

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**СОЗДАНИЕ СЕРВИСА ДЛЯ ОБРАБОТКИ И АНАЛИЗА СНИМКОВ  
СТОПЫ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ И  
КОМПЬЮТЕРНОГО ЗРЕНИЯ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы  
направления 02.03.02 — Фундаментальная информатика и информационные  
технологии  
факультета КНиИТ  
Забоева Максима Владиславовича

Научный руководитель  
доцент

\_\_\_\_\_

Б. А. Филиппов

Заведующий кафедрой  
к. ф.-м. н.

\_\_\_\_\_

С. В. Миронов

Саратов 2024

## ВВЕДЕНИЕ

В современном мире обработка и анализ изображений являются одной из наиболее активно развивающихся областей искусственного интеллекта. Распознавание и классификация объектов на изображениях стали возможными благодаря применению нейронных сетей — мощных алгоритмов, вдохновленных биологическими нейронными сетями человеческого мозга.

Одной из важных задач в области обработки изображений является анализ медицинских снимков. Анализ снимков стоп, сделанных на плантоскопе, имеет широкий спектр применений, включая биометрическую идентификацию, медицинскую диагностику и анализ походки. Правильное распознавание стопы может быть полезным инструментом в улучшении качества медицинского обслуживания. Вместе с факультетом фундаментальной медицины и медицинских технологий была поставлена задача по обработке снимков стопы и созданию удобно в использовании сервиса, в котором пользователи могут загружать свои снимки стоп и получать подробные и качественные характеристики анализа, вместе с визуальным отображением контрольных точек и линий.

Целью выпускной квалификационной работы является создание онлайн веб-сайта для анализа снимков стопы с использованием нейронных сетей и компьютерного зрения.

Для этого поставлены следующие задачи:

- Создать архитектуру нейронной сети и обучить ее нахождению масок стопы;
- Разработать алгоритм расчета по маске контрольных точек стопы и таким ее характеристикам, как индекс Штритера и коэффициент распластанности;
- Создать интернет сервис для онлайн исследования стопы, в котором пользователь может войти в свой личный кабинет, загрузить снимок стопы и получить по нему результат, а так же просмотреть анализы прошлых исследований.

**Структуры и объем работы.** Для решения поставленных задач выполнена выпускная квалификационная работа, включающая в себя введение, 2 основные главы, заключение, список использованных источников из 20 наименований и 2 приложения. Работа изложена на 100 страницах, содержит 65 рисунков. Первая глава имеет название «Обзор исследуемой области» и содер-

жит основную информацию о нейронных сетях, компьютерном зрении, задачи подоскопии, клиент-серверной модели. Вторая глава имеет название «Реализация интернет сервиса и разработка алгоритмов оценки стопы», данная глава содержит подробное описание процесса выполнения работы. Выпускная квалификационная работа заканчивается заключением, списком использованных источников, а также приложениями с кодом А-Б.

Данная работа выполнялась в рамках проектной деятельности «Стартап как диплом» совместно с студентом 2 курса магистратуры факультета КНИиТ направления МОАИС Пантелеевым Дмитрием Сергеевичем.

## 1 Основное содержание работы

**Нейронные сети.** Нейронные сети являются важным инструментом в области машинного обучения и искусственного интеллекта. Они были вдохновлены биологическими нейронными сетями, которые присутствуют в человеческом мозге и отвечают за обработку информации и принятие решений.

Как и в биологии нейрон в математической модели должен иметь следующие параметры:

- Входные и выходные значения, так как и в биологии нейрон должен получать нервный импульс и передавать его соседям;
- Должна быть связь с соседним нейроном;
- Он должен по достаточно простой формуле обрабатывать входные значения и передавать результат вычислений дальше;
- Выходное значение нейрона должно моделировать возбуждение нейрона.

У искусственного нейрона каждый вход имеет свой вес  $w$ , Входные значения  $x$  приходят либо из внешней среды либо из других нейронов. Важность каждого входа выражается весом связи  $w$ . Искусственный нейрон считает взвешенную сумму входов, прибавляя смещение  $w_0$ , что является порогом возбуждения. Далее к результату применяется функция активации  $F(S)$ , если результирующее значение достаточно велико, то считается, что нейрон возбуждился. Для анализа изображений чаще всего берут функцию активации ReLu — Rectified liner unit. Так как при моделировании искусственного нейрона стоит брать не два состояния нейрона, а именно возбуждился ли он или нет, а числовую шкалу или «шкалу возбуждения». Данная функция возвращает 0, если принимает отрицательный аргумент, в случае же положительного аргумента, функция возвращает само число. То есть она может быть записана как  $f(z) = \max(0, z)$ .

**Компьютерное зрение.** Компьютерное зрение – это область искусственного интеллекта, которая изучает, как компьютеры могут получать, анализировать и понимать изображения и видео так же, как это делает человеческий глаз. Оно объединяет методы обработки изображений, распознавания образов, машинного обучения и компьютерного зрения для решения различных задач, таких как распознавание объектов, отслеживание движущихся объектов, сегментация изображений и многое другое.

