of Classification Models Performance

A. Classification model selection

The use of convolutional neural network architecture has been considered in this study because the problem relates to image classification field. To evaluate the effectiveness of data

augmentation strategies, it is necessary to choose a reliable classification model with an ability of fast prototyping. Therefore, VGG16 [17] has been chosen as the architecture of the convolutional neural network. The model (Fig. 14) consists of a sequence of convolutional layers, ReLU activation functions, and max pooling layers. The features extracted by convolutional layers are fed to the classifier, presented by fully connected layers with ReLU activation function. In the last layer there are 7 neurons as the number of classes and softmax activation function.

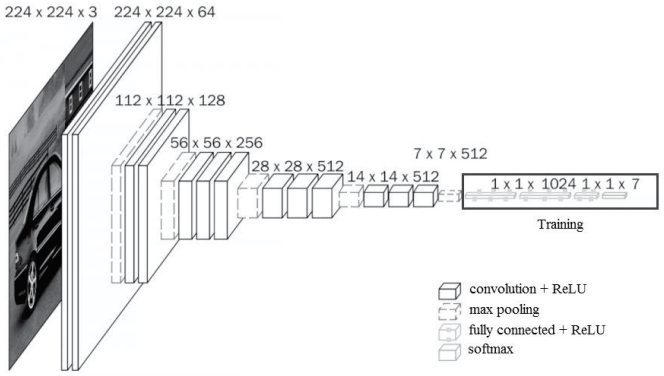


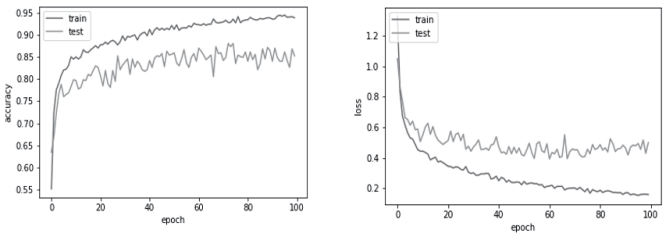
Fig. 14. VGG16 architecture for blood cells classification [17]

To train the model faster, the transfer learning technique is used: weights of the VGG16 model are initialized with values of already trained network. In the research, weights of the convolutional layers are transferred. Only the weights of classification fully connected layers are updated during the training process. Categorical cross-entropy is chosen as a loss function, and the optimization problem is solved with Adam algorithm, based on an adaptive moment estimation. The network is trained for 100 epochs.

B. Model performance research

The results of the training obtained with the selected convolutional neural network model on specifically prepared datasets are represented further in this section. Learning curves as well as precision, recall, and F1-score are shown for each model.

The learning curves of the base model trained on the initial dataset are shown in Fig. 15 and accuracy characteristics are represented in the Table I. The average accuracy of the classification according to the results of validation is 81%. The lowest classification rate is typical for “Monocyte” and “Basophil” classes. Also, the model is subject to overfitting: the difference of accuracy on train and test sets is more than 10%.



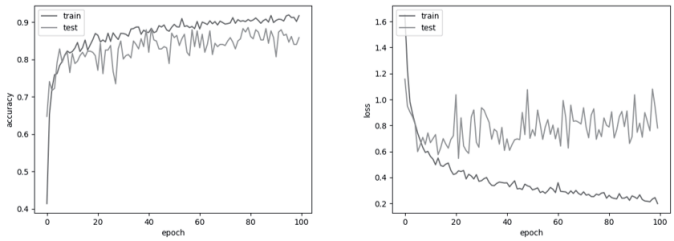
a) accuracy curve; b) loss curve;

Fig. 15. Learning curves of base model training

TABLE I. BASE MODEL ACCURACY CHARACTERISTICS

Class	Precision	Recall	F1-score
Artefact	0.78	0.87	0.82
Basophil	0.74	0.68	0.71
Eosinophil	0.74	0.73	0.74
Lymphocyte	0.86	0.87	0.87
Monocyte	0.75	0.62	0.68
Neutrophil	0.91	0.96	0.94
Thrombocyte	0.88	0.96	0.92

Results of training for the weighted model are shown in Fig. 16 and Table II. The average accuracy of the model is 83%. Weighting the input data according to class distribution allows to decrease overfitting and slightly increase classification rate for “Basophil” and “Monocyte” classes.



