



METHODS OF EXTRACTING ELECTROCARDIOGRAMS FROM ELECTRONIC SIGNALS AND IMAGES IN THE PYTHON ENVIRONMENT

Bakhytgul ZHOLMAGAMBETOVA¹, Talgat MAZAKOV², Sholpan JOMARTOVA³,
Adilzhan IZAT⁴, Olzhas BIBALAYEV⁵

¹ Department of Information Technology and Security, Karaganda State Technical University, Shakhtarov Avenue, 70-191, Karaganda, 100000, Republic of Kazakhstan, bakhytgul_zholmagambetova@rambler.ru

² Department of Computer Science, Al-Farabi Kazakh National University, Zhanar st. 37a,1, Almaty, 500045, Republic of Kazakhstan, e-mail: talgat_mazakov@rambler.ru

³ Department of Artificial Intelligence and Big Data, Al-Farabi Kazakh National University, Zhanar st. 37a,1, Almaty, 500045, Republic of Kazakhstan, e-mail: sholpan_jomartova@rambler.ru

⁴ Department of Applied Mathematics and Computer Science, Karaganda State University named after Academician E. A. Buketov, Shakhtarov Avenue, 70-191, Karaganda, 100000, Republic of Kazakhstan, e-mail: adilzhan_izat@rambler.ru

⁵ Department of Robotics and Intelligent Machine, Scientific Research Laboratory, Karaganda State University named after Academician E. A. Buketov, Satybalдина str., 16-53, Karaganda, 100000, Republic of Kazakhstan, e-mail: olzhas_bibalayev@rambler.ru

Abstract

High-quality signal processing of an electrocardiogram (ECG) is an urgent problem in present day diagnostics for revealing dangerous signs of cardiovascular diseases and arrhythmias in patients. The used methods and programs of signal analysis and classification work with the arrays of points for mathematical modeling that must be extracted from an image or recording of an electrocardiogram. The aim of this work is developing a method of extracting images of ECG signals into a one-dimensional array. An algorithm is proposed based on sequential color processing operations and improving the image quality, masking and building a one-dimensional array of points using Python tools and libraries with open access. The results of testing samples from the ECG database and comparing images before and after processing show that the signal extraction accuracy is approximately 95 %. In addition, the presented application design is simple and easy to use. The proposed program for analyzing and processing the ECG data has a great potential in the future for the development of more complex software applications for automatic analyzing the data and determining arrhythmias or other pathologies.

Keywords: ECG signal, MIT/BIH, Python, image processing, one-dimensional array, OpenCV, Matplotlib, NumPy.

1. INTRODUCTION

The electrocardiogram (ECG) signal shows a lot of information of the condition of the patient's cardiovascular system. One of the important tests in the ECG is detecting cardiac arrhythmias. Monitoring and timely detecting dangerous arrhythmias in a patient will help preventing the threat of stroke or sudden death from heart failure [1]. In a lot of works [2,3], studies have been carried out aimed at identifying life-threatening arrhythmias. This process is a long and tedious procedure for both the patient and the cardiologist. Automation of the ECG analysis process and computer diagnostics are the key and necessary technologies in present day clinical cardiology [4]. The development of automated diagnostic tools can be crucial for the correct diagnosis of cardiovascular disease.

To this end, a lot of studies have been carried out in which various methods of processing and extracting signs of arrhythmias from the ECG signals have been used. Such methods include time domain analysis [5], statistical method [6], hybrid function method [7], frequency and time-frequency analysis [8-10]. However, due to significant differences in the time and morphological characteristics of the ECG signal among different patients, there is still an unresolved problem of accurate detecting foci of violation, which prevents the development of a general classification of signals and requires an individual approach. Since the ECG signal is a whole time-frequency set consisting of sets of harmonic components of heartbeats, network interferences, noise, muscle tremor, etc. For example, the ventricular QRS complex, which is a combination of the Q, R and S waves recorded during ventricular pacing, is the largest and most important complex that is visible

on the ECG. The Q and S waves are downward waves that represent depolarization of the interventricular septum from left-to-right and depolarization of the left ventricle of the heart, respectively. The R wave is an upward wave that represents the excitation of the ventricles of the heart. Therefore, recognition of the necessary signals after applying various filters is also very complex, especially with a weak signal intensity. Therefore, the development of high-precision methods for analyzing electrocardiogram data is relevant to this day.

