МРНТИ 20.51.19, 47.13.07 **https://doi.org/10.58805/kazutb.v.1.30-1270**

МЕТОДЫ ИЗВЛЕЧЕНИЯ ЭЛЕКТРОКАРДИОГРАММ ИЗ ЭЛЕКТРОННЫХ СИГНАЛОВ И ИЗОБРАЖЕНИИ В СРЕДЕ PYTHON

Б. Рысбекқызы

Карагандинский технический университет им. Абылкаса Сагинова, Караганда,Казахстан

Корреспондент-автор: bakhytgulz@mail.ru

Качественная обработка сигнала электрокардиограммы (ЭКГ) является актуальной проблемой в современной диагностике для выявления опасных признаков сердечно-сосудистых заболеваний и аритмий у пациентов. Используемые методики и программы для анализа и классификации сигнала работают с массивами точек для математического моделирования, которые необходимо извлечь из изображения или записи электрокардиографа. Целью этой работы являлось разработать методику извлечения изображений ЭКГ сигналов в одномерный массив. Предложен алгоритм, основанный на последовательном проведении операций обработки цвета и улучшении качества изображений, наложения маски и построение одномерного массива точек с использованием инструментов и библиотек Python с открытым доступом. Результаты тестирования образцов из базы данных MIT/BIH и сравнение изображений до и после обработки показали, что точность извлечения сигнала составляет приблизительно 95%. Кроме того, представленный дизайн приложения отличается простотой и доступностью в использовании. Предложенная программа для анализа и обработки данных ЭКГ имеет большой потенциал в будущем для разработки более сложных программных приложений с целью автоматического анализа данных и определения аритмий или других патологий.

Ключевые слова: сигнал ЭКГ, MIT/BIH, Python, обработка изображений, одномерный массив, OpenCV, Matplotlib, NumPy.

ELECTRONIC SIGNALS AND IMAGES IN THE PYTHON ENVIRONMENT

B. Rysbekkyzy

Abylkas Saginov Karaganda Technical University, Karaganda, Kazakhstan,
e-mail: bakhytgulz@mail.ru

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 MIT/BIH 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.

PYTHON ОРТАСЫНДАҒЫ ЭЛЕКТРОНДЫҚ СИГНАЛДАР МЕН БЕЙНЕЛЕРДЕН ЭЛЕКТРОКАРДИОГРАММАЛАРДЫ АЛУ ӘДІСТЕРІ

Б. Рысбекқызы

Сағынов атындағы Қарағанды техникалық университеті, Қарағанды, Қазақстан,
e-mail: bakhytgulz@mail.ru

Электрокардиограмма сигналын (ЭКГ) сапалы өңдеу пациенттердегі жүрек-қан тамырлары аурулары мен аритмияның қауіпті белгілерін анықтау үшін қазіргі диагностикадағы өзекті мәселе болып табылады. Сигналды талдау және жіктеу үшін қолданылатын әдістер мен бағдарламалар электрокардиоргафтың кескінінен немесе жазбасынан алынуы керек математикалық модельдеу нүктелерінің массивтерімен жұмыс істейді. Бұл жұмыстың мақсаты ЭКГ сигналдарының кескіндерін бір өлшемді массивке шығару әдістемесін жасау болды. Ұсынылған алгоритм түстерді өңдеу операцияларын дәйекті жүргізуге және кескін сапасын жақсартуға, масканы қабаттастыруға және ашық қол жетімді Python құралдары мен кітапханаларын қолдана отырып, бір өлшемді нүктелер массивін құруға негізделген. Mit/BIH дерекқорынан алынған үлгілерді сынау нәтижелері және өңдеуге дейінгі және кейінгі кескіндерді салыстыру сигналды алу дәлдігі шамамен 95% екенін көрсетті. Сонымен қатар, ұсынылған қолданба дизайны қарапайым және қол жетімді. Ұсынылған ЭКГ деректерді талдау және өңдеу бағдарламасы болашақта деректерді автоматты түрде талдау және аритмияны немесе басқа патологияларды анықтау мақсатында күрделі бағдарламалық қосымшаларды әзірлеуге үлкен әлеуетке ие.

Түйін сөздер: ЭКГ сигналы, MIT / BIH, Python, кескінді өңдеу, бір өлшемді массив, OpenCV, Matplotlib, NumPy.