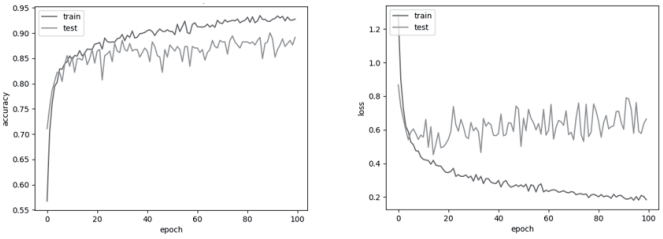
a) accuracy curve; b) loss curve;

Fig. 16. Learning curves of weighted model training

TABLE II. WEIGHTED MODEL ACCURACY CHARACTERISTICS

Class	Precision	Recall	F1-score
Artefact	0.78	0.82	0.80
Basophil	0.80	0.65	0.72
Eosinophil	0.78	0.74	0.76
Lymphocyte	0.91	0.76	0.82
Monocyte	0.63	0.84	0.72
Neutrophil	0.92	0.95	0.94
Thrombocyte	0.88	0.97	0.93

Results for the model trained on the oversampled dataset are shown in Fig. 17 and Table III. The average accuracy of the model is 85%. Oversampling leads to increase of model accuracy characteristics, even though overfitting is stronger in this case than in previous one.



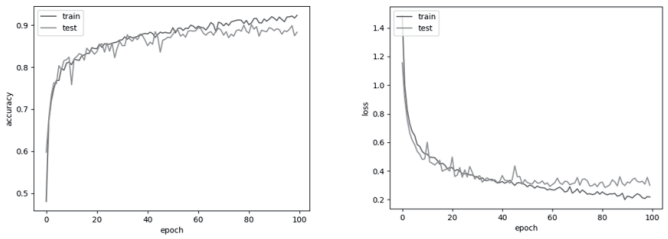
a) accuracy curve; b) loss curve;

Fig. 17. Learning curves of model trained on oversampled dataset

TABLE III. OVERSAMPLED MODEL ACCURACY CHARACTERISTICS

Class	Precision	Recall	F1-score
Artefact	0.85	0.83	0.84
Basophil	0.72	0.84	0.77
Eosinophil	0.85	0.77	0.81
Lymphocyte	0.97	0.77	0.86
Monocyte	0.74	0.86	0.80
Neutrophil	0.95	0.96	0.96
Thrombocyte	0.96	0.94	0.95

Results for the model trained on the augmented dataset are shown in Fig. 18 and Table IV. The average accuracy of the model is 89%. Augmentation with image transformations significantly improves the accuracy characteristics of the model for each class. There is almost no overfitting during training process.



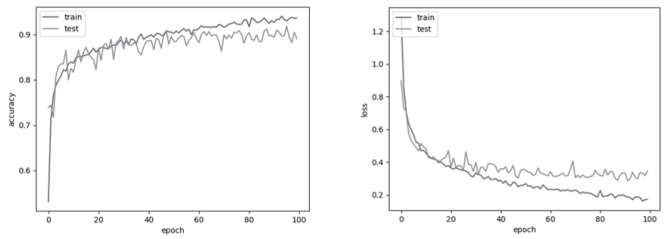
a) accuracy curve; b) loss curve;

Fig. 18. Learning curves of model trained on augmented dataset

TABLE IV. AUGMENTED MODEL ACCURACY CHARACTERISTICS

Class	Precision	Recall	F1-score
Artefact	0.88	0.98	0.92
Basophil	0.78	0.90	0.83
Eosinophil	0.76	0.90	0.83
Lymphocyte	0.86	0.91	0.89
Monocyte	0.89	0.76	0.82
Neutrophil	0.95	0.96	0.96
Thrombocyte	0.96	0.94	0.95

Results for the model trained on the semi-synthetic dataset are shown in Fig. 19 and Table V. The average accuracy of this model is a bit less than the value of previous model: 88%. Classification rate for “Basophil” class is the highest among other models. Also, overfitting is practically absent in this model.



