

## APPROACH TO THE AUTOMATIC CREATION OF AN ANNOTATED DATASET FOR THE DETECTION, LOCALIZATION AND CLASSIFICATION OF BLOOD CELLS IN AN IMAGE

**Kovalenko S. M.** – PhD, Associate Professor, Associate Professor of the Department of Software Engineering and Management Intelligent Technologies of the National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine.

**Kutsenko O. S.** – Dr. Sc., Professor, Professor of the Department of System Analysis and Information-Analytical Technologies of the National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine.

**Kovalenko S. V.** – PhD, Associate Professor, Professor of the Department of System Analysis and Information-Analytical Technologies of the National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine.

**Kovalenko A. S.** – Postgraduate student of the Department of System Analysis and Information-Analytical Technologies of the National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine.

### ABSTRACT

**Context.** The paper considers the problem of automating the creation of an annotated dataset for further use in a system for detecting, localizing and classifying blood cells in an image using deep learning. The subject of the research is the processes of digital image processing for object detection and localization.

**Objective.** The aim of this study is to create a pipeline of digital image processing methods that can automatically generate an annotated set of blood smear images. This set will then be used to train and validate deep learning models, significantly reducing the time required by machine learning specialists.

**Method.** The proposed approach for object detection and localization is based on digital image processing methods such as filtering, thresholding, binarization, contour detection, and filling. The pipeline for detection and localization includes the following steps: The given fragment of text describes a process that involves noise reduction, conversion to the HSV color model, defining a mask for white blood cells and platelets, detecting the contours of white blood cells and platelets, determining the coordinates of the upper left and lower right corners of white blood cells and platelets, calculating the area of the region inside the bounding box, saving the obtained data, and determining the most common color in the image; filling the contours of leukocytes and platelets with said color; defining a mask for red blood cells; defining the contours of red blood cells; determining the coordinates of the upper left and lower right corners of red blood cells; calculating the area of the region within the bounding box; entering data about the found objects into the dataframe; saving to a .csv file for future use. With an unlabeled image dataset and a generated .csv file using image processing libraries, any researcher should be able to recreate a labeled dataset.

**Results.** The developed approach was implemented in software for creating an annotated dataset of blood smear images

**Conclusions.** The study proposes and justifies an approach to automatically create a set of annotated data. The pipeline is tested on a set of unlabelled data and a set of labelled data is obtained, consisting of cell images and a .csv file with the attributes “file name”, “type”, “xmin”, “ymin”, “xmax”, “ymax”, “area”, which are the coordinates of the bounding box for each object. The number of correctly, incorrectly, and unrecognised objects is calculated manually, and metrics are calculated to assess the accuracy and quality of object detection and localisation.

**KEYWORDS:** computer vision, object detection, object localization, digital image processing, classification.

### ABBREVIATIONS

CNN is a convolutional neural network;  
CSV are comma separated values;  
CV is a computer vision;  
FN is a false negative;  
FP is a false positive;  
HSV are hue, saturation, value  
RBC is a red blood cell;  
RGB is a red, green, blue;  
RNN is a recurrent neural networks  
TP is a true positive;  
WBC is a white blood cell.

### NOMENCLATURE

$\delta(a,b)$  is a Kronecker delta;  
 $\sigma$  is a standard deviation of the Gaussian distribution;  
*Acc* is accuracy of cell recognition;  
*B* is a blue component in the RGB model;  
*C* is a most common contour in the image;

*Cnt* is a contour, found in the image;  
 $G_f$  is a Gaussian filter;  
*G* is a green component in the RGB model;  
 $G(u,v)$  is a Gaussian distribution at position  $(u,v)$  ;  
*H* is hue is a component in the HSV model;  
*F1* is a harmonic mean between precision and recall;  
*I* is an image;  
 $I(x,y)$  is a value of a pixel in the image *I* at position  $(x,y)$ ;  
 $M(x,y)$  is a value of the pixel on the mask at position  $(x,y)$ ;  
*Pr* is a precision of cell recognition;  
*R* is a red component in the RGB model  
*Rec* is a recall of recognition;  
*r* is a blur radius;  
*S* is a saturation is a component in the HSV model;  
*T* is a threshold value;  
*V* is a value component in the HSV model;  
 $(x,y)$  are coordinates of the pixel in the image.

## INTRODUCTION

With the development of machine learning models, this branch of artificial intelligence is increasingly penetrating various areas of our lives and is transforming many aspects of the modern world. This technology is reprogramming and modernizing various sectors, from industry and medicine to education and everyday life.

Artificial intelligence can effectively solve the tasks of detecting, localizing, classifying, and counting cells in an image, which is an important task in the field of medicine and biology [1]. It is an important stage in the process of counting the number of blood cells of different types. This procedure, in turn, is an important step in laboratory diagnostics and medical research, which provides valuable data for assessing the health status of patients and detecting various diseases.

Currently, convolutional neural networks and deep learning methods are used to identify and classify blood cells, which are essentially supervised learning methods and require large enough data sets for training and validation. The general pipeline for the machine learning process is shown in Fig. 1.

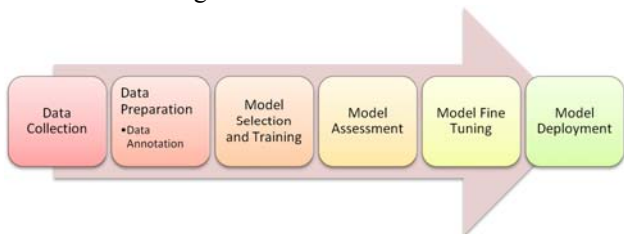


Figure 1 – General scheme of machine learning steps

Sometimes it is believed [2] that machine learning specialists spend the most time training the model, but a study [3] found that data scientists spend approximately 37.75% of their time preparing and cleaning data. In addition to data preparation and cleaning, interpretation of the results also remains important. Data visualization (12.99%) and demonstrating the value of data through reporting and presentation (16.20%) are also important and time-consuming steps to turn data into an actionable tool for answering critical questions. Working with models through model selection, training, and implementation takes approximately 26.44% of respondents' time (Fig. 2).

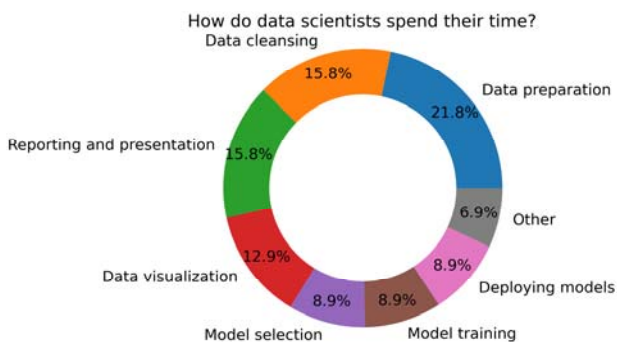


Figure 2 – Time management for machine learning specialists

Thus, the creation of datasets is in itself a rather time- and resource-consuming task. Such datasets are created manually using certain software, such as LabelIMG, VIA, VOTT, and others. Image annotation software has different functionality [4] and can significantly reduce the annotation time compared to simple graphic editors, but still the process of annotating images is a very long process.

Therefore, the automatic creation of annotated datasets for training blood cell classification models will help to significantly reduce the time spent by machine learning specialists.

**The object** of the study is the processes of digital image processing for the detection and localization of blood cells of various types in a blood smear image.

