

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВПО «МГИУ»)
КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

КУРСОВАЯ РАБОТА

по дисциплине «Проектирование и разработка корпоративных
информационных систем»
на тему «Визуализация результатов поверхностного
ЭКГ-Картирования»

Группа

111131

Студент

И.С. Сергиенко

Руководитель работы
доцент

Н.В. Лукьянова

Москва 2014

Аннотация

Курсовая работа посвящена визуализации результатов поверхностного ЭКГ-картирования. Программе на вход будет подаваться файл с результатами обследования пациента. Полученные данные будут представляться в виде цветовой карты.

Содержание

1.	Введение	4
2.	Постановка задачи	6
3.	Анализ подобных систем	6
4.	Выбор архитектуры программного обеспечения	6
5.	Задачи	6
6.	Объекты автоматизации	7
7.	Проектирование	8
8.	Интерфейс	11
9.	Заключение	14

1. Введение

В современное время в кардиологии начинается активное использование поверхностного ЭКГ-картирования. Данное понятие представляет собой одномоментную регистрацию множественных отведений ЭКГ со всей поверхности грудной клетки. Это исследование, основанное на распределении электрических потенциалов сердца на поверхность грудной клетки. Данная методика направлена на изучение нормальной и патологической электрофизиологии сердца. При использовании привычной всем стандартной ЭКГ, все до этого перенесенные инфаркты миокарда, эктопические комплексы, синдромы предвозбуждения желудочков создают условия, которые препятствуют качественной классификации и диагностики электрофизиологических параметров. В отличие от общепринятых методик электрокардиографии, в которых измеряются и анализируются параметры электрического поля сердца в небольшом числе точек поверхности торса, в методах картирования применяются множественные датчики, а сигналы от всех этих датчиков регистрируются синхронно, что позволяет изучать характеристики и параметров ЭКГ, даже при документированных сердечных аномалиях. Получаемые результаты в определенный промежуток времени будут представляться в виде цветовой диаграммы.

По характеристикам поверхностного ЭКГ-картирования можно оценить следующие параметры: характер распределения потенциалов, таких как дипольность и мультипольность, расположение экстремумов потенциалов, величина экстремумов, наличие нетипичных положительных и отрицательных зон, направление и скорость движения фронтов возбуждения.

Примеры изопотенциальных карт приведены на рис. 1. Схематическая развертка грудной клетки представлена как цилиндрическая поверхность, разомкнутая по правой задне-подмышечной линии. Для построения карты (в указанный момент времени сердечного цикла) точки, имеющие равный потенциал, соединяются линиями. Точки, имеющие положительный потенциал, создают позитивную зону (незакрашенная область) с максимумом потенциала («+», числовое значение приведено в мкВ), а точки, имеющие отрицательный потенциал, создают негативную зону (закрашенная область) с минимумом потенциала («-», числовое значение приведено в мкВ).

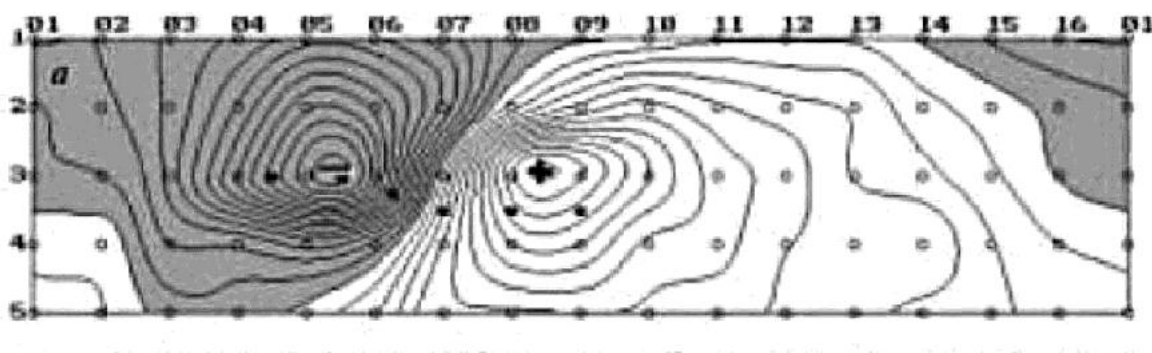


Рис. 1. Типичное распределение потенциалов на изопотенциальных картах процесса деполаризации и реполяризации желудочков