Введение. Сигнал электрокардиограммы (ЭКГ) демонстрирует множество информации о состоянии сердечно-сосудистой системы пациента. Одним из важных анализов в ЭКГ является выявление аритмий сердечных сокращений. Мониторинг и своевременное обнаружение опасных аритмий у пациента поможет предотвратить угрозы инсульта или внезапной смерти от сердечной недостаточности [1]. Во многих работах [2, 3] были проведены исследования, направленные на выявление угрожающих жизни аритмий. Данный процесс представляют собой длительную и утомительную процедуру как для пациента, так и для кардиолога. Автоматизация процесса анализа ЭКГ и компьютерная диагностика являются ключевыми и необходимыми технологиями в современной клинической кардиологии [4]. Разработка автоматизированных диагностических инструментов могут иметь решающее значение для правильной диагностики сердечно-сосудистых заболеваний. С этой целью было проведено множество исследований, в которых были использованы различные методы обработки и выделения признаков аритмий из сигналов ЭКГ. К таким методам относятся анализ во временной области [5], статистический метод [6], метод гибридных функций [7], частотный и частотно-временной анализы [8-10]. Однако из-за значительных различий во временных и морфологических характеристиках сигнала ЭКГ среди разных пациентов все еще остается нерешенная проблема точного обнаружения очагов нарушения, что мешает созданию общей классификации сигналов и требует индивидуальный подход. Поскольку ЭКГ сигнал представляет собой целый частотно-временной набор, состоящих из наборов гармонических составляющих сердечных сокращений сетевых помех, шума, мышечного тремора и т.д., например, желудочковый комплекс QRS, который представляет собой комбинацию зубцов Q, R и S, зарегистрированных во время желудочковой стимуляции, является самым большим и наиболее важным комплексом, который виден на ЭКГ. Зубцы Q и S - это нисходящие волны, которые представляют деполяризацию межжелудочковой перегородки слева направо и деполяризацию левого желудочка сердца соответственно. Зубец R - это восходящая волна, которая представляет возбуждение желудочков сердца. Поэтому распознавание необходимых сигналов после применения различных фильтров также очень сложно, особенно при слабой интенсивности сигнала. Поэтому разработка высокоточных методов анализа данных электрокардиограммы актуальна и по сей день.

В научной литературе за последнее десятилетие представлено множество работ [5, 11-13], посвященных разработкам новых методик автоматического анализа ЭКГ. Например, с помощью инструмента дискретного вейвлет-преобразования можно повысить точность и чувствительность распознавания сигналов [14]. Данный подход, который программируется в среде MATLAB для базы данных аритмии MIT/BIH, предполагает удаление артефактов и обнаружения комплекса QRS и аритмии. Как показывают авторы этой статьи, использование этого метода позволяет увеличит чувствительность (99,85%) с минимальной частотой ошибок (0,221%) при диагностировании аритмии. Аналогичный подход был использован в работе [15], где авторы провели качественный анализ данных ЭКГ с использованием сложных гауссовых вейвлетов. Согласно результатам показано, что применение S-преобразования для непрерывной вейвлет-функции позволяет получить более масштабные и точные данные спектра сигнала электрокардиограммы. В другой работе [16] предложено преобразования Эрмита в качестве инструмента для анализа сигналов ЭКГ. Разработанный в MATLAB метод позволяет анализировать и классифицировать части сигналов (комплексы QRS) путем сравнения с базисными формами, построенными с помощью функций Эрмита. Предложенный подход так же дает внушительные результаты и учитывает высокие показатели точности. Садр и Де Чазаль [17] предложили метод обработки с точностью до 82%, основанный на морфологии ЭКГ и интервалах RR.

Из приведенных выше методик следует вывод, что для более точного определения сигналов аритмии для обработки данных необходимо иметь высоко качественные сигналы ЭКГ, которые можно получить непосредственно при измерении на пациенте. Однако к сожалению высоко точная и чувствительная аппаратура не всегда доступна или отсутствует на местах из-за высокой стоимости и необходимого обслуживания [18].