**The subject** of the study is methods and algorithms for digital image processing.

**The purpose of the work** is to increase the overall speed of the deep learning process for detecting, localizing, and classifying blood cells by automatically creating an annotated data set.

## 1 PROBLEM STATEMENT

The study aims to develop and validate an approach for automatically creating an annotated dataset for classifying, identifying, and localizing blood cells in images.

The result of the study is a pipeline for automatic creation of a labeled dataset based on methods of digital image processing and analysis, noise reduction, contour extraction, binarization, filtering, etc. Pipeline should provide the ability to detect and localize cells in blood smear images, taking into account their features and characteristics.

This also includes the development of software for detecting, isolating, analyzing, and localizing blood cells and the creation of a dataset. The output dataset will be defined as a set of unlabeled images and a .csv file containing the following features:

- “file\_name” is the relative name of the file in the folder;
- “xmin”, “ymin” are coordinates of the upper left corner of the bounding box;
- “xmax”, “ymax” are coordinates of the lower right corner of the bounding box;
- “type” is cell type (“WBC” for leukocyte, “RBC” for erythrocyte, and “platelet”);
- “area” is the area inside the bounding box.

With an unlabeled dataset and a generated .csv file using image processing libraries (e.g. PIL and/or opencv-python for Python programming language), any researcher should be able to recreate a labeled dataset.

The study focuses on improving the speed and efficiency of the dataset generation process, ensuring the representativeness of various cells and optimal image quality. The dataset obtained with the developed approach will be used to train and validate a machine learning model that will recognize blood cells in smear images with high accuracy and reliability.

## 2 REVIEW OF THE LITERATURE

Rapid assessment of the body's condition using less invasive laboratory tests has always been relevant. A regular clinical blood test is a fairly accurate and reliable method of assessing the patient's condition and making a diagnosis. It provides important data for assessing the state of health of patients and early detection of possible diseases.

Fig. 3 presents the main areas in which blood cell counting is used and, accordingly, their classification.

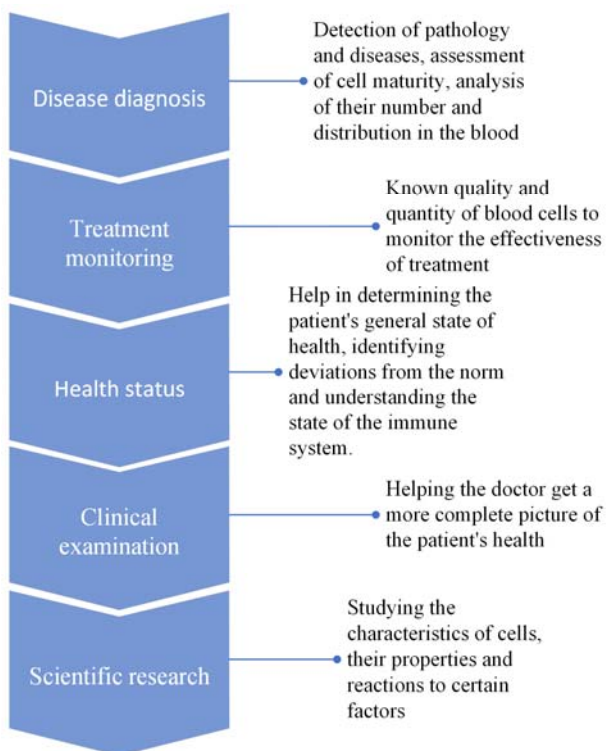


Figure 3 – Fields of application of blood cell classification

Thanks to advances in computer vision and machine learning, the use of automated systems for this task is becoming more common and affordable.

One of the most common tasks is to detect and recognize different types of cells, such as red blood cells (erythrocytes), red blood cells (leukocytes), and platelets.

As mentioned earlier, the classification and counting of blood cells of different types in the image is broad used in diagnosis and monitoring of diseases and treatment effectiveness, medical and scientific research, optimization of processes in laboratories.

There are several approaches to solving this problem. The first of them is the so-called traditional method [5, 6, 7], which combines image processing methods, such as noise reduction, contour extraction, edge detection, segmentation and filtering [8, 9], with machine learning or morphological analysis methods. This approach can be used both to detect different types of blood cells and to detect other biological objects [7]. These methods allow you to significantly speed up the process of object detec-

tion (in comparison with non-automated systems) and provide the ability to use them in real time. However, they depend on the quality of the input image and require pre-processing.

Another approach is based on the use of real-time object detection algorithms, such as YOLO (You Only Look Once) [10, 11] and/or SSD (Single Shot MultiBox Detector) [11]. This method has become quite popular in image classification in recent years because it has high accuracy and the ability to work with different, complex data, such as different resolutions and overlapping object detection. The main feature of this approach is to identify objects in one pass of the image through a neural network. For example, in the case of YOLO, an image is passed through a neural network at once, deriving predicted object boundaries and classes, instead of the traditional approach of first passing the image through an object detection model and then processing the results to determine object boundaries.

However, such methods require a large amount of labeled data for training, which is a separate task. This approach requires large computing power and has a large model size, which can lead to difficulties in storing, implementing and distributing these models. These difficulties can be circumvented by using mechanisms such as transfer of learning [12]. But the problem of having a set of labeled data remains. The size of the dataset can be reduced due to data augmentation [13], but the process of annotation different types of cells still remains time-consuming, especially in the case of determining different types of cells (erythrocytes, leukocytes, platelets).

In [2], an approach to the segmentation of blood cells based on image processing methods is proposed. In this study, it is proposed to apply a similar approach to the automatic creation of a set of annotated data for further application in the automatic detection different types of cells using artificial neural networks.

Machine learning models are usually divided into 3 large groups: unsupervised, supervised, and reinforcement learning models [14] (Fig. 4).

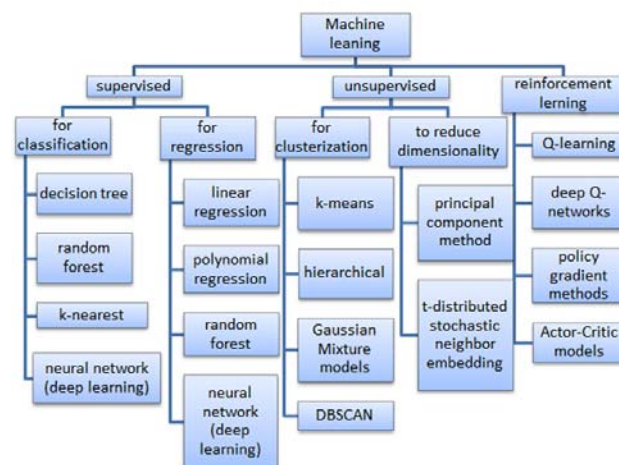


Figure 4 – Classification of machine learning models

One of the fields of artificial intelligence that uses machine learning is computer vision [15]. Computer vision is a branch of artificial intelligence that studies how computers can understand and analyze images and videos. To achieve this goal, various methods and algorithms are used that allow computers to “see” the world around them.

The main tasks of computer vision are [16] (Fig. 5)

1. Image classification: determining the class and type of the object in the image.
2. Object localization: determining the location of the object and selecting it (usually using a rectangular frame).
3. Object detection: determining the location (using frames) and the type and class of objects in the image.
4. Semantic segmentation of the object with selection of the recognized object using a contour and (sometimes) changing certain pixels of the object instead of defining the frame.