Имея такую модель, медицинские работки смогут более наглядно получать данные и сопоставлять их с реальными физическими процессами. Для обеспечения их необходимой поддержкой будет создан интегрированный аппаратно-программный комплекс.

Программный комплекс будет представлять собой интерфейс, состоящий из окошка, в котором данные исследования будут представлены в виде цветовой диаграммы. В отличие от стандартного ЭКГ, где используется 13 датчиков, в поверхностном ЭКГ-картировании используется 80 датчиков (5 рядов по 16). Каждый из 80 значений будет являться числом, которое будет иметь определенный цвет в шкале.

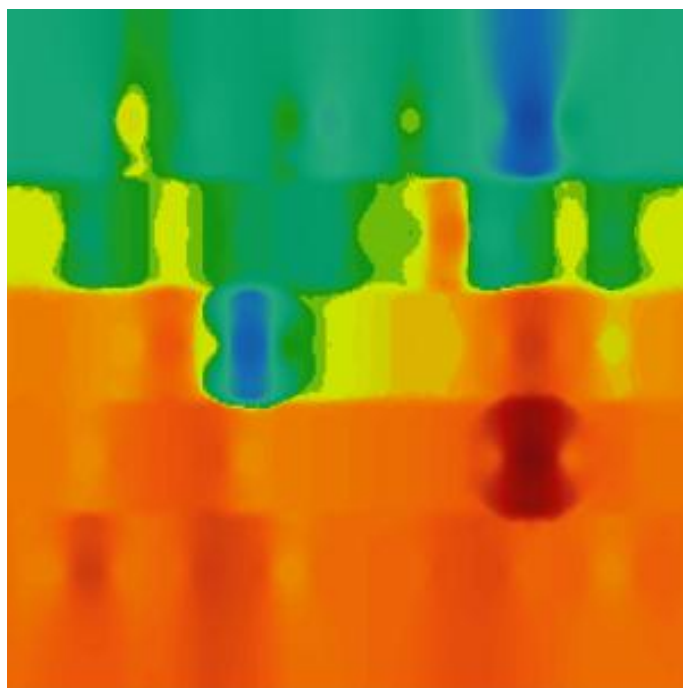


Рис. 2. Пример цветовой диаграммы

Результат обследования должен выглядеть как на рис. 2. Так же в окошке приложения будет расшифровка цветовой шкалы и бегунок, который позволит просмотреть результат в определенный промежуток времени.

Второй частью выполнения работы будет разработка программного комплекса для анализа данных поверхностного ЭКГ-картирования при помощи многослойного персептрона. Данный комплекс будет создаваться с целью выявления у людей, страдающих ожирением, скрытой эшемии на основе анализа данных, взятых при помощи поверхностного ЭКГ-картирования на аппарате Astrocord.

2. Постановка задачи

Разработать программный комплекс для анализа данных, полученных при помощи поверхностного ЭКГ-картирования. Далее задача подразделяется на две подзадачи:

- разработать программу, которая будет представлять собой визуализацию результатов поверхностного ЭКГ-картирования. Программе на вход будет подаваться файл с числами (результатами обследования), и каждое число будет представлять определенный цвет на карте. Для более красивого представления карты цветов будет использоваться интерполяция.
- разработать систему для выявления у людей, страдающих ожирением, скрытой эшемии. Далее будет использоваться сокращенное название системы: «ПКАДАД». ПКАДАД будет применяться к пациентам, страдающим ожирением различных степеней. В начале испытания пациент проходит тест на аппарате *Astrocard*. Затем следует считывание данных на различных стадиях: пик, покой на первой, третьей, пятой и седьмой минутах. После обработки всех данных, программа должна дать ответ на вопрос: «Есть ли скрытая эшемия или нет?».