Кроме того, не все программы могут распознавать форматы файлов jpg и tiff, поскольку обработка информации осуществляется с помощью массива точек, с которыми можно работать в вычислительной среде в специальных программных приложениях. В этой статье делается попытка исправить дисбаланс в предыдущих исследованиях путем разработки программы для получения чистого сигнала ЭКГ с высоким качеством разрешения в виде одномерного массива со сканированной электрокардиограммой или записью электронного файла. В этой статье мы предлагаем новый метод обработки изображений с помощью пакетных приложений на языке программирования Python. Новизна нашего подхода состоит в том, что он основан на удобном и эффективном программном пакете, который предназначен для быстрого преобразования данных ЭКГ в формат, пригодный для декодирования и анализа. Такой подход позволяет поставить точный диагноз. С научной и прикладной точек зрения результаты могут быть полезны при разработке медицинских и других компьютерных программ, и мобильных приложений.

Материалы и методы. Для достижения предполагаемого результата необходимо выбрать подходящий инструмент, пакеты и язык программирования. В качестве языка программирования был выбран Python, который является одним из мощных инструментов для создания программ и проектов. Данный язык отличается простотой, доступностью и адаптирован под современные операционные системы и приложения. Python обладает объектно-ориентированными, структурными, высокоуровневыми, функциональными навыками. В этом исследовании мы используем версию Python 4.2.0, которая предоставляет средства программирования на достаточном уровне, включая множество встроенных функций. Кроме того, для Python существует множество внешних библиотек с открытым исходным кодом, которые доступны в Интернете. В нашем исследовании для эффективной обработки изображения в скрипте будут использоваться библиотеки OpenCV, Matplotlib, NumPy [19,20].

x_array представляет собой пакет Python с открытым исходным кодом, который предостав-

ляет инструментарий и структуры данных для N-мерных помеченных массивов [21]. Данный инструмент позволяет объединить интерфейс прикладного программирования с общей моделью данных, включает индексирование и арифметику на основе меток, совместим с NumPy, Matplotlib.

Исходные данные. Изображения ЭКГ сигналов (рис.1) в виде файлов двух форматов *.dat и *.hea были взяты из общедоступной базы данных MIT/BIH из двух источников (https://physionet.org/ and https://ecgpedia.org/).

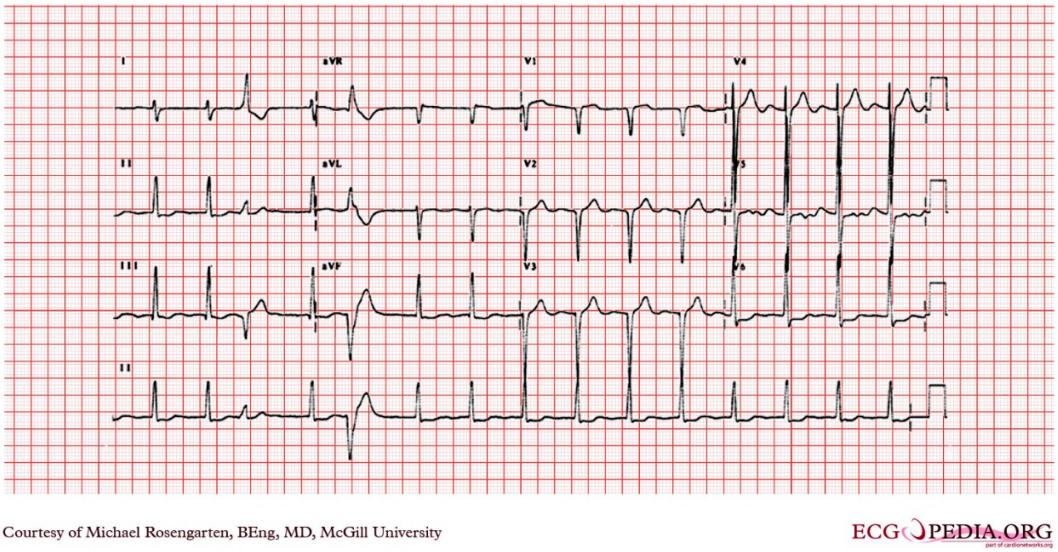


Рис.1 - Рисунок со сканом ЭКГ сигнала

Для дальнейшей работы из каждого изображения вырезалось одна полоса электрокардиограммы (рис.2). Для работы в приложении были отобраны файлы содержащие изображения высокого разрешения. В итоге из 500 файлов было выбрано 100 изображений и 100 записей (100 файлов *.dat и 100 файлов *.hea), которые соответствовали необходимым требованиям. Для построения интерфейса использовалась библиотека PyQt5, т.к. интерфейс этого приложения очень прост и понятен для обычного пользователя.