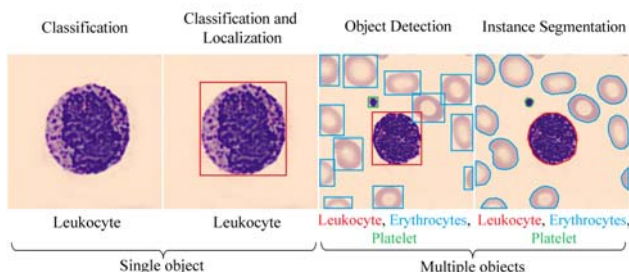


Figure 5 – Classes of machine vision problems

All of these tasks use supervised machine learning techniques to train models based on annotated data and use them in computer vision tasks.

One of the key methods in computer vision is Convolutional Neural Networks (CNN) [15, 17]. These networks are capable to detect patterns and features in images using filters that perform a convolution operation. CNNs are used for object classification, contour detection, face recognition, and many other tasks in computer vision.

Another important method is Recurrent Neural Networks (Recurrent Neural Networks – RNN) [17] that allow you to work with sequences of data, for example, with sequences of video frames. They are useful for analyzing the dynamics of actions or the movement of objects.

In addition, image processing methods are used, such as filtering, segmentation, detection of edges and contours, as well as feature extraction, which help in the recognition of objects in images [2].

Deep learning techniques [17, 18], in particular, are very popular in computer vision. They allow creating complex models that can automatically study the data presented to them, and identify complex patterns and features to solve recognition and classification tasks. Also, geometric modeling methods are used to describe the geometric features of objects in images, which help in solving problems of object recognition and measurement of their characteristics [6, 19].

In general, the combination of these methods and techniques in computer vision allows computers to recognize objects, detect images, classify data, and helps solve many tasks that require the analysis of visual information in real or virtual environments.

That is, different types of neural networks using machine learning methods used in computer vision to recognize objects, detect images, classify data and help solve many tasks that require the analysis of visual information in a real or virtual environment.

### 3 MATERIALS AND METHODS

To automatically create a set of annotated data, the following sequence of actions is proposed (Fig. 6):

1. Loading the image.
2. Noise reduction.
3. Conversion to HVS model.
4. Determining the mask for leukocytes and platelets.
5. Detection contours of leukocytes and platelets
6. Determining the coordinates of the upper left and lower right corners of leukocytes and platelets.
7. Calculation of the area of the region inside the bounding box.
8. Entering data about found objects into a dataframe with columns “file\_name”, “xmin”, “ymin”, “xmax”, “ymax”, “type”, “area”.
9. Determining of the most common color  $C$  in the image.
10. Filling the contours of leukocytes and platelets with color  $C$ .
11. Determining the mask for erythrocytes.
12. Detection the contours of erythrocytes.
13. Detection the coordinates of the upper left and lower right corners of erythrocytes.
14. Calculation of the area of the region inside the bounding box.
15. Entering data about found objects into the dataframe.
16. Saving in .csv file for later use.

To eliminate excess noise that interferes with the recognition of target elements in the image, a Gaussian filter is applied to reduce the sharpness of the image and make it more blurry. This effect is achieved by transforming image pixels. In order to calculate the degree of transformation, this filter uses a normal distribution for two measurements, which can be calculated by formula (1):

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \quad (1)$$

where  $r$  is a blur radius;  $r = \sqrt{x^2 + y^2}$ ;  $\sigma$  is the standard deviation of the Gaussian distribution.

The result of calculations according to this formula is an area of the image that has the form of concentrated circles with a normal distribution from the central point. Pixels with a distribution level that is not equal to 0 are used to create a convolution matrix that is used to analyze the input image. The distribution of the Gaussian value is

as follows: one of the pixels receives the largest weight, while the surrounding pixels, according to their distance from the central one, receive smaller weights. The convolution (core) of the mask is usually much smaller than the entire image. The mask is moved across the image and convolution is applied at each location in the image. The larger the core size, the lower the sensitivity of the detector to noise. The greater the value  $\sigma$ , the smoother the result with less noise. However, with more image smoothing, fewer edges will be detected by the detector.

The algorithm is sequentially implemented for each pixel. A 3×3 or 5×5 cores are often used. An example of a Gaussian filter with a standard deviation  $\sigma = 1.4$ , is calculated using the kernel:

$$G_f = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

The next step is to convert from the RGB color model to the HSV model. The HSV (Hue, Saturation, Value) model is a cylindrical color model used to represent colors in the form of three components: Hue, Saturation, and Value. This model is built on the basis of a transformation from the RGB (Red, Green, Blue) model.

Converting an image from RGB to HSV can simplify color processing and allow various color operations to be performed more efficiently [11]. Converting images from RGB to HSV can make a number of image processing tasks easier and allow us to better understand and work with colors.

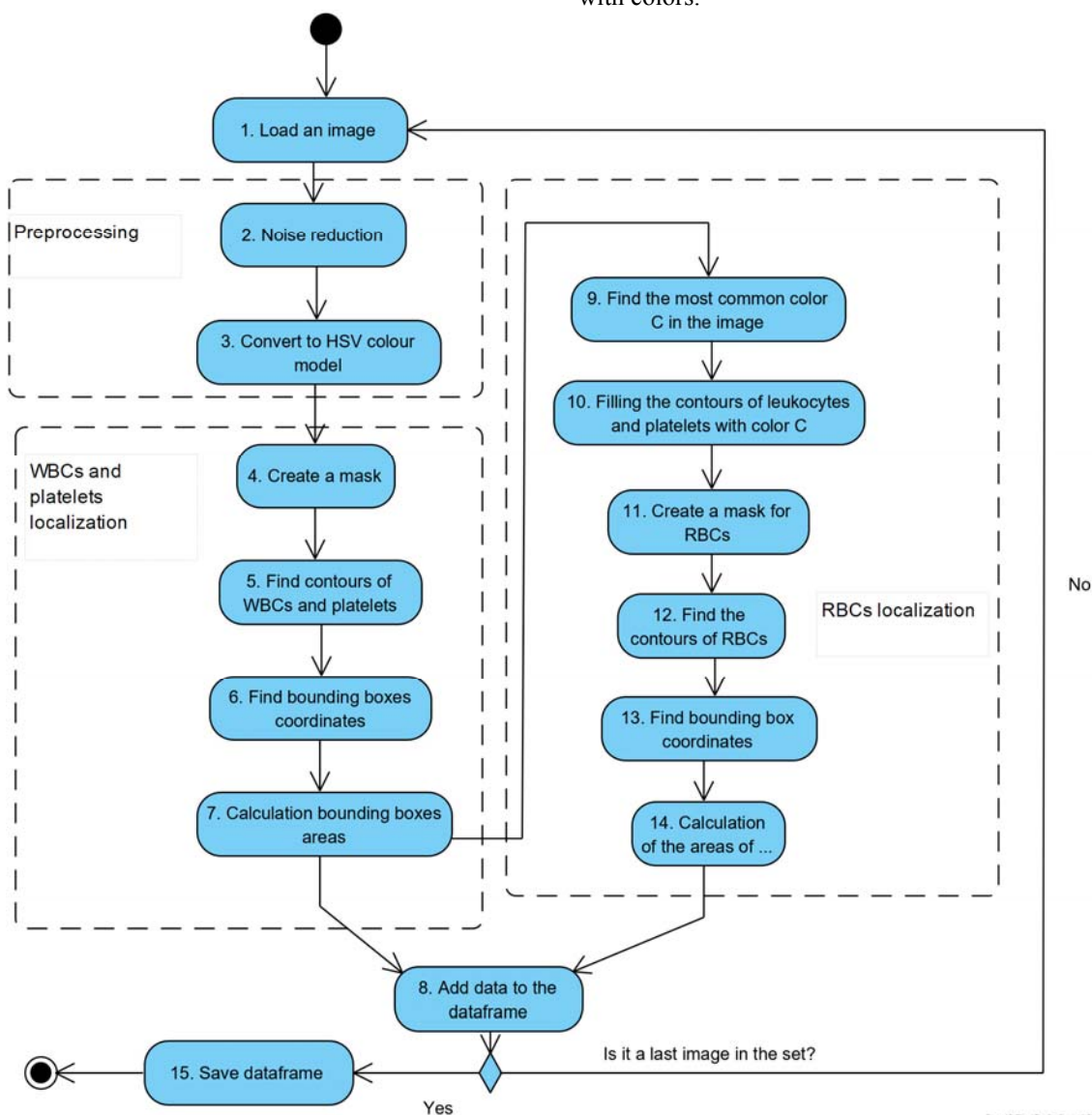


Figure 6 – Activity diagram of automatic creation of an annotated dataset

Converting from the RGB to HSV color model is often used to process images with color information for several reasons (Table 1).