Система разрабатывается в компьютерной лаборатории МГИУ кафедры №11 для института питания РАМН в рамках договора о дружбе.

3. Анализ подобных систем

При анализе подобных систем было установлено, что аналогичной системы для конкретной задачи не существует. Данный программный комплекс будет разрабатываться впервые.

4. Выбор архитектуры программного обеспечения

Система будет разрабатываться для операционной системы *Windows 8*, на языке программирования *c++*, при помощи библиотеки *QT 5.0.2*, для интерполяции будет использоваться библиотека *QOpenGL*. Выбор языка программирования очевиден, т.к. требуется быстро обрабатывать файлы больших размеров. Библиотека *QT* позволяет существенно упростить разработку интерфейса приложения.

5. Задачи

- Сжатие результатов обследования при помощи анализа главных компонент. Одна из самых масштабных задач, т.к. файлы экспорта имеют достаточно большой размер, превышающий 300 мегабайт. Предварительное сжатие файла, позволит сократить его объем до 15 мегабайт. Данная процедура, в дальнейшем, существенно ускорит работу программы. Анализ главных компонент – один из основных способов, позволяющих сократить размерность данных, при котором у нас происходит минимальная потеря количества информации.
- Разработка интерфейса приложения для первой подзадачи, а именно представление результатов в виде цветовой карты.

- Разработка интерфейса для ввода данных и формирования групп пациентов. Интерфейс будет разрабатываться при помощи библиотеки QT 5.0.2.
- Обучение нейронной сети (многослойный персептрон).
- Тестирование нейронной сети и анализ полученных данных. Изучение структуры многослойного персептрона в зависимости от результатов (настройка нейронной сети).

6. Объекты автоматизации

Структурное подразделение	Наименование процесса	Автоматизация
Врач — пользователь	Классификация групп при помощи обученных нейронных сетей.	Полная
Врач — оператор	Формирование групп для обучения нейронной сети, сам процесс обучения.	Частичная

Рис. 3. Объекты автоматизации

1. Классификация групп при помощи обученных нейросетей. Существует 6 групп пациентов: 1у, 2у, 3у, 1п, 2п, 3п. Первая цифра в названии группы — степень ожирения пациента. Буква «у» - есть скрытая ишемия, «п» - скрытая ишемия отсутствует. Соответственно, после прохождения пациентом теста на аппарате «Astrocard», врач — пользователь, будет добавлять полученные данные в систему и система будет давать предварительную классификацию по одной из шести групп. Процесс будет полностью автоматизирован.
2. Формирование групп для обучения нейронной сети. Ввод данных с уже утвержденными данными, формирует группы, вносит информацию.
3. Процесс обучения нейронной сети. Иницирует обучение нейронной сети. При получении новых утвержденных данных может изменять группы подтвержденных пациентов и переучивать нейронную сеть. Последние два процесса имеют частичную автоматизацию.

7. Проектирование

На данный момент спроектирован основной интерфейс приложения, при помощи библиотеки QT 5.0.2, а так же разработан алгоритм обработки файла с данными результатов поверхностного ЭКГ-картирования. Файл экспорта представляет собой набор символов, общим объемом более 50 Мб. Для более наглядно и удобного использования этих данных, был разработан алгоритм на языке программирования с++.

