

---

---

**ХІ Республіканская навучна-практычная канферэнцыя-конкурс  
навучна-даследавальскіх работ учащихся средних,  
средних специальных учебных заведений и студентов вузов  
«От Альфа к Омеге...» (с международным участием)  
Секция 3. Компьютерные науки и программирование  
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЕ РАБОТЫ ШКОЛЬНИКОВ**

---

---

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

Государственное учреждение образования «Высоковская средняя школа» Каменецкого района

**СИСТЕМА ДИАГНОСТИКИ ДИАБЕТИЧЕСКОЙ РЕТИНОПАТИИ “ASCLEPIUS”**

**Сацута Дмитрий Владимирович,**  
учащийся 11 «Б» класса

Лешкевич Александр Николаевич,  
учитель математики и информатики  
ГУО «Высоковская средняя школа»  
Каменецкого района,  
высшая кв. категория учителя математики и  
информатики

---

---

**XI Республиканская научно-практическая конференция-конкурс  
научно-исследовательских работ учащихся средних,  
средних специальных учебных заведений и студентов вузов  
«От Альфа к Омеге...» (с международным участием)  
Секция 3. Компьютерные науки и программирование  
РЕФЕРАТЫ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ ШКОЛЬНИКОВ**

---

---

**СИСТЕМА ДИАГНОСТИКИ ДИАБЕТИЧЕСКОЙ РЕТИНОПАТИИ “ASCLEPIUS”**

**Д. В. Сацута**

*ГУО «Высоковская средняя школа» Каменецкого района, 11 «Б» класс,  
Высокое, Беларусь*

Научный руководитель – А. Н. Лешкевич, учитель математики и информатики ГУО «Высоковская средняя школа» Каменецкого района, высшая кв. категория учителя математики и информатики.

Работа 15 с., 6 ч., 11 рис., 12 источников.

**Ключевые слова:** диабетическая ретинопатия, нейронные сети, система “Asclepius”.

В работе исследуется использование нейронных сетей в анализе изображений глазного дна и создание системы “Asclepius”. Работа системы “Asclepius” основана на анализе изображений глазного дна и определения степени заболевания. Для анализа изображения глазного дна используется нейронная сеть, обученная на соответствующих базах данных. С помощью камеры смартфона система получает изображение, после чего изображение приводится к виду, необходимому для анализа (изображение). Нейронная сеть анализирует полученные данные и выводит результат о принадлежности изображения к определенному классу (no dr, mild, moderate, severe, proliferative).

Объектом исследования являются графические файлы с изображением глазного дна.

Цель работы – разработка системы “Asclepius” для оценки стадий диабетической ретинопатии на основе изображений глазного дна.

Работа посвящена решению задачи классификации изображений глазного дна и созданию нейронной сети для анализа изображений глазного дна.

В результате исследования были получены следующие результаты:

- 1) создана модель комбинированной нейронной сети с точностью классификации 83 %;
- 2) проведено обучение и тестирование модели на тренировочной и тестовой выборке, созданных на основе базы данных;
- 3) создана система “Asclepius” на основе модели полученной нейронной сети и системы очистки для быстрой и эффективной диагностики степени поражения глазного дна диабетической ретинопатии.

Данную разработку можно использовать для диагностики степени поражения глазного дна диабетом в учреждениях здравоохранения.

## ОГЛАВЛЕНИЕ

Введение.....	4
1 Решение задачи классификации изображений глазного дна.....	5
2 Создание архитектуры нейронной сети.....	6
3 Обучение нейронной сети.....	8
4 Тестирование нейронной сети.....	10
5 Разработка системы "Asclepius".....	11
6 Алгоритм работы системы "Asclepius".....	13
Заключение.....	14
Список использованных источников.....	15

## ВВЕДЕНИЕ

Одной из самых актуальных проблем научной медицины и практического здравоохранения начала XXI века являются болезни системы обмена веществ. Сахарный диабет – одна из самых глобальных проблем нашего времени. Число людей, страдающих диабетом, с каждым годом увеличивается, и по прогнозам ВОЗ, к 2025 году их число увеличится до 550 млн. человек. Если не предпринимать своевременные меры, то последствия этого заболевания могут быть крайне плачевными. Одно из наиболее тяжёлых осложнений сахарного диабета – это диабетическая ретинопатия, поражающее сосуды сетчатой оболочки глазного яблока, наблюдаемое у 90% пациентов при сахарном диабете.