Over the past decade the scientific literature has presented a lot of works [5,11-13] dealing with the development of new methods of the automatic ECG analysis. For example, using the discrete wavelet transform tool, you can increase the accuracy and sensitivity of signal recognition [14,15]. This approach that is programmed in the MATLAB environment for the MIT/BIH arrhythmia database, involves removal of artifacts and detection of a complex of QRS and arrhythmias. Another works [16,17] proposed Hermite transformations as a tool for analyzing the ECG signals.

From the above methods, it follows that for more accurate determining the arrhythmia signals for data processing, it is necessary to have high-quality ECG signals that can be obtained directly when measured on a patient. However, unfortunately, highly accurate and sensitive equipment is not always available or is not available locally due to the high cost and needed maintenance [18].

Besides, not all programs can recognize jpg and tiff file formats because the information processing is performed with an array of points with which you can work in a computing environment in special software applications. This paper attempts to redress the imbalances in previous studies through the development of a program for obtaining a clean ECG signal with a high resolution quality in the form of a one-dimensional array with a scanned electrocardiogram or electronic file recording. In this article we propose a new method of image processing using batch applications in the Python programming language. The novelty of our approach is that it is based on an easy-to-use and effective software package, which is designed to quickly convert ECG data into a format that is suitable for decoding and analysis. This approach enables an accurate diagnosis. From scientific and applied points of view, the results may be useful in designing medical and other computer programs and mobile applications.

2. MATERIALS AND METHODS

2.1. Tools and software packages

To achieve the intended result, one needs to select the right tool, packages and the programming language. The Python was selected as the programming language. It is one of the powerful

tools for creating programs and projects. This language is simple, accessible and adapted to up-to-date operating systems and applications. The Python has object-oriented, structural, high-level, functional skills. In this study, we are using Python version 4.2.0 that provides a sufficient amount of programming tools, including many built-in functions. In addition, there are many open source external libraries for the Python that are available in the Internet. In our study, the libraries OpenCV, Matplotlib, NumPy [19,20] will be used for the effective image processing in the script.

x_array is an open source Python package that provides tools and data structures for N-dimensional labeled arrays [21]. This tool allows combining the application programming interface with a common data model, includes indexing and label-based arithmetic, it is compatible with *NumPy*, *Matplotlib*.

2.2. Initial data

The ECG signal images (Fig. 1) in the form of files of two formats *.dat and *.hea were taken from the public database MIT/BIH (<https://physionet.org/> and <https://ecgpedia.org/>).

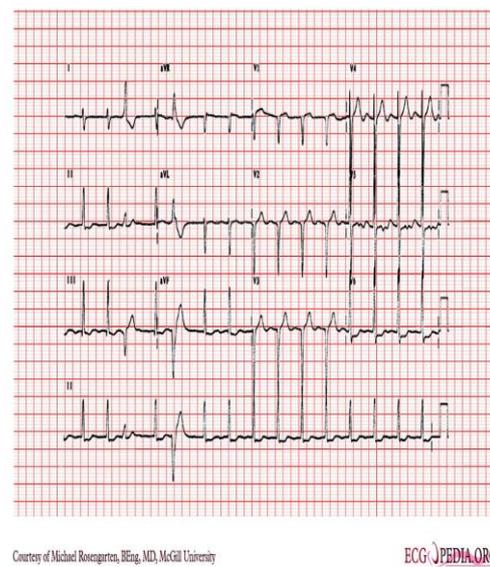


Fig. 1. An image with the ECG signal scan

For further work, one strip of the electrocardiogram was cut out of each image (Fig. 2). To work in the application, files containing high-resolution images were selected. As a result, out of 500 files, 100 images and 100 records (100 *.dat files and 100 *.hea files) were selected that met the needed requirements. The PyQt5 library was used to build the interface, because the interface of this application is very simple and understandable for an average user.

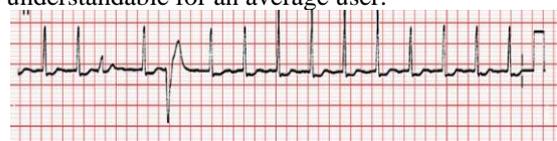


Fig. 2. One-strip image of the ECG signal

2.3. Building the Algorithm of Image Processing in the Python

Depending on the downloaded data type, the program selects the appropriate data processing algorithm (see Fig.3).