Table 1 – Advantages of image processing in HSV

Advantage	Explanation
Easy color interpretation	In the HSV model, each parameter (hue, saturation, value) represents a specific color characteristic, which simplifies color interpretation. For example, the hue component defines the color tone, saturation defines the brightness of the color, and value defines the brightness of the light source.
Easier color processing	Switching from RGB to HSV can make it easier to process colors and perform a variety of color operations, such as changing hue, increasing or decreasing saturation, adjusting brightness, and more.
Operations with colors	Converting to HSV allows us to perform color-related operations, such as filtering by specific colors, finding color ranges for specific effects, and more.
Color segmentation	Since the HSV model is closer to how humans perceive colors, it can be useful for segmenting objects in an image by color. For example, to distinguish objects of a certain color or group of colors.
Brightness and contrast control	Changing the values in the HSV model allows for better control of image brightness and contrast, which can be useful for enhancing an image or increasing its clarity.

An HSV model can be represented as a cylindrical coordinate system, where hue is located on a circular disk, and saturation and value are displayed on the vertical and radial axes, respectively. This model makes it easier to understand and manipulate individual aspects of color, compared to the RGB model. This is how this model is built:

1. Hue is measured on the color wheel and expressed in degrees, where  $0^\circ$  corresponds to red,  $120^\circ$  to green, and  $240^\circ$  to blue. It is a quantitative value that defines a specific color in the spectrum.

2. Saturation is defined as the distance from the center of the color wheel to its edge. It indicates the “purity” or “vibrancy” of the color. Maximum saturation means a vibrant, rich color, while low saturation can lead to gray-scale.

3. Value measures the brightness of a color. This is the relative brightness of a color from black to white. This is an important aspect because it determines how light or dark the color will be.

To convert from RGB to HSV, the following ratios are used (2): in the case of 8-bit and 16-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range:

$$\begin{aligned}
 V &= \max(R, G, B), \\
 S &= \begin{cases} \frac{V - \min(R, G, B)}{V}, & \text{if } V \neq 0, \\ 0, & \text{otherwise,} \end{cases} \\
 H &= \begin{cases} 60(G - B) / (V - \min(R, G, B)), & \text{if } V = R, \\ 120 + 60(B - R) / (V - \min(R, G, B)), & \text{if } V = G, \\ 240 + 60(R - G) / (V - \min(R, G, B)), & \text{if } V = B, \\ 0, & \text{if } R = G = B. \end{cases} \quad (2)
 \end{aligned}$$

If  $H < 0$  then  $H = H + 360$ . On output  $0 \leq V \leq 1$ ,  $0 \leq S \leq 1$ ,  $0 \leq H \leq 360$ .

The values are then converted to the destination data type (8-bit images):

$$V = 255V, \quad S = 255S, \quad H = H / 2 \quad (\text{to fit to } [0, 255]).$$

Obtaining a mask on an image can be mathematically written using threshold binarization, where the pixel value is compared to a certain threshold and a certain range of values is set to create the mask.

We have an image  $I$ . To get a mask  $M$ , where the white pixels correspond to contour areas, a threshold  $T$  must be determined, which will be used for binarization.

Mathematically, the binarization process can be written as (3):

$$M(x, y) = \begin{cases} 255, & \text{if } I(x, y) > T, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where  $M(x, y)$  is the value of the pixel in the mask at the position  $(x, y)$ ;  $I(x, y)$  is the value of a pixel in the image  $I$  in position  $(x, y)$ ,  $T$  is the threshold for binarization.

Specifying the threshold  $T$  for binarizing the image and create a mask is an important step that determines which pixels are considered part of the contour area. If the threshold value is too low, there may be a lot of “noise” or background detail in the contour mask. If the threshold value is too high, part of the contour may be lost.

Therefore, the choice of threshold depends on the specific image and its properties. There are several approaches to select a threshold:

1. Global threshold using statistical methods, such as the Otsu method, which automatically determines the threshold by maximizing the variance between pixel classes (background and object).

2. Adaptive threshold, where instead of one global threshold, you can use an adaptive threshold for different areas of the image, especially when the illumination in the image is uneven, i.e. use different thresholds for different parts of the image.

3. Experimental selection, when several threshold values are examined and the one that gives the best results for a specific task is chosen.

4. Using data knowledge, where knowledge of image characteristics or regions to be detected is used.

In our study, a combined approach was used to determine the threshold value, namely a combination of an experimental approach and knowledge of the data. It is known [20] that Leishman staining is used to stain blood on smears for microscopy. It is usually used to distinguish and identify leukocytes. It is based on a methanol mixture of “polychrome” methylene blue and eosin, which stain leukocytes and platelets purple, while erythrocytes remain pink. Based on this knowledge and experiment, the threshold value was determined as  $T = 180$ .

To detect and localize RBC, it is proposed to first “fill” the contours of leukocytes and platelets with the color that is most present in the image. The following algorithm is used for this purpose:

Let us denote  $H$ ,  $S$ , and  $V$  as the sets of possible values of hue, saturation, and brightness values, respectively. Based on the features of the opencv-python library [21], which is used for image processing in this study, each of the values of  $H$ ,  $S$  and  $V$  varies in the range  $[0, 255]$ .

Let’s calculate the amount of each color in the HSV space. For this, we will create a matrix  $count\_hsv_{256 \times 256 \times 256}$  to count the number of pixels of each color. To do this, use the formula (4):

$$count\_hsv(H_i, S_j, V_k) = \sum_{x,y \in I} \delta(I_H(x,y) = H_i \& I_S(x,y) = S_j \& I_V(x,y) = V_k), \quad (4)$$

where  $I_H(x,y)$ ,  $I_S(x,y)$  and  $I_V(x,y)$  are values of hue, saturation and brightness of a pixel  $(x,y)$  in the image  $I$ ;  $H_i, S_j, V_k$  are current values of hue, saturation and brightness;  $i, j, k \in [0, 255]$ ;  $\delta(a,b)$  – the Kronecker delta (5):

$$\delta(a,b) = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Finding the most common combination of  $H$ ,  $S$  and  $V$  is represented by formula (6):

$$C = \{H, S, V\} = \arg \max_{(H_i, S_j, V_k) \in H \times S \times V} count\_hsv(H_i, S_j, V_k). \quad (6)$$

The next step is to select the contours. There are several ways to accomplish this task. One of the most common is to use of mathematical morphology operations and find contours based on a mask. The contours can be obtained as the difference between the extended object and the (initial) object itself.