Ниже приведен код алгоритма экспортного файла.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include "polyfmt.h"
#define BLOCK_DELIMETER "=====\n"

int main(void){
int fileWithDateDescriptor = open("./export.240", O_RDONLY);
if(fileWithDateDescriptor == -1){
printf("Can't open file");
exit(0);
}
ExportFileHeader* fileHeader = (ExportFileHeader*)malloc(sizeof(ExportFileHeader));
read(fileWithDateDescriptor, fileHeader, 256);
printf("Start ExportFileHeader\n");
printf(BLOCK_DELIMETER);
printf("Header text: ");
printf("%s\n", fileHeader->HeaderText);
printf("HeaderSize: ");
printf("%u\n", fileHeader->HeaderSize);
printf("RecordUnitSize: ");
printf("%u\n", fileHeader->RecordUnitSize);
printf("SamplingRate: ");
printf("%u\n", fileHeader->SamplingRate);
printf("WeightOfLSB: ");
printf("%u\n", fileHeader->WeightOfLSB);
printf("\nSYSTEMTIME\n\n");
printf("wYear: %hu\n", fileHeader->RecordTime.wYear);
printf("wMonth: %hu\n", fileHeader->RecordTime.wMonth);
printf("wDayOfWeek: %hu\n", fileHeader->RecordTime.wDayOfWeek);
printf("wDay: %hu\n", fileHeader->RecordTime.wDay);
printf("wHour: %hu\n", fileHeader->RecordTime.wHour);
printf("wMinute: %hu\n", fileHeader->RecordTime.wMinute);
printf("wSecond: %hu\n", fileHeader->RecordTime.wSecond);
printf("wMilliseconds: %hu\n", fileHeader->RecordTime.wMilliseconds);
```



```

    printf("\nEND_SYSTEMTIME\n\n");
    printf("SecondFromStart: ");
    printf("%u\n", fileHeader->SecondFromStart);
    printf("Unused2: ");
    printf("%u\n", fileHeader->Unused2);
    printf("Patient: ");
    printf("%s\n", fileHeader->Patient);
    printf("Unused: ");
    printf("%u\n", fileHeader->Unused);
    printf("StationSerialNumber: ");
    printf("%s\n", fileHeader->StationSerialNumber);
printf(BLOCK_DELIMETER);
int bytes_read = 0;
int currentBlock = 1;
int i = 0 ;
int j = 0;
ExportRecordUnit* record = NULL;
do{
    if (j==256) j = 0;
    record = (ExportRecordUnit*)malloc(sizeof(ExportRecordUnit));
    bytes_read = read(fileWithDateDescriptor, record, 512);
    if(bytes_read == 512){
        if (j < 1000){
            printf("ExportRecordUnit");
printf(" %d\n", currentBlock);
printf(BLOCK_DELIMETER);
printf("StandardECG: ");
printf("%hu\n", *record->StandardECG);
printf("unused: ");
printf("%hu\n", *record->unused);
printf("ExtroECG: ");
            printf("%hd\n", record->StandardECG[3]);
            //printf(BLOCK_DELIMETER);
        }
        free(record);
        currentBlock += 1;
    }
    j++;
} while (bytes_read == 512 );
printf("%d", j);
close(fileWithDateDescriptor);
free(fileHeader);
return 0;
}

```

Также, на данный момент спроектирован алгоритм, который позволяет получить большее число точек для представления данных в цвете.