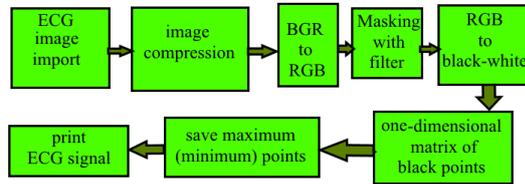


Fig. 3. Block diagram of the proposed algorithm of image processing.

Let's consider them separately. First, let's consider the algorithm of processing the downloaded image. For the beginning let's import the libraries:

```

import numpy as np
import cv2
import os
import matplotlib.pyplot as plt

```

Read off the image file through the path to the file (FILE) that was obtained from the loader program:

```
bgr_img_array = cv2.imread(FILE)
```

Now the `bgr_img_array` variable stores a matrix of image points. Each point now contains the information of its color in the BGR color format. After converting the pattern into the matrix, we can observe single points that will subsequently appear in the form of noise. To eliminate single points that appear due to noise, you can apply image compression to filter noise. It is worth noting that the quality of the downloaded image should be above 100 pixels, which is prescribed in the following way:

```

height = 100
width =
bgr_img_array.shape[1]/bgr_img_array.shape[0]*height
resized_bgr_img_array =
cv2.resize(bgr_img_array, (int(width), height))

```

Having completed the above operations, we will compress the image to eliminate single noise. Now the compressed matrix, to which the noise filter has been applied, belongs to the variable `resized_bgr_img_array`. Since the color of each point is stored in the BGR format, it is necessary to change the color data to the RGB format, which is written as follows:

```

b,g,r = cv2.split(resized_bgr_img_array)
rgb_img = cv2.merge([r,g,b])

```

Now the point color data is stored in the RGB format.

Next, unnecessary colors should be excluded from the matrix of the presented electrocardiogram (Fig. 2). Since the color data is now stored in the [R, G, B] format, we can get rid of colors that are different from black through masking from the OpenCV library. To do this, we define the color boundaries of the desired points with the lower (smallest value) and upper (largest value) variables. For convenience, in further code writing, the matrix in which the image data is stored is called the *image*. Now let's create a mask through which we eliminate unnecessary colors in the image and call it the *mask*. We set the output variable to the matrix that has already been masked, and fill the empty points with white. The code for this program has the form:

```

lower = np.array([0,0,0], dtype = "uint8")
upper = np.array([150,150,150], dtype = "uint8")
image = resized_bgr_img_array
mask = cv2.inRange(image, lower, upper)
output = cv2.bitwise_and (image, image, mask = mask)
output[mask==0] = (255,255,255)

```

Now for the final selection of the black color and its shades, we convert the image to black and white from the RGB format. To do this, use the following function:

```

out_gray = cv2.cvtColor(output, cv2.COLOR_BGR2GRAY)

```

The value of the `out_gray` variable is the matrix with the remaining points, for which the color value is from 0 to 150, i.e. from black to gray. In the matrix there are still white dots. To compose a one-dimensional matrix, you need to check the values from top to bottom on each vertical line. This is as follows. If the image width is X and the height is Y, then we must take the line X and check the black points at the height Y. To do this, replace the X values with Y in the `out_gray` variable, because horizontal lines in it represent Y, and X is considered to be the height:

```
out_gray_reverse = np.swapaxes(out_gray, 0, 1)
```

Then we can compile the matrix. We pass along each line X and Y, comparing each point with black color. We write the result to the previously initialized variable:

```

ecg_points_all_blacks = []

for i in range (0, len(out_gray_reverse)):
    vertical_line_founds = []
    for j in range (0, len(out_gray_reverse[i])):
        if(int(out_gray_reverse[i][j]) < 150):

```

```

vertical_line_founds.append(j)
ecg_points_all_blacks.append(vertical_line_founds
)

```

The value of `ecg_points_all_blacks` is a two-dimensional matrix in which the first level is the line where the black points were. This line contains the coordinates of the Y axis for each black point, which have the following form:

```
[12, 26], [12, 34], [18, 23, 32, 40], 41, 46, 46,
46, 46, 46, 46, 46,
```

where on the first line here are two black points with the coordinates 12 and 26; on the second line there are two black points with the coordinates 12 and 34. There are a number of black points like in [15-17]. Equating the value in a one-dimensional array with the arithmetic mean will lead in the future to the problem of calculating extrema: The R, Q, and S peaks. Therefore, the best solution is to maintain the common midpoint by summing the coordinates along the Y axis of all black points in the given line, which is then divided by their number.