В XXI веке для диагностики заболеваний всё чаще используют технологии машинного обучения. Для оценки последствий влияния диабета на зрение была разработана система “Asclepius” для быстрой и эффективной диагностики степени поражения глазного дна.

Цель: создать систему “Asclepius” для оценки стадий диабетической ретинопатии на основе изображений глазного дна.

Задачи работы:

1. Решение задачи классификации изображений глазного дна.
2. Создание архитектуры нейронной сети.
3. Обучение нейронной сети.
4. Проведения тестирования нейронной сети.
5. Разработка системы “Asclepius”.

Объект исследования: графические файлы с изображением глазного дна.

Предмет исследования: классификация результатов на основе анализа графических файлов.

Методы исследования: анализ, моделирование, программирование.

# 1 РЕШЕНИЕ ЗАДАЧИ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ ГЛАЗНОГО ДНА

Первый этап исследования включает в себя рассмотрение задачи классификации степени поражения глазного дна диабетической ретинопатией.

Диабетическая ретинопатия наиболее часто развивается при длительном течении сахарного диабета, однако своевременное офтальмологическое обследование позволяет выявить развитие ретинопатии на ранней стадии. Нарушение зрения — одно из инвалидизирующих проявлений сахарного диабета. Слепота у пациентов с сахарным диабетом наступает в 25 раз чаще, чем среди лиц, не страдающих этим недугом (см. рисунок 1.1) [1].



Рисунок 1.1 – Стадии диабетической ретинопатии

Задача классификации – задача, в которой имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется выборкой. Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества, т. е. указать номер (или наименование) класса, к которому относится данный объект [2].

Для классификации изображений глазного дна были созданы 5 классов, соответствующие стадиям диабетической ретинопатии.

- 1) NO DR – здоровое глазное дно.
- 2) Mild – «мягкая» степень поражения.
- 3) Moderate – «умеренная» степень поражения.
- 4) Severe – «сильная» степень поражения.
- 5) Proliferative – «крайняя» степень поражения.

Таким образом перед нами представлена задача мульти-классовой классификации.

Перед классификацией были подготовлены данные для последующей обработки. Каждый графический файл был преобразован в матрицу числовых значений (см. рисунок 1.2).



Рисунок 1.2 – Обработка графического файла

Для подготовки данных использовалась функция ImageDataGenerator из библиотеки Keras:

$datageneneration = ImageDataGenerator (rescale = 1./255, zoom\_range = 0.2, width\_shift\_range = 0.2, height\_shift\_range = 0.2, validation\_split = 0.2)$

## 2 СОЗДАНИЕ АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ

Для написания нейронной сети был выбран язык программирования Python, так как он обладает высокой производительностью при обработке данных. Ещё одним важным достоинством этого языка является большое количество библиотек для работы с машинным обучением.

При решении задачи классификации изображений наиболее высокую точность показывают сверточные нейронные сети. Поэтому в основу архитектуры нашей модели были добавлены сверточные слои, которые позволяют выделить наиболее важные признаки для определения принадлежности изображения к тому или иному классу (см. рисунок 2.1).