Рис.2 - Однополосное изображение сигнала ЭКГ

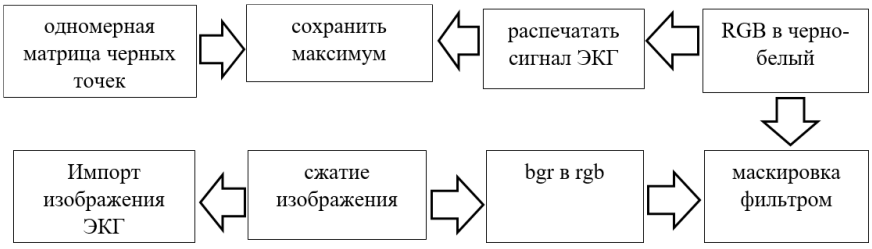


Рис.3 - Блок-схема алгоритма обработки изображений

Построение алгоритма обработки изображений в Python. В зависимости от загруженного типа данных программа выбирает соответствующий алгоритм обработки данных. Рассмотрим их отдельно. В начале рассмотрим алгоритм обработки загруженного изображения.

Для начала импортируем библиотеки:

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

Считываем файл изображения через путь к файлу (FILE), который был получен из программы загрузчика:

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

В переменной bgr_img_array теперь хранится матрица точек изображения. Каждая точка теперь содержит информацию о своем цвете в цветовом формате BGR. Так же, после преобразования рисунка в матрицу, мы можем наблюдать одиночные точки, которые в дальнейшем будут проявляться в виде шумов. Чтобы исключить одиночные точки, которые появились из-за шумов можно применить сжатие изображения для фильтрации шумов. Стоит обратить внимание, что качество загружаемого изображения должно быть выше 100 пикселей, которое прописывается следующим способом:

```
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))
```

Выполнив выше указанные операции, сделаем сжатие изображения для исключения одиночных шумов. Теперь сжатая матрица, к которой был применен фильтр шумов, принадлежит к переменной resized_bgr_img_array. Поскольку цвет каждой точки хранится пока в формате BGR, то необходимо изменить данные цвета на формат RGB, который записывается следующим образом:

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

Теперь данные цветов точек хранятся в формате RGB.

Далее следует исключить лишние цвета из матрицы представленной электрокардиограммы (рис.2). Поскольку сейчас данные цветов хранятся в формате [R, G, B], мы можем через наложение маски из библиотеки OpenCV избавиться от цветов, которые отличаются от черного. Для этого определим границы цветов искомых точек переменными lower (наименьшее значение) и upper (наибольшее значение). Для удобства при дальнейшем написании кода матрицу, в которой хранятся данные изображения, назовем image. Теперь создадим маску, через которую будем исключать ненужные цвета в изображении и назовем ее mask. Переменной output зададим матрицу, на которую уже наложена маска, а пустые точки заполним белым цветом. Код данной программы имеет вид:

```
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)
```

Теперь для окончательного выделения черного цвета и их оттенков, преобразуем изображение в черно-белый формат из формата RGB. Для этого применим следующую функцию:

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

Значение переменной out_gray представляет собой матрицу с оставшимися точками, у которых значение цвета от 0 до 150, т.е. от черного к серому. Так же в матрице существуют еще белые точки. Чтобы составить одномерную матрицу, необходимо проверить значения сверху вниз по каждой вертикальной линии. Это выглядит следующим образом. Если ширина изображения X, а высота Y, то мы должны взять линию X и проверять черные точки на высоте Y. Для этого заменим значения X на Y в переменной out_gray, т.к. в ней горизонтальные линии представляют собой Y, а высотой считает X:

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

После чего мы можем набрать матрицу. Проходим по каждой линии X и Y, сравнивая каждую точку с черным цветом. Полученный результат запишем в ранее инициализированную переменную:

```
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)
```

Значением переменной ecg_points_all_blacks является двумерной матрицей, в которой первым уровнем является линия, где были черные точки. Данная линия содержит координаты оси Y для каждой черной точки, которые имеют следующий вид:

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

где на первой линии расположены две черные точки со значениями координат 12 и 26; на второй линии расположены две черные точки с координатами 12 и 34. В линиях встречается ряд черных точек вроде [15-17]. Приравнивать значение в одномерном массиве к арифметической средней приведет в будущем к проблеме вычисления экстремумов: пиков R, Q и S. Поэтому лучшим решением будет сохранить общую среднюю точку путем суммирования координат по оси Y всех черных точек в данной линии, которое потом делится на их количество.