Applying the check to each point for belonging to the lower or upper side of the midline, depending on the result, we save the maximum or minimum points in the rows, respectively, as in [15-17]. Thus, we remove a lot of points that stand behind each other. We also keep extrema showing only maxima (preservation of minima is performed by an identical code):

```

ecg_points_blacks_maximums = []
for i in range(0, len(ecg_points_all_blacks)):
    if(len(ecg_points_all_blacks[i]) == 0):
        continue
    last_point = 0
    max_points = []
    for j in range(0,
len(ecg_points_all_blacks[i])):
        if (j == 0):
            last_point = ecg_points_all_blacks[i][j]
            max_points.append(last_point)
        elif(j != 0 and last_point !=
(ecg_points_all_blacks[i][j] - 1)):
            last_point = ecg_points_all_blacks[i][j]
            max_points.append(last_point)
        else:
            last_point = ecg_points_all_blacks[i][j]
            continue
    if(len(max_points) == 1):
        max_points = max_points[0]
ecg_points_blacks_maximums.append(max_points)

```

Concatenate the matrixes with the points removing identical points and derive the result as follows:

```
print(ecg_points_blacks_result)
```

```
print ("Signal length: " +
len(ecg_points_blacks_result) + "points")
```

In the console window we will see the result, which is given as an example:

```
[45, 45, 45, 45, 45, 45, 44, 43, 43, 43, 43, 43,
43, 43, 31, 14, 12, .....,
15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
38, 43]
```

The signal length is 748 points.

2.4. The extracting the data from the files with .dat and .hea extensions

In this context it is much simpler, because in up-to-date devices electronic recordings are performed with great frequency, and the signals are immediately recorded in the form of a one-dimensional array. The problem is how to read this data correctly. These files store the following information: the sampling frequency, the signal length, the recording date, the recording time, the time zone in which the recording was made, the unit of measurement, the signal name, comments and many other values. To read electronic signals and derive points from them, the `wfdb` library will help us, which is designed to work with file data from the MIT database and work with *.dat and *.hea files. Having selected the files in the program and clicking the Download button, we identify the new name for the two files. The files names must be the same.

```
record = wfdb.rdsamp(PATH)
points = record [0]
```

where `PATH` is the path to the files and previously identified file names, `points` is the one-dimensional array with all the points.

4. RESULTS AND DISCUSSION

The interface of the developed application for signal conversion is shown in Fig. 4. As you can see, in the application window it is proposed to select the type of data that is presented as an image or record. Depending on the type selected, the program will request the needed files for reading. You can see in the Figure that, when selecting an entry from MIT/BIH, the program offers to select two corresponding files that are stored in different formats. Otherwise, you need to select only one scan or photo file. The interface of the developed application looks quite simple and understandable for any user and does not require additional instructions and introduction of other operations for downloading the file.

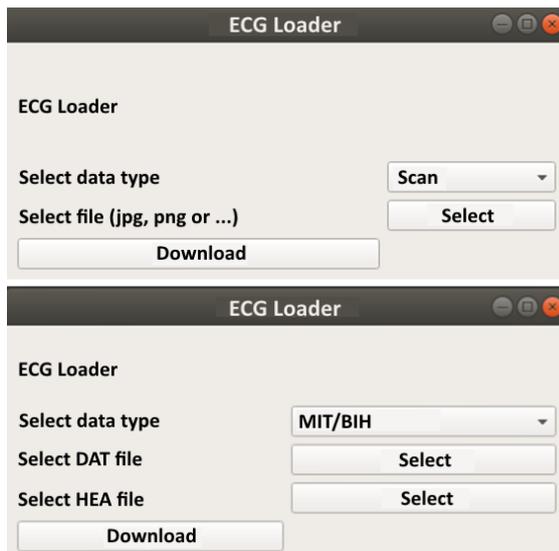


Fig. 4. Application interface for reading the ECG signals from images and electronic records

After selecting a file in the bootloader window, a window opens on the screen with a graph of the image in the form of a one-dimensional array (Fig. 5) that can be used for the application that analyzes the provided full ECG signal.