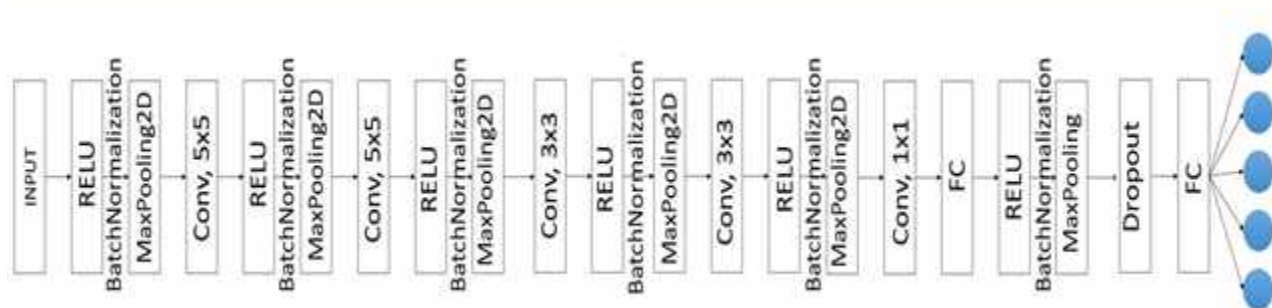


Рисунок 2.1 – Архитектура нейронной сети

Также были использованы полносвязные слои нейронов, слои активации, объединение и нормализация данных. Для улучшения работы сети, повышения её устойчивости и предотвращения переобучения применяется исключение (dropout). В итоге благодаря комбинации различных видов слоев удалось получить модель нейронной сети, которая показывает наибольшую точность при классификации изображений глазного дна.

В работе использовался метод скейлинга, который равномерно скейлит глубину/ширину/разрешение. Скейлинг – это когда фиксируются производимые внутри сетки операции и меняются лишь глубина (количество повторений одних и тех же модулей), ширина (количество каналов в свёртках) и разрешение [7].

Для создания нейронной сети использовали NAS (Network Attached Storage – сервер для хранения данных), оптимизировали гиперпараметры (повышение Accuracy).

Был проведен скейлинг для различных моделей, который позволил увеличить их точность.

В итоге с учетом всех особенностей входных изображений была написана модель комбинированной нейронной сети на языке python с использованием различных библиотек: Keras, Pandas, Scikit-learn, NumPy и др.

Разработка программного обеспечения была реализована в Google Colabe.

Google Colab — это бесплатный облачный сервис на основе Jupyter Notebook. Google Colab предоставляет всё необходимое для машинного обучения прямо в браузере, даёт бесплатный доступ к невероятно быстрым GPU (Graphics Processing Unit графический процессор) и TPU (Tensor Processing Unit – тензорный процессор от Google) (см. рисунок 2.2).



The screenshot shows a Jupyter Notebook interface with the following elements:

- Header: "Asclepius.ipynb" with a star icon, and a menu bar with "Файл", "Изменить Вид", "Вставка", "Среда выполнения", "Инструменты", "Справка", and "Сохранение...".
- Right sidebar: "Комментировать" and a user icon.
- Top bar: "+ Код" and "+ Текст" buttons, and a status bar showing "ОЗУ" and "Диск" usage.
- Code cell 1: 

```
[1] from google.colab import drive
drive.mount('/content/drive')
```
- Code cell 2: 

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```
- Code cell 3: 

```
import numpy as np
import pandas as pd

import os
for dirname, _, filenames in os.walk('/content/drive/My Drive/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```
- Output: A list of file paths, such as `/content/drive/My Drive/input/gaussian_filtered_images/gaussian_filtered_images/Moderate/f460608cf4cc.png`.

**Рисунок 2.2 – Скриншот листинга программы в Jupyter Notebook**

Использование этого сервиса позволило получить существенный прирост в скорости обучения и снизить время обучения с 37 мин до 7 – 8 мин для каждой эпохи.

### 3 ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

Используя базы данных (базы данных взяты из соревнования APTOS 2019 Blindness Detection) были созданы тестовая и тренировочная выборки.

Тестовая выборка была создана с помощью следующего кода:

```
x_test = datageneneration.flow_from_directory('/content/drive/My Drive/input/APTOS2019',  
target_size = (224, 224),  
batch_size = 32,  
class_mode = 'categorical',  
subset = 'training')
```

На основе тренировочной выборки было реализовано обучение нейронной сети.

В работе использовался алгоритм обучения с учителем. Процесс обучения с учителем представляет собой предъявление сети выборки обучающих примеров. Каждый образец подается на входы сети, затем проходит обработку внутри структуры нейронной сети, вычисляется выходной сигнал сети, который сравнивается с соответствующим значением целевого вектора, представляющего собой требуемый выход сети (см. рисунок 3.1).

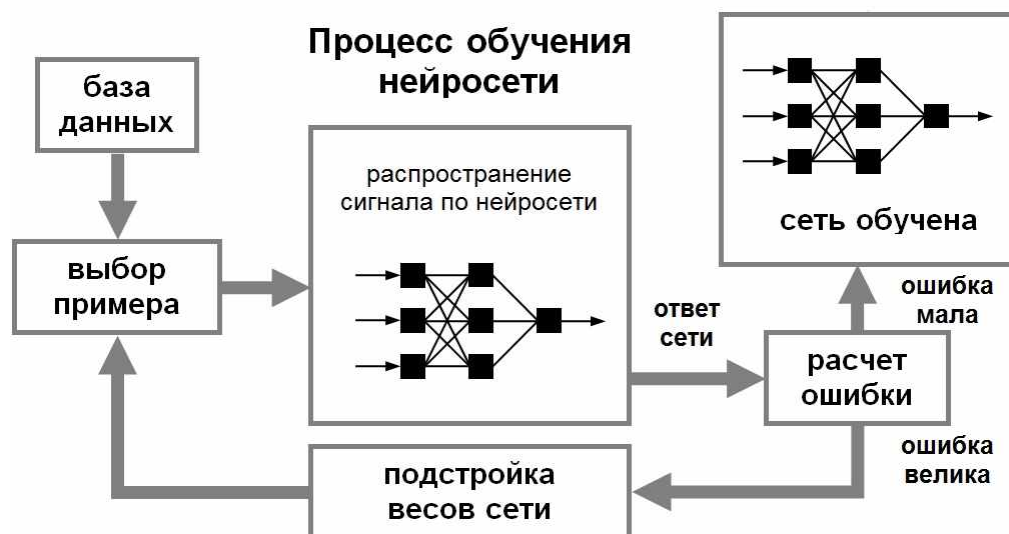


Рисунок 3.1 – Процесс обучения нейросети

Затем по определенному правилу вычисляется ошибка, и происходит изменение весовых коэффициентов связей внутри сети в зависимости от выбранного алгоритма. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

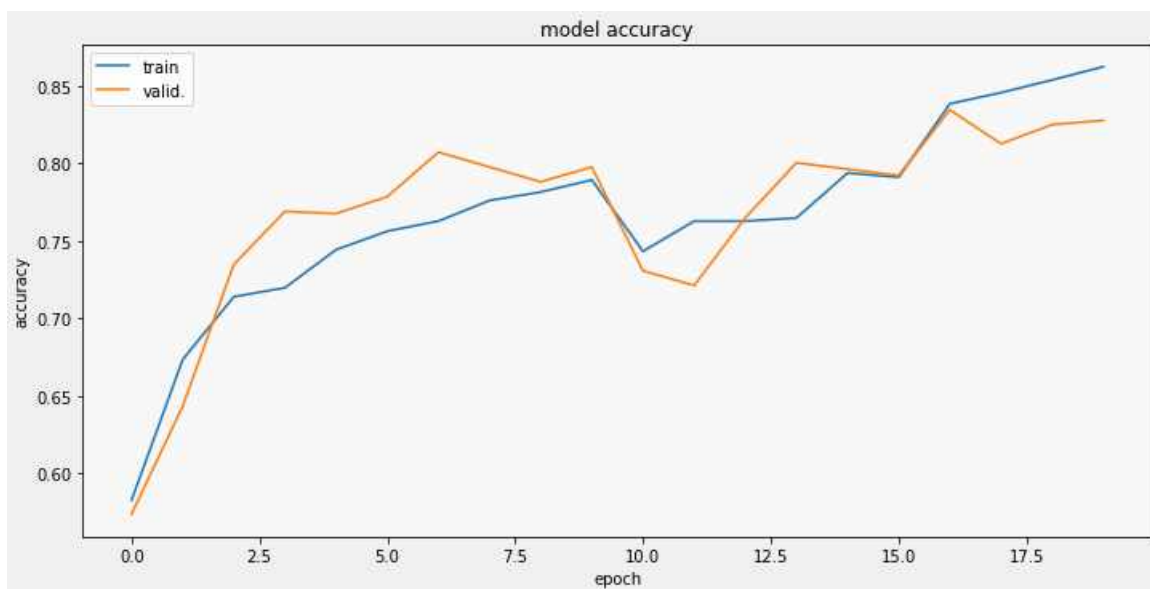
Вывод процесса обучения нейронной сети на экран имеет следующий вид:

```
Epoch 1/40  
92/92 [=====] - 2062s 22s/step - loss: 1.0349 - acc:  
0.6499 - val_loss: 1.2006 - val_acc: 0.5451  
Epoch 2/40  
92/92 [=====] - 749s 8s/step - loss: 0.7463 - acc: 0.7281 -  
val_loss: 0.9379 - val_acc: 0.6749  
Epoch 3/40  
92/92 [=====] - 736s 8s/step - loss: 0.6409 - acc: 0.7752 -  
val_loss: 0.8055 - val_acc: 0.7063  
Epoch 4/40
```



92/92 [=====] - 735s 8s/step - loss: 0.5978 - acc: 0.7776 -  
val\_loss: 0.6854 - val\_acc: 0.7486  
Epoch 5/40  
92/92 [=====] - 736s 8s/step - loss: 0.5750 - acc: 0.7844 -  
val\_loss: 0.6175 - val\_acc: 0.7541  
...  
Epoch 39/40  
92/92 [=====] - 736s 8s/step - loss: 0.0836 - acc: 0.9717 -  
val\_loss: 0.0884 - val\_acc: 0.8211  
Epoch 40/40  
92/92 [=====] - 737s 8s/step - loss: 0.0728 - acc: 0.9758 -  
val\_loss: 0.0785 - val\_acc: 0.8301

Таким образом после 40 эпох сеть является обученной, процесс обучение можно увидеть на данных графика (см. рисунок 3.2).



**Рисунок 3.2 – График процесса обучения нейросети**

После обучения было выполнено сохранение модели с весами, полученными в процессе обучения для дальнейшего использования.

## 4 ТЕСТИРОВАНИЕ НЕЙРОННОЙ СЕТИ

Следующий этап разработки включал в себя проведение тестирования, используя тестовую выборку и получения различных метрик машинного обучения. Для оценки модели на тестовых данных был запущен следующий код:

```
scores = model.evaluate (x_test, y_test, verbose = 0)
```

Следующий код программы генерирует прогноз для тестовой выборки:

```
y_pred = model.predict_classes (x_test, batch_size = 32)  
print ("Точность обученной модели: ", round (scores[1] * 100), "%")
```

Точность обученной модели: 83.0 %.

С помощью метрики Accuracy была оценена точность модели. Также для оценки качества работы алгоритма на каждом из классов по отдельности были использованы метрики precision, recall и F-мера, которые подтвердили высокое качество созданной модели.

Таким образом, по результатам тестирования нейронная сеть показала точность 83 %.

## 5 РАЗРАБОТКА СИСТЕМЫ “ASCLEPIUS”

Система “Asclepius” включает в себя:

- мобильную часть;
- серверную часть;
- облачная часть.

Мобильная часть была реализована в Android Studio на языке программирования Java (см. рисунок 5.1).

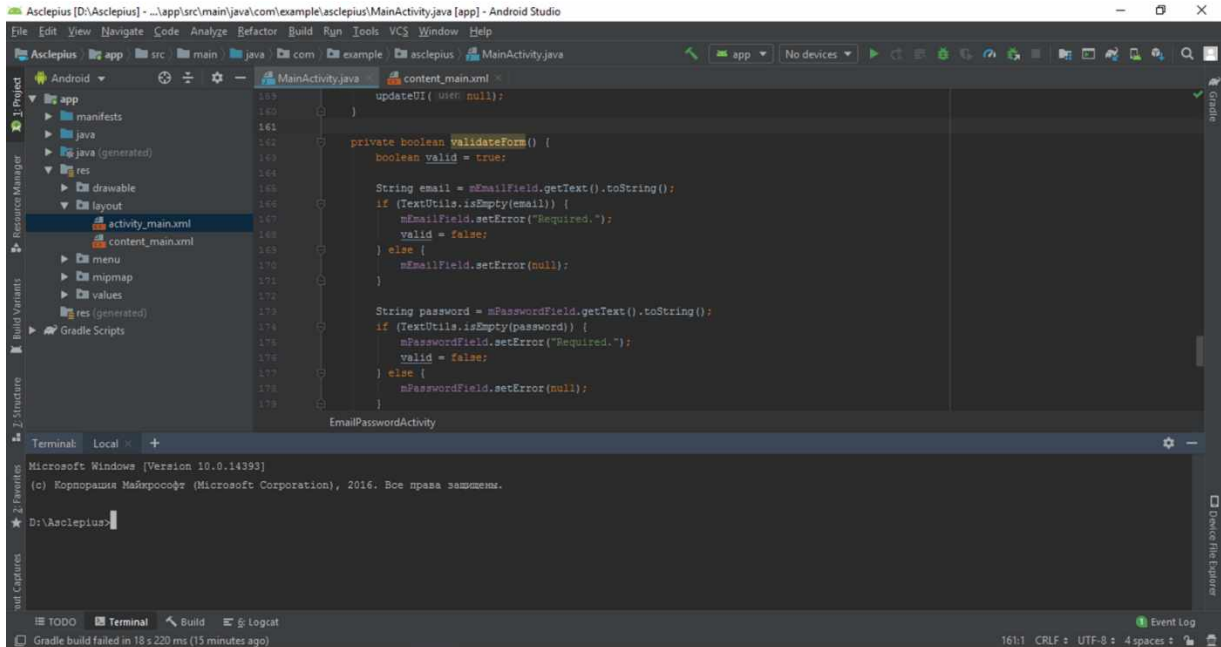


Рисунок 5.1 – Скриншот листинга программы в Android Studio

В ней реализованы следующие функции:

- регистрация пользователя;
- аутентификация пользователя;
- классификация глазного дна в режиме реального времени;
- просмотр предыдущих результатов.

Интерфейс приложения имеет следующий вид (см. рисунок 5.2).

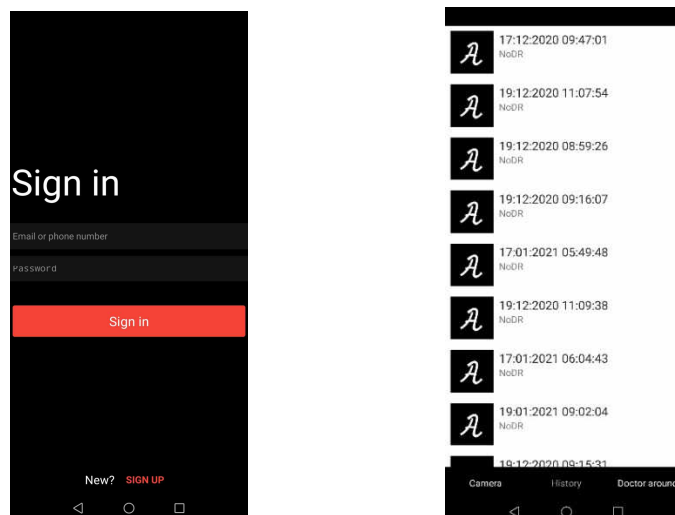


Рисунок 5.2 – Скриншот мобильного приложения “Asclepius”

Серверная часть представляет собой программу, написанную на python. Она предназначена для очистки и классификации изображений с помощью комплекса нейронных сетей в режиме реального времени.

Система очистки представляет собой нейронную сеть, выделяющую необходимый фрагмент на изображении. Она удаляет задний фон и оставляет лишь изображение глазного дна, к которому применяет фильтр Гаусса. Фильтр Гаусса снижает шумы изображения и выполняет первоначальную свертку. После фильтрации изображение передается во вторую нейронную сеть, где осуществляется классификация (см. рисунок 5.3).

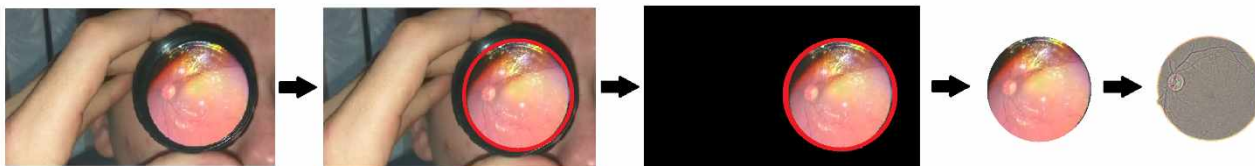


Рисунок 5.3 – Фильтрация изображения

Облачная часть выполняет следующие функции:

- хранение результатов классификации;
- хранение изображений;
- хранение данных пользователей.