Применяя проверку к каждой точке по принадлежности к нижней или верней стороне от средней линии, в зависимости от результата сохраня-

ем соответсвенно максимальные или минимальные точки в рядах как [15-17]. Таким образом мы убираем множество точек, которые стоят друг за другом. Так же сохраняем экстремумы, показывая только максимумы (сохранение минимумов осуществляется идентичным кодом):

```
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)
```

Конкатенируем матрицы с точками, убирая идентичные точки, и выводим результат следующим образом:

```
print(ecg_points_blacks_result)
print (”Длина сигнала: ” + len(ecg_points_blacks_result) + ” точек”)
```

В окне консоли увидим результат, который приведен в качестве примера:

```
[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]
```

Длина сигнала: 748 точек.

Извлечение данных из файлов с расширениями .dat и. hea. В данном контексте намного проще, поскольку в современных аппаратах электронные записи ведутся с огромной частотой, и сигналы сразу записываются в виде одномерного массива. Проблема состоит в том, как эти данные правильно прочесть. Эти файлы хранят следующую информацию: частота дискретизации, длина сигнала, дата записи, время записи, часовой пояс в которой была сделана запись, единица измерения, название сигнала, комментарии и множество других значении. Чтобы прочитать электронные сигналы и вывести из них точки нам поможет библиотека wfdb, которая предназначена для работы с файловыми данными из базы MIT и работа с файлами *.dat и *.hea. Выбрав файлы в программе и нажав кнопку «Загрузить», мы идентифицируем новое имя для двух файлов. Имена файлов должны быть одинаковыми.

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

где PATH-путь к файлам и ранее идентифицированное название файлов, points-одномерный массив со всеми точками.

Результаты и обсуждение. Интерфейс разработанного приложения для преобразования сигнала показан на рис.4. Как видно, в окне приложения предлагается выбрать тип данных, которые представляются в виде изображения или записи. В зависимости от выбранного типа программа запросит необходимые файлы для чтения. Как видно из рисунка при выборе записи из MIT-BIH программа предлагает выбрать два соответствующих файла, которые хранятся в разных форматах. В ином случае нужно выбрать всего один файл скана либо фотографии. Интерфейс разработанного приложения выглядит довольно просто и понятно для любого пользователя, что не требует дополнительных инструкций и введения других операций для загрузки файла.

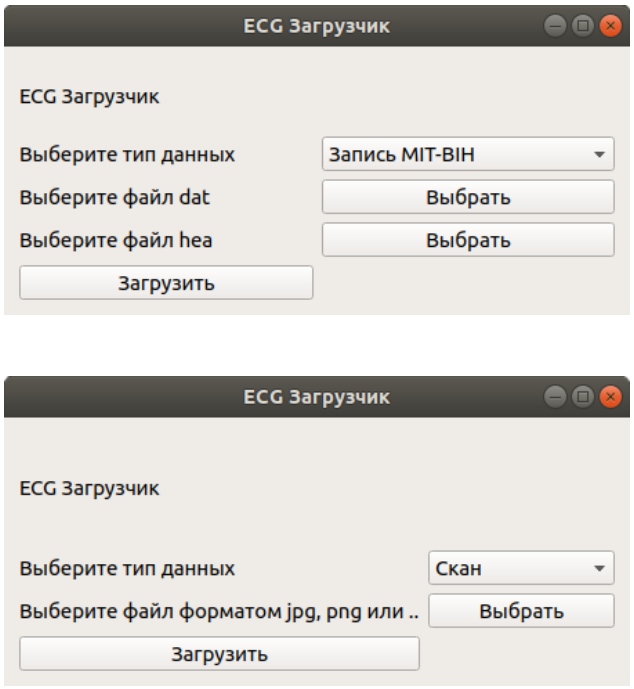


Рис.4 - Интерфейс приложения для чтения ЭКГ сигналов с изображений и электронных записей

После выбора файла в окне загрузчика, на экране открывается окно с графиком изображения в виде одномерного массива (рис.5), который можно использовать для приложения, анализирующего предоставленный полный сигнал ЭКГ.