In turn, the dilation operator accepts two pieces of data as input. First, is the image to be expanded. The second is a (usually small) set of coordinate points known as a structural element. It is this structural element that determines the exact dilation effect on the input image. To calculate the dilation of the input image using this structuring element, we consider each background pixel in the input image in turn. For each background pixel, the structural element is superimposed over the input image so that the origin of the structural element coincides with the position of the input pixel. If at least one pixel in the

structural element matches a foreground pixel in the image below it, then the input pixel is set to the foreground value. However, if all matching pixels in the image are background, the input pixel remains.

These mathematical operations help to extract contours in an image based on a binary mask.

To calculate the bounding box that wraps the contour, we need to find the minimum rectangle that joins all points of the contour.

Let  $Cnt$  be a contour found in the image consisting of an array of points with coordinates  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , that form a line or a closed shape of an object. We have  $Cnt = \{(x_1, y_1), (x_2, y_2)\}$ .

Mathematically, a bounding box consists of coordinates  $(x_1, y_1)$  of upper left corner and  $(x_2, y_2)$  of lower right corner. That is, to find the bounding box, it is necessary to find the smallest and largest coordinate values  $x$  and  $y$  among all points on the contour. We have the value calculated by formula (7):

$$Cnt = \{(\min(x_i), \min(y_i)), (\max(x_i), \max(y_i))\}, \quad (7) \\ i = 1, 2, \dots, n.$$

We will store the data in a csv file with the following features:

- “file\_name” is the relative name of the file in the folder;
- “xmin”, “ymin” are the coordinates of the upper left corner of the bounding box;
- “xmax”, “ymax” are the coordinates of the lower right corner of the bounding frame;
- “type” is the cell type (WBC for leukocyte, RBC for erythrocyte, and platelet)
- “area” is the area inside the bounding box.

#### 4 EXPERIMENTS

The developed pipeline was tested on images of the dataset [2, 22]. This dataset is released under the CC BY 4.0 license and consists of JPG images with dimensions of  $360 \times 363$  pixels.

The Python programming language in the Jupyter notebook environment was used for image processing. Third-party libraries were also used:

- PIL and opencv-python for image processing;
- matplotlib, seaborn for plotting graphs and diagrams;
- pandas for creating and processing a set of structured data;
- numpy – for processing images as arrays of pixels.

Fig. 7 shows the transformations that blood smear images undergo after each step of the algorithm.

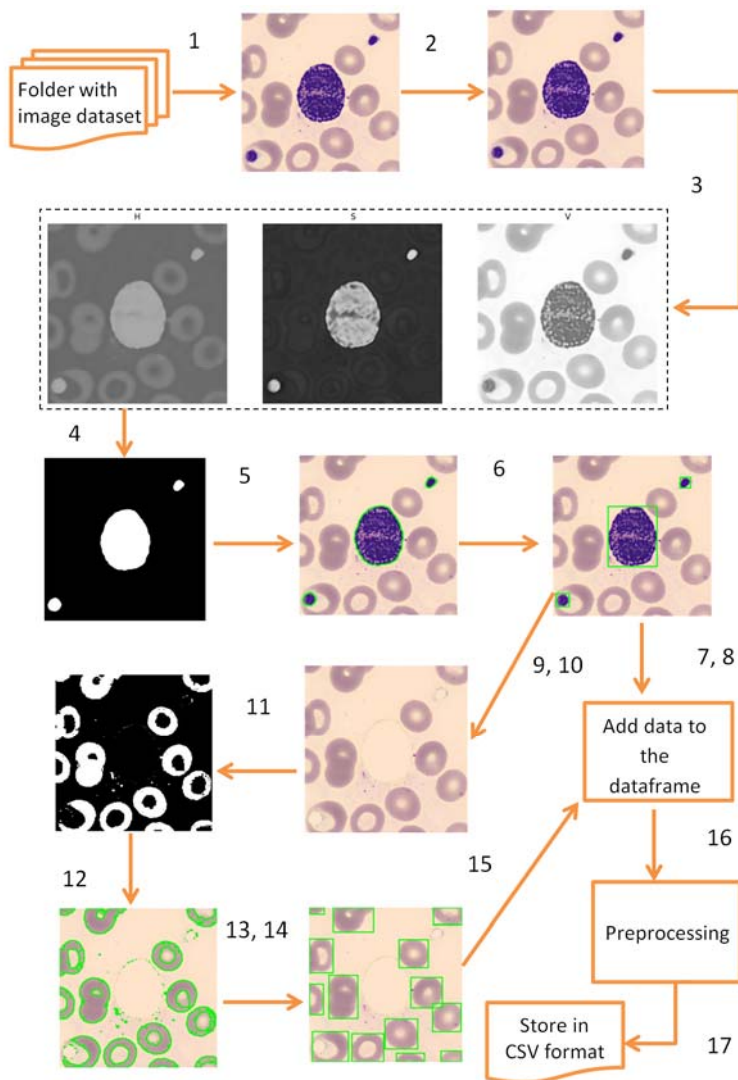


Figure 7 – Share of cells of each type of cells in the total number of detected objects

### 5 RESULTS

A total of 2159 RBCs, 177 WBCs, and 148 platelets were detected and localized, as shown in Fig. 8.

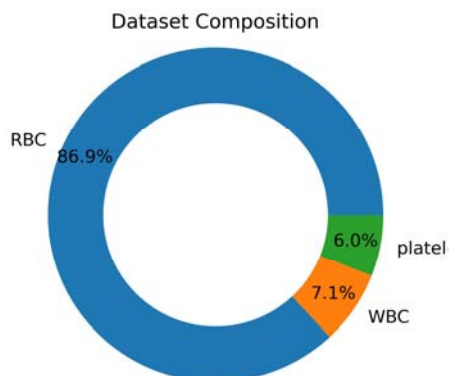


Figure 8 – Share of cells of each type of cells in the total number of detected objects

As a result, we receive a file in csv format, in which the coordinates of the bounding boxes, cell types on certain images are stored (Fig. 9).

	file_name	xmin	ymin	xmax	ymax	type	area
0	BA_206112.jpg	227	307	248	327	platelet	420
1	BA_206112.jpg	141	298	165	322	platelet	576
2	BA_206112.jpg	87	251	103	268	platelet	272
29	BA_208485.jpg	247	135	259	181	platelet	552
55	BA_209608.jpg	266	349	285	362	platelet	247

Figure 9 – Structure of the annotation dataframe

It took 9 minutes and 51 seconds to process 1000 images and create the resulting annotated dataset using the computer with Windows 10 operating system, 8GB of RAM, and an AMD Rizen 5 4500U processor using Python 3.10.0, pandas 1.3.5, and opencv-python 4.9.0.



## 6 DISCUSSION

To assess the quality of automatic annotation, bounding boxes were added to images and 160 images were processed manually.

The following cases were obtained during processing:

- cells were recognized correctly;
- incorrectly recognized leukocytes;
- unrecognized leukocytes;
- incorrectly recognized platelets;
- unrecognized platelets;
- incorrectly recognized erythrocytes;
- unrecognized erythrocytes.

Of course, cases b)-g) can be present in different combinations.

Fig. 10 shows most common cases: a) all objects are recognized; b) the presence of incorrectly recognized erythrocytes; c) recognition of overlapping erythrocytes as one object; c) the presence of unrecognized erythrocytes.