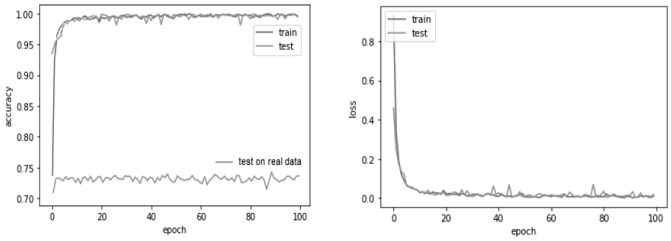
a) accuracy curve; b) loss curve;

Fig. 19. Learning curves of model trained on semi-synthetic dataset

TABLE V. SEMI-SYNTHETIC MODEL ACCURACY CHARACTERISTICS

Class	Precision	Recall	F1-score
Artefact	0.94	0.83	0.88
Basophil	0.97	0.95	0.96
Eosinophil	0.77	0.99	0.87
Lymphocyte	0.91	0.93	0.92
Monocyte	0.89	0.93	0.88
Neutrophil	0.99	0.71	0.83
Thrombocyte	0.83	0.97	0.90

Results for the model trained only on the synthetic dataset are shown in Fig. 20 and Table VI. The average accuracy on the validation set is 74%, although evaluation on test set formed with synthetic images shows almost 99%. The overfitting problem in this case is caused by different data distributions of train and validation set.



a) accuracy curve; b) loss curve;

Fig. 20. Learning curves of model trained on fully synthetic dataset

TABLE VI. FULLY SYNTHETIC MODEL ACCURACY CHARACTERISTICS

Class	Precision	Recall	F1-score
Artefact	0.74	0.84	0.78
Basophil	0.64	0.76	0.69
Eosinophil	0.62	0.76	0.69
Lymphocyte	0.71	0.78	0.75
Monocyte	0.76	0.66	0.68
Neutrophil	0.81	0.83	0.82
Thrombocyte	0.82	0.80	0.81

VI. RESULTS DISCUSSION

The results of the performance analysis showed that two classifiers trained on augmented dataset and on semi-synthetic dataset have the highest average F1-score (Table VII). The accuracy characteristic of both models is 89%. It should be noted that:

- Random oversampling increases accuracy.
- Weighting input data decreases overfitting.
- Classification model can be trained on synthetic data only, although accuracy of the model will not be sufficient for real-world applications.

TABLE VII. MODEL COMPARISON

Model	Average precision	Average recall	Average F1-score
Base	0.80	0.81	0.81
Weighted	0.81	0.82	0.81
Oversampled	0.86	0.85	0.85
Augmented	0.87	0.91	0.89
Semi-synthetic	0.90	0.90	0.89
Fully synthetic	0.73	0.78	0.74

Generating semi-synthetic dataset with simple DCGAN has proven its effectiveness in data balancing of white blood cells. However, augmentation with image transformations is much simpler and does not require training a separate model.

Table VIII demonstrates confusion matrix for model trained on semi-synthetic dataset. The main trends shown in Table VIII are typical for all trained classifiers. From the analysis of the confusion matrix it follows that the main classification errors fall on:

- Classification of artefacts as thrombocytes.
- Classification of lymphocytes as monocytes and vice versa.
- Classification of neutrophils as eosinophils.

TABLE VIII. CONFUSION MATRIX FOR SEMI-SYNTHETIC MODEL

	Art	**Bas**	**Eos**	**Lym**	**Mon**	**Neu**	**Thr**
Artefact	115	0	0	1	3	0	19
Basophil	0	112	0	0	6	0	0
Eosinophil	0	0	113	0	0	1	0
Lymphocyte	0	3	3	115	4	0	0
Monocyte	0	0	4	11	108	0	0
Neutrophil	4	0	26	0	4	86	1
Thrombocyte	3	0	0	0	0	0	99

These observations are explained by several reasons:

- Errors of annotation (assignment of cell images to an erroneous group).
- According to medical sources [10], one of the most difficult problems in practice is to distinguish between lymphocytes and monocytes, therefore it can be assumed that the classifier has reached the base classification error.