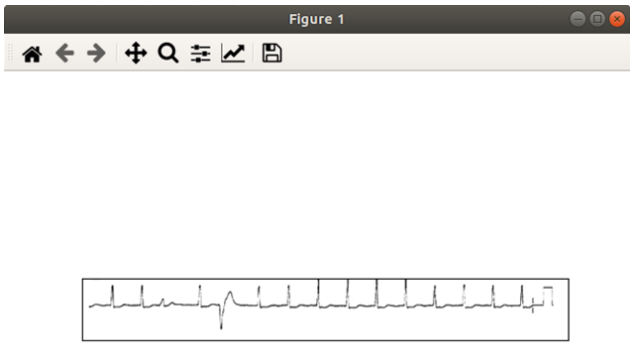


Рис.5 - График одномерного массива построенный с помощью библиотеки matplotlib

Как видно, предложенный алгоритм обработки сканов и других изображений электрокардиограмм демонстрирует хорошие результаты. Используемый подход наложения маски с исключением ненужных цветов, как оказалось, является весьма эффективным. Как видно на рис.6 сравнение результатов до и после обработки скана разработанная программа довольно точно идентифицирует положение точек, соответствующие ЭКГ сигналу. Кроме того, сохраняется масштаб и малейшие отклонения, которые могут скрывать в себе очень важную информацию при расшифровке.



Рис.6 - До и после наложения маски с фильтром выделения черного цвета

Аналогичные результаты были получены после обработки записей MIT / BIH. Сравнительный анализ 100 образцов изображений до и после обработки показывают (на рис.6) полное совпадение сигналов ЭКГ. Такой высокий показатель связан также с предварительной подготовкой изображения и правильной последовательностью распознавания положения точек, которые были предложены при разработке алгоритмов распознавания сигнала. Таким образом, данный подход повышает чувствительность и точность методики обработки ЭКГ изображений, а разработанный в Python проект представляет собой быстрое и простое в пользовании приложение, что является очень необходимым при анализе сигнала и его классификации. Например, для классификации сигналов ЭКГ используют такие признаки как начало и смещение волн, QRS и период, которые выявляются при анализе во временной области сигнала [1].

Для извлечения одномерного массива из изображения применялась множество других подходов. В работе [22] предложен способ распознавания формы сигнала во временной области на основе одномерной сверточной нейронной сети. В данном подходе предложено построение нейронной сети для прямого распознавания образов в форме волны. Несколько другой подход был применен в [23] для обработки изображений рамановских оптоволоконных датчиков. Суть предложенной авторами методики состояла в объединении одномерных рамановских сигналов в двумерный массив с целью подавления шума и улучшении самого сигнала при обработке. Приведенные результаты работ, как и в этой статье, демонстрируют высокую эффективность используемых методик обработки изображений с целью повышения качества сигнала. Однако предложенные методики учитывают только специфику того сигнала, под который они были разработаны, и не являются универсальными в отличие от нашего подхода. Разработанный в этой работе алгоритм может быть усовершенствован и адаптирован для извлечения не только ЭКГ сигналов, но и для других измерений, которые используют схожую методику записи данных.

Расширенный фильтр Калмана в сочетании с Дискретно-Вейвлет преобразованием, реализуется для улучшения качества сигналов ЭКГ, доступных на выходе РФК. Производительность алгоритма оценивается двумя параметрами: отношения сигнала к шуму (ОСШ (дБ) – и средне-

квадратичной ошибки (СКО). Результаты эффективности при использование математической модели где погрешности в Расширенном фильтре Калмана стали на много меньше, в сравнение с Вейвлет преобразованием. Фильтрации сведены в таблицу 1 для сравнительного исследования с помощью РФК удалось подавить шумы.

Таблица 1 - Результат эффективности при использовании математической модели в РФК и ВП