Облачная часть представлена базами данных Firebase от Google (см. рисунок 5.4). При регистрации каждому пользователю присваивается уникальный идентификатор, который определяет раздел в базе данных, где хранятся его результаты и изображения глазного дна, с которыми работает серверная часть.

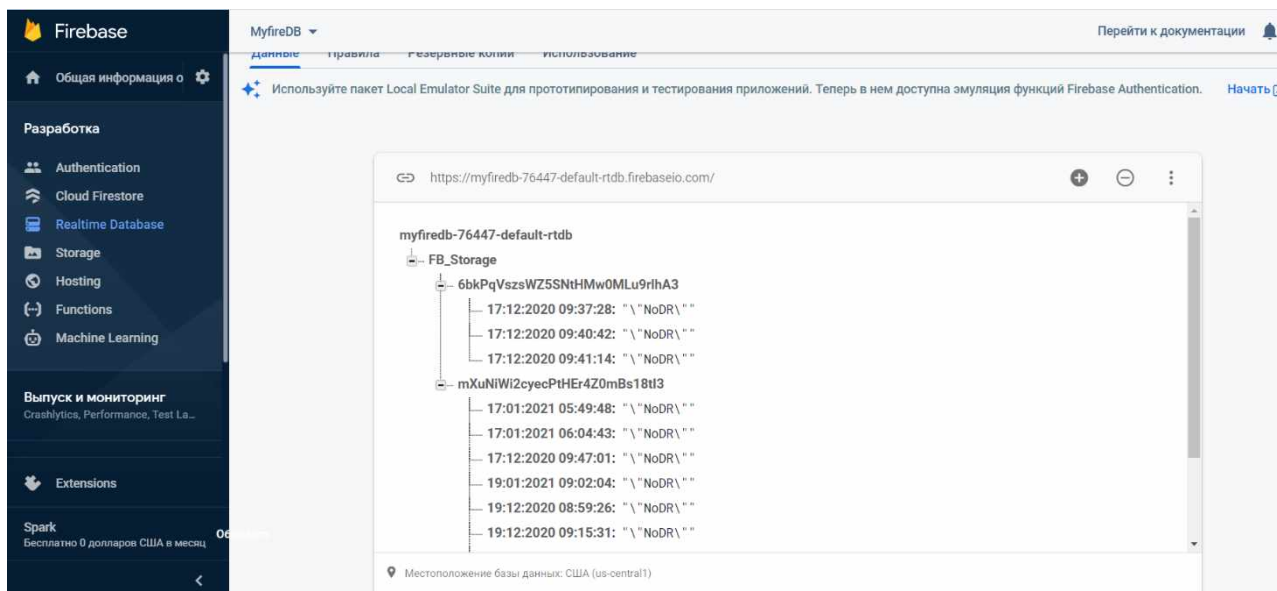


Рисунок 5.4 – Скриншот базы данных

## 6 АЛГОРИТМ РАБОТЫ СИСТЕМЫ “ASCLEPIUS”

Алгоритм работает по следующей схеме (см. рисунок 6.1):



Рисунок 6.1 – Схема алгоритма работы системы «Asclepius»

При загрузке мобильного приложения перед пользователем появляется экран аутентификации, где ему предлагается ввести логин и пароль. Если пользователь не зарегистрирован, то необходимо пройти простую процедуру регистрации. После этого пользователь попадает на главный экран приложения, где он видит историю результатов. Нажимая кнопку Camera на навигационной панели открывается камера смартфона для получения изображения. Чтобы получить изображение глазного дна нужно использовать специальное оборудование или линзу 20D.

Полученное изображение отправляется в облачное хранилище, откуда его экспортирует серверная часть системы “Asclepius”.

К изображению применяется система очистки, которая выделяет изображение глазного дна. После очистки обработанное изображение поступает в нейронную сеть для классификации по категориям. Результат классификации сохраняется в облачном хранилище и отправляется на мобильное устройство.

На экране смартфона выводится результат классификации.

## ЗАКЛЮЧЕНИЕ

Создана модель комбинированной нейронной сети с точностью классификации 83 %.

Проведено обучение и тестирование модели на тренировочной и тестовой выборке, созданных на основе базы данных.

Создана система “Asclepius” на основе модели полученной нейронной сети и системы очистки для быстрой и эффективной диагностики степени поражения глазного дна диабетической ретинопатии.