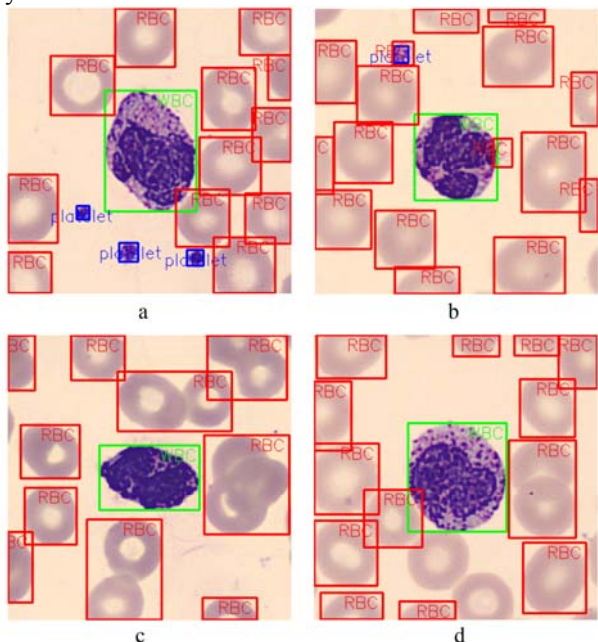


Figure 10 – Cases of cell recognition and localization

For further processing, the number of correct (True Positive, TP), incorrect (False Positive, FP) and unrecognized cells (False Negative, FN) of the considered types was calculated.

Let's define in more detail:

TP (True Positives) is the number of cells that are correctly recognized and assigned to a certain class;

FP (False Positives) is the number of cells that are recognized but incorrectly assigned to a certain class;

FN (False Negatives) is the number of cells that are not recognized as cells of a certain class.

The results are shown in Table 2.

Table 2 – Quality of recognition of blood cells

	Cell type		
	Leukocytes	Platelets	Erythrocytes
Correctly Recognized (TP)	160	127	2161
Incorrectly recognized (FP)	14	20	170
Not Recognized (FN)	1	13	178

To visualize the obtained results, let's build a heat map for the number of cells of each type and the number of correctly, incorrectly, and unrecognized blood cells (Fig. 11).

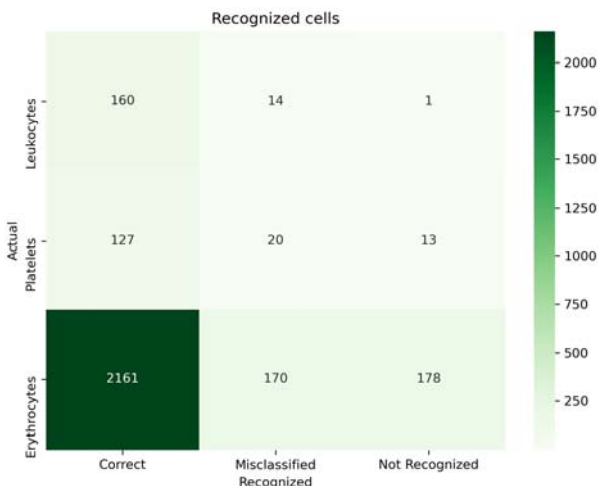


Figure 11 – Quality of cell recognition by type

To visually display the proportion of correctly, incorrectly, and unrecognized cells of each type in the total volume of cells, we will build a stacked chart (Fig. 12).

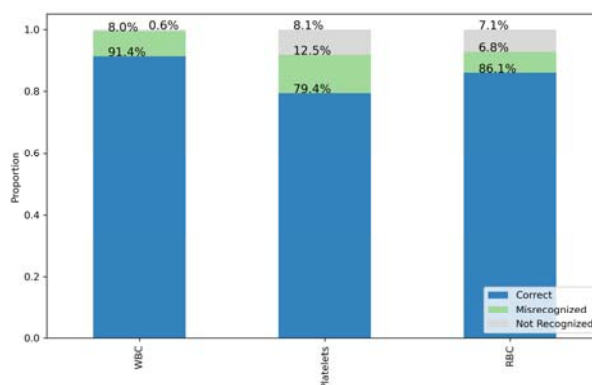


Figure 12 – Relative contribution of each cell type

Figure 12 shows that the number of correctly recognized white blood cells (WBCs) is 91.4% of the total, where 8% are misrecognized and only 0.6% WBCs are unrecognized. 79.4% of platelets are correctly recognized, 12.5% of the total are FPs, and 8.1% are not identified as platelets. TP red blood cells (RBCs) are 86.1%, 6.8% are FP, and 7.1% are FN.

The following metrics were calculated to assess the quality of object annotation:

– accuracy, which calculates the overall classification accuracy (7):

$$Acc = \frac{TP}{Total\ number}, \quad (7)$$

– precision, which estimates the percentage of objects that are correctly identified as a certain class (8):

$$Pr = \frac{TP}{TP + FP}, \quad (8)$$

– recall shows how many objects of a given class were correctly identified relative to all real objects of that class (9):

$$Rec = \frac{TP}{TP + FN}, \quad (9)$$

– F1 is the harmonic mean between precision and recall, which combines these two metrics into a single number and makes it possible to evaluate the effectiveness of the classification model, by choosing a balance between precision and recall (10):

$$F1 = 2 \left( \frac{Pr \cdot Rec}{Pr + Rec} \right). \quad (10)$$

The values of the calculated metrics are given in the table. 3

Table 3 – Quality metrics of automatic recognition of blood cells

	Cell type		
	Leukocytes	Platelets	Erythrocytes
Accuracy	0.9143	0.7937	0.8613
Precision	0.9195	0.8639	0.9271
Recall	0.9938	0.9071	0.9239
F1	0.9552	0.8850	0.9255

The obtained metric values indicate a sufficiently high efficiency of recognition of leukocytes and erythrocytes and insufficient efficiency of platelet recognition, which is associated with a high number of incorrectly recognized ones and not recognized cells. In our opinion, this is due to an unsuccessful choice of the threshold value when constructing the mask.

In further research, we plan to investigate the impact of different approaches to creating binary masks and contour extraction to improve the quality of cell recognition and localization. Also, this approach does not handle overlapping erythrocytes well enough. Processing of such types of cells is also the object of further research.

### CONCLUSIONS

In this paper, we propose an approach to automatically generate an annotated dataset for recognizing different

types of blood cells in an image. Our method is based on the use of image processing techniques to detect, segment and localize blood cells. The approach has been evaluated on a blood cell microscopy dataset and shown to achieve high accuracy. Our method has several advantages over existing manual methods for creating annotated datasets. First, it is fully automated, which makes it more efficient and scalable. Second, it can be used to create annotated datasets for any type of blood cell.

The following results were obtained:

1. An approach for automatically creating a set of annotated data for further use for object recognition using supervised machine learning methods is proposed and substantiated.
2. The pipeline is tested on a set of unlabeled data.
3. An annotated dataset is obtained consisting of cell images and a .csv file with the features “file\_name”, “type”, “xmin”, “ymin”, “xmax”, “ymax” which are bounding box coordinates for each object.
4. The number of correctly, incorrectly and unrecognized objects is counted and metrics were calculated to evaluate the accuracy.
5. Metrics for evaluating the quality of object detection and localization are obtained.

**The scientific novelty** of obtained results consists in the application of digital image processing methods to automatically create an annotated dataset.

**The practical significance** of the obtained results is a significant reduction in the time spent by machine learning specialists on creating an annotated dataset, which will positively affect the overall time spent on the model.