Номер ЭКГ	ВП		РФК	
	ОСШ (дБ)	СКО	ОСШ (дБ)	СКО
101	6,7326	2,0e-03	3,8286	1,73e-04
105	3,8545	3,12e-04	2,5305	1,28e-07
106	8,4883	6,0e-03	4,1280	1,04e-03
108	9,0482	1,7e-03	5,2344	1,5e-05
111	5,3831	5,01e-04	2,9266	2,1e-05
115	10,8044	9,34e-03	9,7907	6,46e-04
117	6,5982	5,76e-03	4,4769	2,31e-05
118	9,5621	6,30e-04	7,3690	2,6e-05
119	8,9988	2,06e-02	6,5206	1,4e-04
120	7,3859	6,89e-04	3,4411	1,57e-06
121	5,7911	4,7e-03	3,3853	3,03e-04
144	8,7677	8,1e-03	7,5622	6,81e-05
145	9,9772	2,6e-03	6,3583	1,7e-04
156	5,7318	4,0e-03	3,6376	3,5e-05
158	10,1275	9,13e-03	5,7961	4,3e-05
160	6,2156	6,13e-04	5,9036	1,3e-06
161	8,0652	2,4e-03	7,7988	1,06e-04
165	7,5938	6,89e-04	6,3384	1,39e-04
166	5,7915	4,7e-03	5,3323	3,01e-05
167	8,7677	7,0e-03	8,2256	5,81e-05
171	9,7297	2,7e-03	7,8335	2,5e-06
172	4,7318	4,0e-03	3,7663	3,3e-06
175	9,7215	4,41e-03	4,7581	2,89e-07
183	5,7629	5,30e-04	5,3690	2,8e-05
188	7,9624	6,57e-04	7,9075	3,3e-05

Аналогичный подход был использован в [24], где авторы разработали алгоритм восстановления изображений в массиве с помощью одномерной интегрированной системы обработки изоб-

ражений. На основе теории геометрической оптики и теории изучения глубины изображений был разработанный алгоритм заполнение недостающих точек происходит по имитационной карте глубины изображений путем сравнения и оценки качества изображения. Более усовершенствованную технологию обработки растровых изображений представили Khamdamov и др. [25]. Новизной их подхода являлось использование вейвлет-функция Daubechies для фильтрации, сжатия и сглаживания двумерных сигналов. Алгоритм вычислительных процессов представлен в среде OpenMP на языке C/C++, который является более сложным и сопоставим не для всех приложений по сравнению с Python и его доступной библиотекой, и пакетом инструментов с открытым исходным кодом.

Выводы. Согласно представленной методики и результатам, можно сделать следующий вывод, что в настоящей работе разработана новая методика обработки изображений и записей ЭКГ сигналов в виде одномерного массива. Использование последовательный операций выбора формата обработки цвета, улучшения качества изображений, наложения маски и построение одномерного массива точек позволяет распознать сигнал ЭКГ с высокой точностью. Сравнительный анализ изображений до и после обработки показывает хорошее соответствие сигналов ЭКГ, что свидетельствует о высокой чувствительности предложенного метода. Кроме того, использование пакетов и библиотек Python с открытым доступом позволяет разработать простой и доступный дизайн приложения для быстрой обработки. Цель исследования была достигнута путем разработки приложения Python для преобразования фото или отсканированных изображений в одномерный массив точек, который затем может быть обработан с помощью программы декодирования. Данная методика и предложенная программа для анализа и обработки данных ЭКГ имеет большой потенциал в будущем для разработки более сложных приложений для автоматического анализа данных и более быстрого определения аритмий и других патологий у пациентов.

References

1. Bhirud, B., & Pachghare, V. K. Arrhythmia Detection Using ECG Signal: A Survey //Proceeding of International Conference on Computational Science and Applications. -2020.-P.329-341. DOI 10.1007/978-981-15-0790-8_32.

2. Trayanova, N. A., Boyle, P. M., & Nikolov, P. P. Personalized imaging and modeling strategies for arrhythmia prevention and therapy //Current opinion in biomedical engineering -2018. Vol.5.-P.21-28. DOI 10.1016/j.cobme.2017.11.007

3. Chatha, S. R., Chahal, C. A. A., Westwood, M., & Khanji, M. Arrhythmia and sarcoidosis: the role of myocardial tissue characterization to guide diagnosis and management //European Heart Journal-Cardiovascular Imaging – 2020.-Vol.21(3): 344. DOI 10.1093/ehjci/jez229.

4. Abdelazim I.A., Bekmukhambetov Y., Aringazina R., Shikanova S., Amer O.O., Zhurabekova G., Otessin M.A., Astrakhanov A.R. The outcome of hypertensive disorders with pregnancy. //Family Medicine and Primary Care. -2020.9(3). -P.1678-1683. DOI 10.4103/jfmpc.jfmpc_1054_19

5. Raj S., Maurya K., & Ray K. C. A knowledge-based real time embedded platform for arrhythmia beat classification//Biomedical Engineering Letters. -2015. Vol.5.-P.271-280. DOI 10.1007/s13534-015-0196-9.