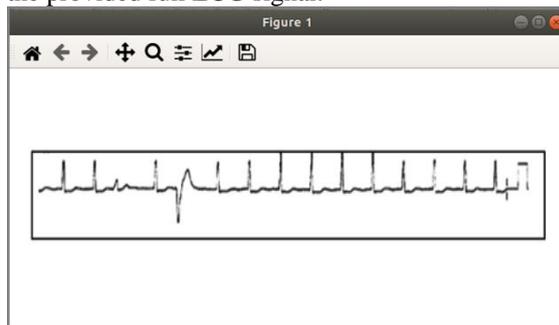


Fig. 5. Graph of the one-dimensional array built using the *Matplotlib* library

You can see that the proposed algorithm for processing scans and other images of electrocardiograms shows good results. The masking approach used, with the exception of unnecessary colors, turned out to be very effective. It can be seen in Fig. 6 that in the comparison of the results before and after processing the scan, the developed program quite accurately identifies the position of the points corresponding to the ECG signal. In addition, the scale and the slightest deviations are preserved, which can conceal very important information during decryption.

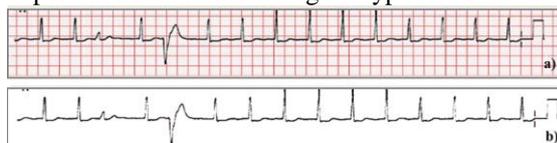


Fig. 6. ECG signal before (a) and after (b) processing by developed application

Similar results were shown after processing MIT/BIH records. The comparative analysis of 100 image samples before and after processing (as

shown on Fig. 6) shows a complete match of ECG signals. Such a high indicator is also associated with the preliminary image preparation and the correct sequence of recognition of the point position that were proposed in the development of signal recognition algorithms. Thus, this approach increases the sensitivity and accuracy of the ECG image processing technique, and the project developed in the Python is a quick and easy to use application, which is highly needed when analyzing the signal and its classification. For example, such signs are used to classify the ECG signals as the beginning and wave displacement, QRS, and the period that are detected by analyzing in the time domain of the signal [1].

A lot of other approaches were used to extract a one-dimensional array from an image. In [22], a method of recognizing the waveform in the time domain based on a one-dimensional convolution neural network is proposed. In this approach, building a neural network for direct recognition of patterns in the form of a wave is proposed. A slightly different approach was applied in [23] for processing images of the Raman fiber optic sensors. The essence of the technique proposed by the authors was to combine one-dimensional Raman signals into a two-dimensional array in order to suppress noise and improve the signal itself during processing. The presented results of the work, as in this article, demonstrate the high efficiency of the image processing techniques used to improve the signal quality. However, the proposed methods take into account only the specifics of the signal for which they were developed, and are not universal in contrast to our approach. The algorithm developed in this work can be improved and adapted to extract not only the ECG signals but also for other measurements that use a similar technique for recording data.

A similar approach was used in [24], where the authors developed an algorithm for restoring images in an array using a one-dimensional integrated image processing system. Based on the theory of geometric optics and the theory of studying the depth of images, an algorithm was developed to fill in the missing points using a simulated image depth map by comparing and evaluating the image quality. Khamdamov et al. [25] introduced a more advanced technology for processing raster images. The novelty of their approach was the use of the Daubechies wavelet function to filter, compress, and smooth two-dimensional signals. The computational process algorithm is presented in the OpenMP environment in the C/C++, which is more complex and not comparable for all the applications compared to the Python and its available library and open source toolkit.

5. CONCLUSIONS

According to the presented methodology and results, we can conclude that in the present work a new technique of processing images and recordings the ECG signals in the form of a one-dimensional array is developed. Using sequential operations of selecting the color processing format, improving the image quality, masking and building a one-dimensional array of points allows recognizing the ECG signal with high accuracy. The comparative image analysis before and after processing demonstrates a good conformity of ECG signals, which indicates the high sensitivity of the proposed method. In addition, using open source Python packages and libraries allows developing a simple and affordable application design for quick processing. The research goal was achieved by developing a Python application for the conversion of photo or scan images into a one-dimensional array of points, which can be further processed with a decoding program. This technique and the proposed program for analyzing and processing the ECG data have a great potential in the future for the development of more complex applications for the automatic data analysis and faster determining arrhythmias and other pathologies in patients.