**Prospects for further research** are to extract overlapping cell contours and more accurate platelet recognition using other binarization and contour extraction algorithms.

### REFERENCES

1. Chadha G. K., Srivastava A., Singh A., Gupta R., Singla D. An Automated Method for Counting Red Blood Cells using Image Processing. *Procedia Computer Science, Procedia Computer Science*, 2020, Vol. 167, pp. 769-778. DOI: 10.1016/j.procs.2020.03.408.
2. Kovalenko S. Kovalenko S., Mikhnova O., Kovalenko A., Pelikh D., Severin V. An Approach to Blood Cell Classification Based on Object Segmentation and Machine Learning, *IEEE 4th KhPI Week on Advanced Technology (KhPI-Week)*, 2023, pp. 1–6. DOI: 10.1109/KhPIWeek61412.2023.10312903.
3. 2022 State of Data Science by Anaconda [Electronic resource]. Access mode: <https://www.anaconda.com/resources/whitepapers/state-of-data-science-report-2022>.
4. Aljabri M., AlAmir M., AlGhamdi M., Abdel-Mottaleb M., Collado-Mesa F. Towards a better understanding of annotation tools for medical imaging: a survey, *Multimedia tools and applications*, 2022, Vol. 81(18), 202225877-25911. DOI: 10.1007/s11042-022-12100-1.
5. Kutsenko A. S., Megel Y. Y., Kovalenko S. V., Kovalenko S. M., Omiotek Z., Zhunisova U. An approach to quality evaluation of embryos based on their geometrical parameters, *Proc. SPIE 11176, Photonics Applications in Astron-*

- omy, Communications, Industry, and High-Energy Physics Experiments, 2019. 111762G. DOI: 10.1117/12.2536420.
6. Megel Y., Kalimanova I., Rybalka A., Kovalenko S., and Kovalenko S. Automation of measurement of objects geometrical parameters, *Proceedings of International Scientific Symposium "Metrology and Metrology Assurance"*, 2017, pp. 225–259.
  7. Burger W., Burge M. J. Digital Image Processing: An Algorithmic Introduction. Springer Nature, 2022, 927 p. DOI: 10.1007/978-1-4471-6684-9
  8. Chityala R., Sridevi P. Image processing and acquisition using Python. CRC Press, 2020, 420 p. DOI: 10.1201/9780429243370.
  9. Dey S. Python image processing cookbook: over 60 recipes to help you perform complex image processing and computer vision tasks with ease. Packt Publishing Ltd, 2020, 438 p. ISBN: 978-1-789537147.
  10. Gaudenz B. Object Detection in 2024: The Definitive Guide. [Electronic resource]. Access mode: <https://viso.ai/deep-learning/object-detection/>.
  11. Plataniotis K., Venetsanopoulos A. N. Color Image Processing and Applications. Springer Science & Business Media, 2000, 355 p. DOI: 10.1007/978-3-662-04186-4.
  12. Zhuang F., Qi Z., Duan K., Xi D., Zhu Y., Zhu H., Xiong H., He Q. A Comprehensive Survey on Transfer Learning, *Proceedings of the IEEE*, 2021. Vol. 109, No. 1, pp. 43–76. DOI: 10.1109/JPROC.2020.3004555.
  13. Mikołajczyk A. and Grochowicki M. Data augmentation for improving deep learning in image classification problem, *2018 International Interdisciplinary PhD Workshop (IIPHDW)*. Świnouście, Poland, 2018, pp. 117–122. DOI: 10.1109/IIPHDW.2018.8388338.
  14. Bonaccorso G. Machine Learning Algorithms – Second Edition. Packt Publishing Ltd., 2018, 522 p. ISBN 978-1-78934-799-9.
  15. Szeliski R. Image-Based rendering, *Computer Vision. Texts in Computer Science*. Springer, Cham, 2022, pp. 681–722. DOI: 10.1007/978-3-030-34372-9\_14.
  16. Brownlee J. A gentle introduction to object recognition with deep learning, *Machine Learning Mastery 5* [Electronic resource]. Access mode: <https://machinelearningmastery.com/object-recognition-with-deep-learning/>.
  17. Sinha R. K., Pandey R., Pattnaik R. Deep learning for computer vision tasks: a review, *arXiv preprint arXiv:1804.03928*, 2018. DOI: 10.26438/ijcse/v7i7.195201.
  18. Lu L., Wang X., Carneiro G., Yang L. Deep learning and convolutional neural networks for medical imaging and clinical informatics. Berlin/Heidelberg, Germany, Springer International Publishing, 2019, 461 p. DOI: 10.1007/978-3-030-13969-8.
  19. Megel Yu., Chaly I., Kovalenko S., Mikhnova O. Doslidzhennja dyhal'nyh ruhiv za dopomogoj ekspertnoi systemy na bazi obchysljuvalnogo intelektu [Study of Respiratory Movements Using an Expert System Based on Computational Intelligence], *Systemy obrobky informacii [System. Information Processing Systems]*, 2022, № 3 (170), pp. 41–46. DOI: 10.30748/soi.2022.170.05.
  20. Tigner A., Ibrahim S. A., Murray I. V. Histology, White Blood Cell, *StatPearls Publishing*, 2022 [Electronic resource]. Access mode: <https://www.ncbi.nlm.nih.gov/books/NBK563148>.
  21. OpenCV. Color conversions [Electronic resource]. Access mode: [https://docs.opencv.org/3.4/de/d25/imgproc\\_color\\_conversions.html](https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html).
  22. Acevedo A., Merino A., Alférez S., Molina Á., Boldú L. and Rodellar J. A dataset for microscopic peripheral blood cell images for development of automatic recognition systems, *Data in brief*, 2022, Vol. 30. 105474. DOI: 10.1016/j.dib.2020.105474.

Received 26.01.2024.  
Accepted 28.02.2024.

UDC 004.932.2:004.93'1

## ПІДХІД ДО АВТОМАТИЧНОГО СТВОРЕННЯ АНОТОВАНОГО ДАТАСЕТУ ДЛЯ ВИЯВЛЕННЯ, ЛОКАЛІЗАЦІЇ ТА КЛАСИФІКАЦІЇ КЛІТИН КРОВІ НА ЗОБРАЖЕННІ

**Коваленко С. М.** – канд. техн. наук, доцент, доцент кафедри програмної інженерії та інтелектуальних технологій управління Національного технічного університету «Харківський політехнічний інститут», Харків, Україна.

**Куценко О. С.** – д-р техн. наук, професор, професор кафедри системного аналізу та інформаційно-аналітичних технологій Національного технічного університету «Харківський політехнічний інститут», Харків, Україна.

**Коваленко С. В.** – канд. техн. наук, доцент, професор кафедри системного аналізу та інформаційно-аналітичних технологій Національного технічного університету «Харківський політехнічний інститут», Харків, Україна.

**Коваленко А. С.** – аспірант кафедри системного аналізу та інформаційно-аналітичних технологій Національного технічного університету «Харківський політехнічний інститут», Харків, Україна.

### АНОТАЦІЯ

**Актуальність.** Розглянуто проблему автоматизації створення анотованого набору даних для його подальшого використання в системі виявлення, локалізації та класифікації клітин крові на зображенні з використанням глибокого навчання. Об'єктом дослідження є процеси обробки цифрових зображень для виявлення та локалізації об'єктів.

**Мета роботи** – розробка пайплайну із послідовності методів обробки цифрових зображень для автоматичного створення анотованого набору зображень мазків крові з подальшим використанням для навчання та валідації моделей глибокого навчання, що має суттєво скоротити час спеціалістів з машинного навчання.