6. Ye C., Kumar B. V., & Coimbra M. T. Heartbeat classification using morphological and dynamic features of ECG signals //IEEE Transactions on Biomedical Engineering. -2012. Vol.59(10). -P.2930-2941. DOI 10.1109/TBME.2012.2213253.

7. Zhang, Z., Dong, J., Luo, X., Choi, K. S., & Wu, X. Heartbeat classification using disease-specific

feature selection//Computers in biology and medicine//Computers in Biology and medicine. -2014. Vol.46. P.79-89. DOI 10.1016/j.compbiomed.2013.11.019.

8. Yıldırım Ö., Pławiak P., Tan R. S., & Acharya U. R. Arrhythmia detection using deep convolutional neural network with long duration ECG signals//Computers in biology and medicine -2018. Vol.102.-P.411-420. DOI 10.1016/j.compbiomed.2018.09.009.

9. Patil V., & Patil S. N. Robust system for patient specific classification of ECG signal using PCA and Neural Network// IRJET. -2015.-Vol.2(9). e-ISSN: 2395-0056.

10. Raj, S., & Ray, K. C. ECG signal analysis using DCT-based DOST and PSO optimized SVM //IEEE Transactions on instrumentation and measurement. -2017.66(3), P.470-478. DOI 10.1109/TIM.2016.2642758.

11. D'Aloia M., Longo A., & Rizzi M. Noisy ECG signal analysis for automatic peak detection //Information. - 2019, 10(2), P.35. DOI 10.3390/info10020035.

12. Wasimuddin M., Elleithy K., Abuzneid A., Faezipour M., & Abuzaghleh O. ECG Signal Analysis Using 2-D Image Classification with Convolutional Neural Network//2019 International Conference on Computational Science and Computational Intelligence (CSCI). -2019.-P.949-954.

13. 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.-Vol.52.-P.111-119. DOI 10.1016/j.bspc.2019.04.003.

14. Kaur I., Rajni R. & Marwaha A. ECG Signal Analysis and Arrhythmia Detection using Wavelet Transform//Institute of Chemists (India).-2016.-P.1-11. DOI 10.1007/s40031-016-0247-3.

15. S. Agarwal, V. Krishnamoorthy and S. Pratiher, "ECG signal analysis using wavelet coherence and s-transform for classification of cardiovascular diseases//International Conference on Advances in Computing. - 2016. DOI 10.1109/ICACCI.2016.7732481.

16. 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 -2017. DOI 10.1109/MECO.2017.7977145.

17. Sadr. N., 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.-2019.DOI 10.1109/EMBC.2019.8857414.

18. Bazan V., Marchlinski.F. Usefulness of the 12-Lead ECG to Identify Epicardial Ventricular Substrate and Epicardial Ventricular Tachycardia Site of Origin//Cardiac Mapping. -2019. DOI 10.1002/9781119152637.ch81.

19. Nelli. F. Python data analytics: with pandas, numpy, and matplotlib//Apress. -2018. DOI 10.1007/978-1-4842-3913-1.

20. Lebanon G., El-Geish M. Visualizing Data in R and Python: An introduction to the Data Industry//In: Computing with Data. -2018.-P.277-324. DOI 10.1007/978-3-319-98149-9_8.

21. Hoyer S., & Hamman J. xarray: ND labeled arrays and datasets in Python //Journal of Open Research Software. -2017. Vol.5(10). -P.1-6. DOI 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//IEEE. -2018. DOI 10.1109/CMD.2018.8535761.

23. Soto M. A., Ramírez J. A., & Thévenaz L. Reaching millikelvin resolution in Raman distributed

temperature sensing using image processing//SPIE. -2016.-Vol.9916. DOI 10.1117/12.2236934.

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. -2019. Vol.11187:111870K. DOI 10.1117/12.2538177.

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)). -2018. DOI 10.1109/ICCSN.2018.8488270.

Сведение об авторе

Б. Рысбекқызы - PhD, и.о.доцента Карагандинского технического университета им. Абылкаса Сагинова, Караганда, Казахстан, e-mail: bakhytgulz@mail.ru.

Information about the author

B. Rysbekkyzy - PhD, Acting Associate Professor, Abylkas Saginov Karaganda Technical University, Karaganda, Kazakhstan, e-mail: bakhytgulz@mail.ru.