VII. CONCLUSION

In the course of the research, a comparative study of data augmentation techniques for white blood cells classification is carried out. A number of datasets is created in accordance with selected unbalanced data strategies. Six VGG16 convolutional neural networks are trained on the prepared datasets. According to the model evaluation results, augmentation with image transformations and generating images with DCGAN are the two methods that provide the highest accuracy characteristics of classification model. Considering the difficulties of building additional DCGAN models for data balancing, augmentation with Albumentations library is the first choice for developing the digital analysis system of cytological blood microsamples. It is expected to apply the latter method for data augmentation in building semantic segmentation system [18] of blood cells images.

ACKNOWLEDGMENT

Authors would like to thank Research and Educational Medical-Technological Center of Bauman Moscow State Technical University and hematologist Irina Anatolievna Perkovskaya for blood microsamples and laboratory equipment.

REFERENCES

[1] P. Lakhani, “Hello world deep learning in medical imaging”, *Journal of digital imaging*, vol.31, 2018, pp. 283-289.

[2] D.A. Dobrolyubova, T.A. Kravtsova, O.A. Samorodova, A.V. Samorodov, E.N. Slavnova, N.N. Volchenko, “Automatic image analysis algorithm for quantitative assessment of breast cancer estrogen receptor status in immunocytochemistry”, *Pattern Recognition and Image Analysis*, vol. 26(3), 2016, pp. 552-557.

[3] N.S. Konnova, M.A. Basarab, D.A. Basarab, “Image processing using artificial intelligence methods in cardiovascular decision support systems”, *in Proc. SPIE 10836*, *International Conference on Image and Video Processing, and Artificial Intelligence*, vol. 108361, Oct. 2018.

[4] Y. Fedorenko, V. Chernenkiy, Y. Gapanyuk, (2019, July). “The Neural Network for Online Learning Task Without Manual Feature Extraction”, *in International Symposium on Neural Networks*, *Springer*, *Cham*, Jul. 2019, pp. 67-76.

[5] O.A. Samorodova, A.V. Samorodov, “Fast implementation of the Niblack binarization algorithm for microscope image segmentation”, *Pattern recognition and image analysis*, vol. 3, 2016, pp. 548-551.

[6] A.V. Samorodov, “Biotechnological Systems for Automated Microscopy of Cytology Specimens”, *Biomedical Engineering*, vol. 52(6), 2019, pp. 387-390.

[7] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, “Handling imbalanced datasets: A review”, *GESTS International Transactions on Computer Science and Engineering*, vol. 30(1), 2006, pp. 25-36.

[8] P. Bruce and A. Bruce, *Practical statistics for data scientists: 50 essential concepts*. O'Reilly Media, 2017.

[9] B. Krawczyk, “Learning from imbalanced data: open challenges and future directions”, *Progress in Artificial Intelligence*, vol. 5(4), 2016, pp. 221-232.

[10] H. Diem, H. Torsten, H. Theml. *Color Atlas of Hematology: Practical Microscopic and Clinical Diagnosis*. Thieme, 2004.

[11] C. X. Ling, C. Li, “Data mining for direct marketing: Problems and solutions”, *In Kdd*, vol. 98, Aug. 1998, pp. 73-79.

[12] Albumentations Docs website, Python library documentation, Web: https://albumentations.readthedocs.io/en/latest/index.html.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, Y. Bengio, “Generative adversarial nets”, *Advances in neural information processing systems*, 2014, pp. 2672-2680.

[14] C. Doersch, “Tutorial on variational autoencoders”, *arXiv preprint*, arXiv:1606.05908, 2016.

[15] A. Oord, N. Kalchbrenner, K. Kavukcuoglu, “Pixel recurrent neural networks”, *arXiv preprint*, arXiv:1601.06759, 2016.

[16] M. Mirza, S. Osindero, “Conditional generative adversarial nets”, *arXiv preprint*, arXiv:1411.1784, 2014.

[17] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint*, arXiv:1409.1556, 2014.

[18] D. Parpulov, A. Samorodov, D. Makhov, E. Slavnova, N. Volchenko, V. Iglovikov, “Convolutional neural network application for cells segmentation in immunocytochemical study”. *In 2018 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT) IEEE*, May 2018, pp. 87-90.