```
#include <iostream>
#include <memory>
#include <thread>
#include <vector>
double ** convert_from_5_17_to_9_33(double ** array, int h, int w){
    double **res;
    int rows = w + w - 1;
    int cols = h + h - 1;
    res = (double **)malloc(rows * sizeof(double *));
    for (int i = 0; i < cols; i++)
        res[i] = (double *)malloc(rows * sizeof(double));
    int arr_i = 0, arr_j = 0;
    for(int i = 0; i < cols; i++){
        for(int j = 0; j < rows; j++){
            if(i%2 == 0){
                if(j%2 == 0){
                    res[i][j] = array[arr_i][arr_j];
                    arr_j++;
                }
                else continue;
            }
            i++;
            arr_i++;
            arr_j = 0;
        }
    }
    for(int i = 0; i < cols; i++){
        for(int j = 0; j < rows; j++){
            if(i%2 == 0 && j%2 == 1){
                res[i][j] = (res[i][j-1] + res[i][j+1])/2;
            }
            else if(i%2 == 1 && j % 2 != 1){
                res[i][j] = (res[i-1][j] + res[i+1][j])/2;
            }
            else if(i%2 == 1 && j % 2 == 1){
                res[i][j] = (res[i-1][j - 1] + res[i - 1][j + 1] +
                    res[i+1][j-1] + res[i+1][j+1])/4;
            }
        }
    }
    return res;
}
```

8. Интерфейс

На данный момент разработан некоторый интерфейс приложения. Разработано стартовое окно. Окно для ввода данных в форму, начальный вариант цветовой диаграммы. Оно содержит две кнопки. Каждая кнопка для своей подзадачи.

Приведем пример стартового окна.

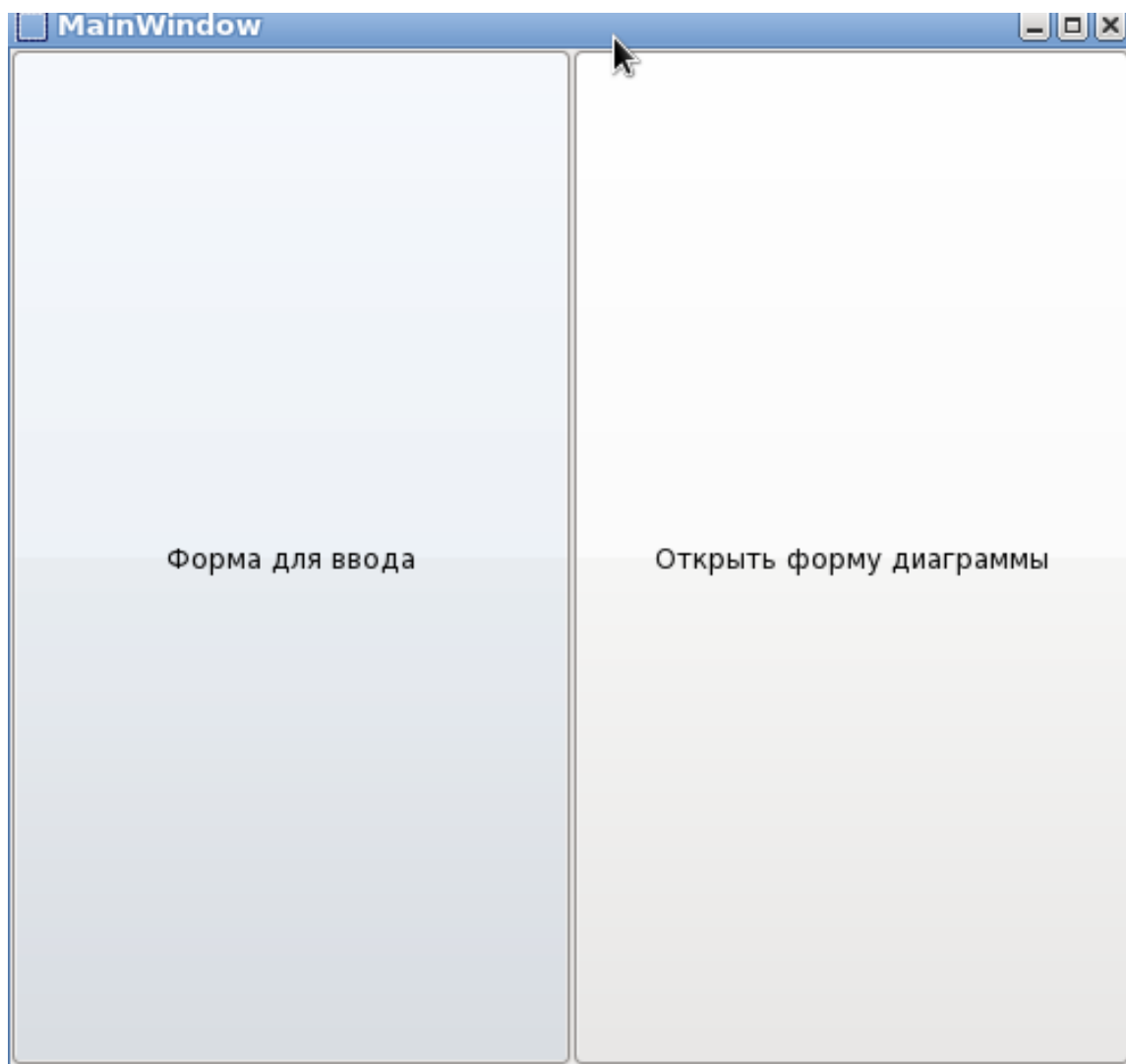


Рис. 4. Стартовое окно приложения

Также разработано приложение для визуализации полученных данных. В первоначальной версии, на экран выводилась диаграмма в следующем виде:

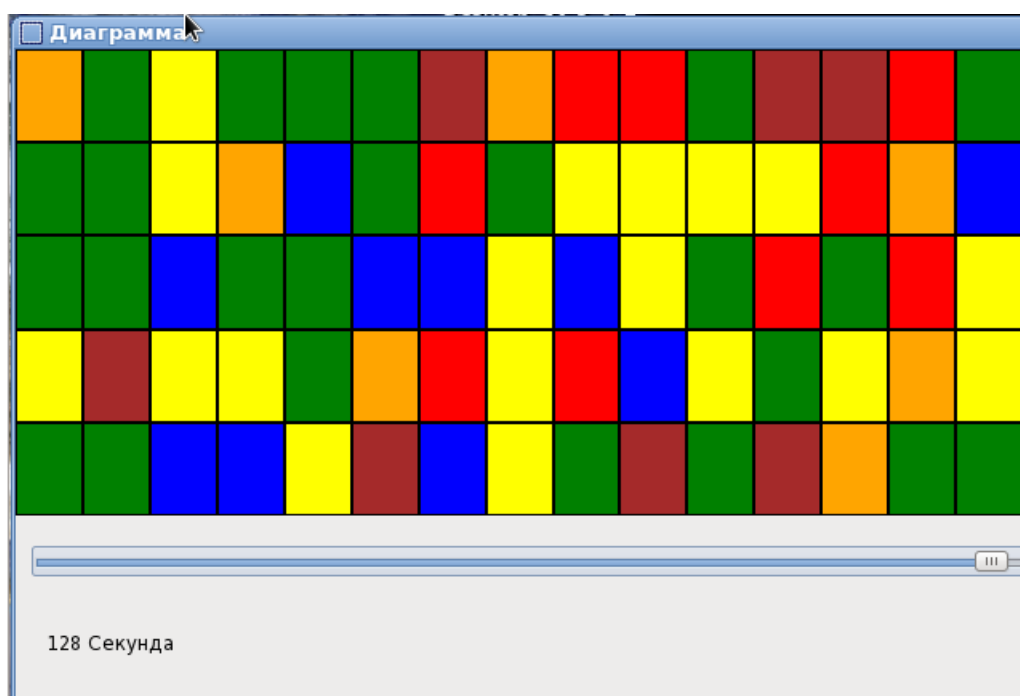


Рис. 5. Начальный вариант

Затем, после некоторых преобразований, был получен следующий результат:

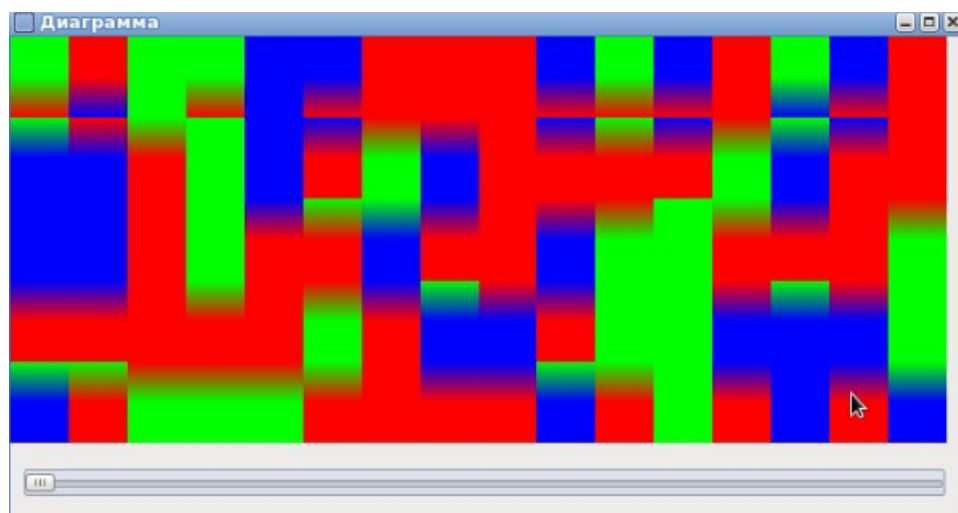


Рис. 6. Промежуточный результат

В конечном результате мы должны получить следующее цветовое представление:

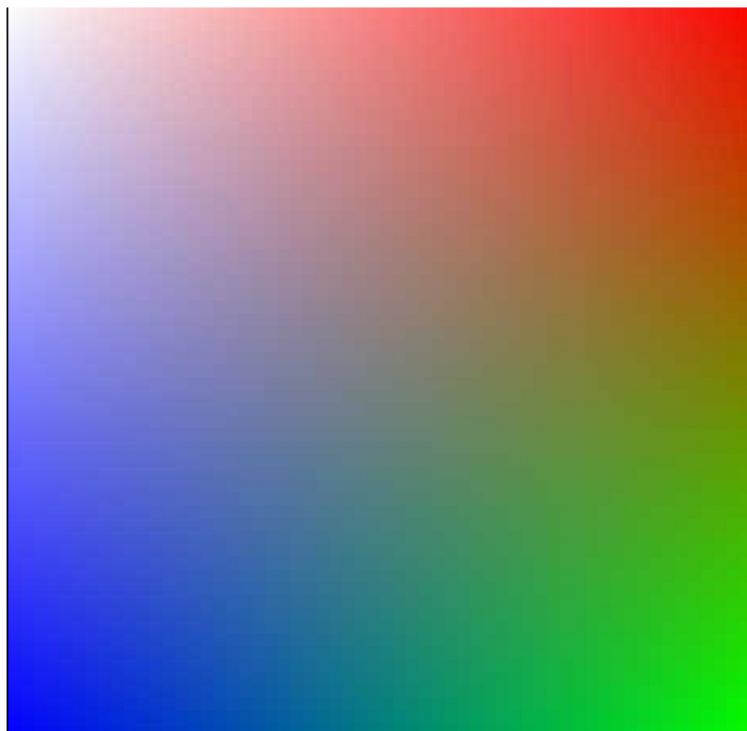


Рис. 7. Ожидаемый результат

Во второй части программы будет разработан интерфейс для ввода данных в форму, считывания файла, классификации и формирования групп, а также обучения нейронной сети.

9. Заключение

В процессе выполнения поставленной задачи было разработано несколько приложений, которые могут существенно облегчить работу медицинскому персоналу, давать более точный диагноз и назначать правильное лечение.

Список литературы и интернет-ресурсов

- [1] <http://ru.wikipedia.org/> — Википедия (свободная энциклопедия).
- [2] <http://qt-project.org/> — Информация по библиотеке QT.
- [3] <http://heart-master.com/clinic/diagnostic/pkecg/> — Поверхностное ЭКГ-картирование.
- [4] С.М. Львовский. *Набор и вёрстка в системе \LaTeX* , 3-е изд., испр. и доп. — М., МЦНМО, 2003. Доступны исходные тексты этой книги.
- [5] D. E. Knuth. *The \TeX book*. — Addison-Wesley, 1984. Русский перевод: Дональд Е. Кнут. *Все про \TeX* . — Протвино, РД \TeX , 1993.