SOURCE OF FUNDING

The work has been carried out as a part of a grant in 2018-2020 for the project AR05132044 "Development of a Hardware-medical Complex for Assessing the Psycho-physiological Parameters of a Person".

REFERENCES

- Bhirud B, Pachghare VK. Arrhythmia Detection Using ECG Signal: A Survey. In Proceeding of the International Conference on Computational Science and Applications, Springer, Singapore; 2020, 329-341.
- Trayanova NA, Boyle PM, Nikolov PP. Personalized imaging and modeling strategies for arrhythmia prevention and therapy. *Current opinion in biomedical engineering* 2018; 5: 21-28.
- Chatha SR, Chahal CAA, Westwood M, Khanji M. Arrhythmia and sarcoidosis: the role of myocardial tissue characterization to guide diagnosis and management. *European Heart Journal-Cardiovascular Imaging* 2020; 21(3): 344-344. <http://dx.doi.org/10.1093/ehjci/jez229>.
- Abdelazim IA, Bekmukhambetov Y, Aringazina R, Shikanova S, Amer OO, Zhurabekova G., Otessin MA, Astrakhanov AR. The outcome of hypertensive disorders with pregnancy. *Journal of Family Medicine and Primary Care* 2020; 9(3): 1678
- Raj S, Maurya K, Ray KC. A knowledge-based real time embedded platform for arrhythmia beat classification. *Biomedical Engineering Letters* 2015; 5(4): 271-280. <https://dx.doi.org/10.1007/s13534-015-0196-9>
- Ye C, Kumar BV, Coimbra MT. Heartbeat classification using morphological and dynamic features of ECG signals. *IEEE Transactions on Biomedical Engineering* 2012; 59(10): 2930-2941. <https://dx.doi.org/10.1109/TBME.2012.2213253>
- Zhang Z, Dong J, Luo X, Choi KS, Wu X. Heartbeat classification using disease-specific feature selection. *Computers in biology and medicine* 2014; 46: 79-89. <https://dx.doi.org/10.1142/S0219519419500040>
- Yıldırım Ö, Pławiak P, Tan RS, Acharya UR. Arrhythmia detection using deep convolutional neural network with long duration ECG signals. *Computers in biology and medicine* 2018; 102: 411-420. <https://dx.doi.org/10.1016/j.compbimed.2018.09.009>
- Patil V, Patil SN. Robust system for patient specific classification of ECG signal using PCA and Neural Network. *International Research Journal of Engineering and Technology* 2017; 4: 1400-1404.
- Raj S, Ray KC. ECG signal analysis using DCT-based DOST and PSO optimized SVM. *IEEE Transactions on instrumentation and measurement* 2017;66(3):470-478. <https://dx.doi.org/10.1109/TIM.2016.2642758>.
- D'Aloia M, Longo A, Rizzi M. Noisy ECG signal analysis for automatic peak detection. *Information* 2019;10(2):35. <https://dx.doi.org/10.3390/info10020035>.
- Wasimuddin M, Elleithy K, Abuzneid A, Faezipour M, Abuzagheh O. ECG Signal Analysis Using 2-D Image Classification with Convolutional Neural Network. In 2019 International Conference on Computational Science and Computational Intelligence (CSCI) IEEE. December; 2019, 949-954.
- Barmpoutis P, Dimitropoulos K, Apostolidis A, Grammalidis N. Multi-lead ECG signal analysis for myocardial infarction detection and localization through the mapping of Grassmannian and Euclidean features into a common Hilbert space. *Biomedical Signal Processing and Control* 2019; 52: 111-119. <https://dx.doi.org/10.1016/j.bspc.2019.04.003>
- Kaur I, Rajni R, Marwaha A. ECG Signal Analysis and Arrhythmia Detection using Wavelet Transform. *Journal of the Institution of Engineers (India): Series B* 2016; 97: 499-507. <https://dx.doi.org/10.1007/s40031-016-0247-3>
- Agarwal S, Krishnamoorthy V, Pratihir S. ECG signal analysis using wavelet coherence and s-transform for classification of cardiovascular diseases. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur; 2016, 2765-2770. <http://dx.doi.org/10.1109/ICACCI.2016.7732481>.
- Vulaj Z, Draganić A, Brajović M, Orović I. A tool for ECG signal analysis using standard and optimized Hermite transform. In 2017 6th Mediterranean Conference on Embedded Computing (MECO). IEEE. June; 2017, 1-4.
- Sadr N, de Chazal P. Non-invasive Diagnosis of Sleep Apnoea Using ECG and Respiratory Bands. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE. July; 2019, 1609-1612.
- Bazan V, Marchlinski FE. Usefulness of the 12-Lead ECG to Identify Epicardial Ventricular Substrate and Epicardial Ventricular Tachycardia Site of Origin. *Cardiac Mapping* 2019, 1028-1049.