Данную разработку можно использовать для диагностики степени поражения глазного дна диабетом в учреждениях здравоохранения.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Диабетическая ретинопатия [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/диабетическая\\_ретинопатия](https://ru.wikipedia.org/wiki/диабетическая_ретинопатия). – Дата доступа: 21.09.2020.
2. Задача классификации [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/задача\\_классификации](https://ru.wikipedia.org/wiki/задача_классификации). – Дата доступа: 21.09.2020.
3. Начинаем работать с Firebase на Android [Электронный ресурс]. – Режим доступа: <https://code.tutsplus.com/ru/tutorials/get-started-with-firebase-for-android--cms-27248>. – Дата доступа: 12.11.2020.
4. Новые архитектуры нейронных сетей [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/498168/>. – Дата доступа: 10.10.2020.
5. Обучение нейросети с учителем, без учителя, с подкреплением — в чем отличие? Какой алгоритм лучше? [Электронный ресурс]. – Режим доступа: <https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/>. – Дата доступа: 20.09.2020.
6. Распознавание [Электронный ресурс]. – Режим доступа: <http://www.machinelearning.ru/wiki/index.php>. – Дата доступа: 20.09.2020.
7. Рубрика «Читаем статьи за вас». Июль – сентябрь 2019 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/472671/>. – Дата доступа: 10.10.2020.
8. Сверточная нейронная сеть [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Сверточная\\_нейронная\\_сеть](https://ru.wikipedia.org/wiki/Сверточная_нейронная_сеть). – Дата доступа: 20.09.2020.
9. APTOS 2019 Blindness Detection [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/c/aptos2019-blindness-detection>. – Дата доступа: 29.09.2020.
10. Google AI Blog [Электронный ресурс]. – Режим доступа: <https://ai.googleblog.com/>. – Дата доступа: 05.01.2021.
11. Matplotlib [Электронный ресурс]. – Режим доступа: <https://matplotlib.org/#>. – Дата доступа: 15.10.2020.
12. NumPy [Электронный ресурс]. – Режим доступа: <https://numpy.org/#>. – Дата доступа: 12.10.2019.
13. Python Data Analysis Library [Электронный ресурс]. – Режим доступа: <https://pandas.pydata.org/>. – Дата доступа: 12.10.2020.