**Метод.** Запропонований підхід для виявлення та локалізації об'єктів базується на методах обробки цифрових зображень: методах фільтрації, порогової фільтрації, бінаризації, знаходження та заливки контурів тощо. Пайплайн по виявленню та локалізації складається з наступних кроків: приглушення шумів; перетворення в HVS кольорову модель; визначення маски для лейкоцитів та тромбоцитів; визначення контурів лейкоцитів та тромбоцитів; визначення координат верхнього лівого та правого нижнього кутів лейкоцитів та тромбоцитів; обчислення площі області всередині обмежувальної рамки; збереження

отриманих даних; визначення найпоширенішого кольору на зображенні; заливка цим кольором контурів лейкоцитів та тромбоцитів; визначення маски для еритроцитів; визначення контурів еритроцитів; визначення координат верхнього лівого та правого нижнього кутів еритроцитів; обчислення площі області всередині обмежувальної рамки; занесення до датафрейму даних про знайдені об'єкти; збереження в файлі .csv для подальшого використання.

**Результати.** Розроблений підхід був впроваджений у програмне забезпечення для створення анотованого набору даних зображень мазків крові.

**Висновки.** В дослідженні запропоновано та обґрунтовано підхід для автоматичного створення набору анотованих даних. Пайплайн протестовано на наборі нерозмічених даних та отримано набір розмічених даних, що складається з зображень клітин та файлу в форматі .csv, що має ознаки «назва файлу», «тип клітини», «xmin», «ymin», «xmax», «ymax», що є координатами обмежувальної рамки для кожного об'єкту. Підраховано кількість правильно, неправильно та нерозпізнаних об'єктів та розраховано метрики для оцінки точності та якості виявлення та локалізації об'єктів.

**КЛЮЧОВІ СЛОВА:** комп'ютерний зір, визначення об'єктів, локалізація об'єктів, обробка цифрових зображень, класифікація.

## ЛІТЕРАТУРА

1. An Automated Method for Counting Red Blood Cells using Image Processing. *Procedia Computer Science* / [G. K. Chadha, A. Srivastava, A. Singh et al.] // *Procedia Computer Science*. – 2020. – Vol. 167. – P. 769–778. DOI: 10.1016/j.procs.2020.03.408.
2. An Approach to Blood Cell Classification Based on Object Segmentation and Machine Learning / [S. Kovalenko, S. Kovalenko, O. Mikhnova et al.] // *IEEE 4th KhPI Week on Advanced Technology (KhPIWeek)*. – 2023. – P. 1–6. DOI: 10.1109/KhPIWeek61412.2023.10312903.
3. 2022 State of Data Science by Anaconda [Electronic resource]. – Access mode: <https://www.anaconda.com/resources/whitepapers/state-of-data-science-report-2022>.
4. Towards a better understanding of annotation tools for medical imaging: a survey / [M. Aljabri, M. AlAmir, M. AlGhamdi et al.] // *Multimedia tools and applications*. – 2022. – Vol. 81(18). – 202225877–25911. DOI: 10.1007/s11042-022-12100-1.
5. An approach to quality evaluation of embryos based on their geometrical parameters / [A. S. Kutsenko, Y. Y. Megel, S. V. Kovalenko et al.] // *Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments*. – 2019. – 111762G. DOI: 10.1117/12.2536420.
6. Automation of measurement of objects geometrical parameters / [Y. Megel, I. Kalimanova, A. Rybalka, S. Kovalenko, and S. Kovalenko] // *Proceedings of International Scientific Symposium “Metrology and Metrology Assurance”*. – 2017. – P. 225–259.
7. *Burger W. Digital Image Processing: An Algorithmic Introduction* / W. Burger, M. J. Burge. – Springer Nature, 2022. – 927 p. DOI: 10.1007/978-1-4471-6684-9.
8. Chityala R. *Image processing and acquisition using Python* / R. Chityala, P. Sridevi. – CRC Press, 2020. – 420 p. DOI: 10.1201/9780429243370.
9. Dey S. *Python image processing cookbook: over 60 recipes to help you perform complex image processing and computer vision tasks with ease* / S. Dey – Packt Publishing Ltd, 2020. – 438p. ISBN: 978-1-789537147.
10. Gaudenz B. *Object Detection in 2024: The Definitive Guide*. / B. Gaudenz [Electronic resource]. – Access mode: <https://viso.ai/deep-learning/object-detection/>.
11. Plataniotis K. *Color Image Processing and Applications* / K. Plataniotis, A. N. Venetsanopoulos. – Springer Science & Business Media, 2000. – 355 p. DOI: 10.1007/978-3-662-04186-4.
12. A Comprehensive Survey on Transfer Learning / [F. Zhuang, Z. Qi, K. Duan et al.] // *Proceedings of the IEEE*. – 2021. – Vol. 109, No. 1. – P. 43–76. DOI: 10.1109/JPROC.2020.3004555.
13. Mikołajczyk A. Data augmentation for improving deep learning in image classification problem / A. Mikołajczyk and M. Grochowski // *2018 International Interdisciplinary PhD Workshop (IIPHDW)*. – Świnouście, Poland, 2018. – P. 117–122. DOI: 10.1109/IIPHDW.2018.8388338.
14. Bonaccorso, G. *Machine Learning Algorithms – Second Edition* / G. Bonaccorso. – Packt Publishing Ltd., 2018. – 522 p. ISBN 978-1-78934-799-9.
15. Szeliski R. *Image-Based rendering* / R. Szeliski // *Computer Vision. Texts in Computer Science*. – Springer, Cham, 2022. – P. 681–722. DOI: 10.1007/978-3-030-34372-9\_14.
16. Brownlee J. *A gentle introduction to object recognition with deep learning*. / J. Brownlee // *Machine Learning Mastery 5* [Electronic resource]. – Access mode: <https://machinelearningmastery.com/object-recognition-with-deep-learning/>.
17. Sinha R. K. *Deep learning for computer vision tasks: a review* / R. K. Sinha, R. Pandey, R. Pattnaik // *arXiv preprint arXiv:1804.03928*, 2018. DOI: 10.26438/ijcse/v7i7.195201.
18. *Deep learning and convolutional neural networks for medical imaging and clinical informatics* / [L. Lu, X. Wang, G. Carneiro, L. Yang]. – Berlin/Heidelberg, Germany : Springer International Publishing, 2019. – 461 p. DOI: 10.1007/978-3-030-13969-8.
19. Дослідження дихальних рухів за допомогою експертної системи на базі обчислювального інтелекту / [Ю. Є. Мегель, І. В. Чалий, С. М. Коваленко, О. Д. Міхнова] // *Системи обробки інформації*. – 2022. – № 3 (170). – P. 41–46. DOI: 10.30748/soi.2022.170.05.
20. Tigner A. *Histology, White Blood Cell* // A. Tigner, S. A. Ibrahim, I. V. Murray // *StatPearls Publishing*, 2022 [Electronic resource]. – Access mode: <https://www.ncbi.nlm.nih.gov/books/NBK563148>
21. *OpenCV. Color conversions* [Electronic resource]. – Access mode: [https://docs.opencv.org/3.4/de/d25/imgproc\\_color\\_conversions.html](https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html).
22. A dataset for microscopic peripheral blood cell images for development of automatic recognition systems / [A. Acevedo, A. Merino, S. Alferez et al.] // *Data in brief*. – 2002. – Vol. 30. – 105474. DOI: 10.1016/j.dib.2020.105474.