19. Nelli F. Python data analytics: with pandas, numpy, and matplotlib. Apress; 2018.
20. Lebanon G, El-Geish M. Visualizing Data in R and Python. In: Computing with Data. Springer, Cham; 2018.
21. Hoyer S, Hamman J. Xarray: ND labeled arrays and datasets in Python. Journal of Open Research Software 2017; 5(1).
<https://dx.doi.org/10.5334/jors.148>
22. Wan X, Song H, Luo L, Li Z, Sheng G, Jiang X. Pattern recognition of partial discharge image based on one-dimensional convolutional neural network. In 2018 Condition Monitoring and Diagnosis (CMD). IEEE. September; 2018; 1-4.
23. Soto MA, Ramirez JA, Thévenaz L. Reaching millikelvin resolution in Raman distributed temperature sensing using image processing. In Sixth European Workshop on Optical Fibre Sensors International Society for Optics and Photonics. May 2016; 9916: 99162A.
24. Han Y, Wang S, Wei J, Song C. Hole filling algorithm for image array of one-dimensional integrated imaging. In Optoelectronic Imaging and Multimedia Technology VI International Society for Optics and Photonics. November 2019; 11187: 111870K.
25. Khamdamov U, Zaynidinov H. Parallel algorithms for bitmap image processing based on daubechies wavelets. In 2018 10th International Conference on Communication Software and Networks (ICCSN) IEEE. July; 2018, 537-541.

Received 2020-05-07

Accepted 2020-08-14

Available online 2020-08-19



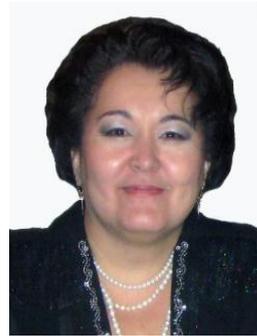
Bakhytgul R. ZHOLMAGAMBETOVA
 Doctoral student of L. Gumilyov Eurasian National University, Senior Lecturer of Karaganda Technical University (Karaganda, Kazakhstan). Research interests: computational mathematics, computer modeling, computer software and development

of hardware and software systems.



Talgat Z. MAZAKOV
 Doctor of Physics and Mathematics, professor of the Computer Science Department, Al-Farabi Kazakh National University (Almaty, Kazakhstan). Research interests: mathematical control theory, computational mathematics, computer modeling, development of hardware and software

systems and automated information analytical systems.



Sholpan A. JOMARTOVA
 Doctor of Engineering, Professor of the Artificial Intelligence and Big Data Department of al-Farabi Kazakh National University (Almaty, Kazakhstan). Research interests: studying the problems of controllability of interval mathematics systems, mathematical modeling of various technical, economic

and social systems, development of applied software.



Adilzhan Z. IZAT
 Undergraduate of the Applied Mathematics and Computer Science Department, Karaganda State University named after Academician E.A. Buketov (Karaganda, Kazakhstan). Research interests: information systems, automated production processes and management, computer

software



Olzhas D. BIBALAYEV
 Engineer of the Robotics and Intelligent Machine Scientific Research laboratory of Karaganda State University named after Academician E.A. Buketov (Karaganda, Kazakhstan). Research interests: information systems, automated production processes and management, computer modeling, computer software