В компьютерном зрении используются различные методы и алгоритмы

для обработки и анализа изображений:

1. Предобработка изображений — улучшение качества изображений;
2. Извлечение признаков — выделение характеристик изображения;
3. Классификация и распознавание;
4. Сегментация — выделение объектов на изображении;

**Сверточная нейронная сеть.** Для работы с изображениями была создана специальная архитектура нейронной сети — сверточная нейронная сеть. Каждый нейрон (ячейка матрицы) последующего слоя соединен только с нейронами определенной области предыдущего слоя. Нейроны второго слоя могут иметь пересекающиеся области с предыдущего слоя. Операция двумерной свертки представляет собой простой процесс, который начинается с применения ядра в виде матрицы весов. Ядро проходит по двумерному изображению, выполняя поэлементное умножение с соответствующей частью входных данных, на которой оно в данный момент находится, и затем суммирует все полученные значения для формирования одного выходного пикселя. Выходные признаки представляют собой взвешенную сумму признаков входных данных (с весами, заданными ядром), которые примерно находятся в том же месте, что и выходной пиксель на входном слое. Независимо от того, попадает ли входной признак в «примерно то же место», это определяется тем, находится ли он в зоне действия ядра, формирующего выходные данные, или нет. Таким образом, размер ядра сверточной нейронной сети определяет количество признаков, которые будут объединены для создания нового признака на выходе.

**Сегментационная нейронная сеть** Сегментация в компьютерном зрении — это процесс разделения изображения на семантические или структурные части. Вместо того, чтобы просто классифицировать всё изображение целиком, сегментация позволяет выделить отдельные объекты, регионы или пиксели, которые имеют схожие характеристики или принадлежат к одному классу. Чаще всего сегментацию используют для выделения контуров объектов, в этом случае определяются границы и формы относительно цветовой гаммы или текстуре.

**Нейронная сеть U-net.** U-net — полносвязная сверточная нейронная сеть, позволяющая решать задачи сегментации изображений. Для U-net характерно:

1. Высокий показатель точности на биомедицинских данных;
2. Использование небольшого кол-во данных.

U-net обучается методом стохастического градиентного спуска на основе входных изображений и соответствующих им масок сегментации. Применяемая попиксельно, функция soft-max вычисляет энергию по окончательной карте свойств вместе с функцией кросс-энтропии:

$$E = \sum_{X \in \Omega} w(X) \log(\rho_{l(X)}(X))$$

Граница разделения вычисляется с использованием морфологических операций. Затем вычисляется карта весовых коэффициентов:

$w(X) = w_c(X) + w_0 * \exp(-\frac{(d_1(X)+d_2(X)^2)}{2\sigma^2})$ , где  $w_c$  — карта весов для балансировки частот классов,  $d_1$  — расстояние до границы ближайшей ячейки,  $d_2$  — расстояние до границы второй ближайшей ячейки.

**Задача подоскопии.** Подоскопия — это обследования состояния стоп и определения их уплощенности с помощью плантоскопа. Плантоскоп представляет собой современное медицинское устройство нового поколения, предназначенное для визуального диагностирования состояния сводов стопы. Назначение подоскопии:

1. фиксация различных степеней уплощения сводов стопы;
2. определение зон перегрузок, а также распределения давления на определенные сегменты стопы;
3. оценка положения пяточной кости;
4. выявление состояния деформации позвоночника (сколиоз, лордоз, кифоз и др.);
5. фиксация нарушений в тазобедренном суставе.

**Клиент-серверная модель.** Клиент-серверная модель — основная модель, использующаяся при проектировании Rest API проекта. Клиент — программа, которая запрашивает у сервера доступ к ресурсам, используя API.

Сервер — программа, в которой хранятся и обрабатываются данные. Он может располагаться на одном или нескольких компьютерах. В результате получается структура, при которой клиент направляет к серверу запрос, а в ответ получает данные, такой принцип позволяет разделить клиента и сервера между собой и не создавать связи между ними, что позволяет им работать отдельно друг от друга, что позволяет ускорить процесс разработки.

## **2 Реализация интернет сервиса и разработка алгоритмов оценки стопы**

В практической части дипломной работы был реализован веб-сайт, где внешняя часть сайта (клиент) была написана на языке JavaScript, используя библиотеку React на платформе Node.js, а внутренняя часть (сервер) на языке Python используя фреймворк Django с базой данных на SQLite. В программе используются следующие алгоритмы и методы:

- Для разработки нейронной сети:
  1. Аугментация данных;
  2. Подготовка данных для обучения;
  3. Спроектировать архитектура нейронной сети U-net и ее обучение;
  4. Обработка и разметка полученных масок;
- Для сервера:
  1. Разработать архитектура базы данных;
  2. Авторизация пользователя с использованием JWT токенов;
  3. Получение и валидация данных клиента, полученных через API;
  4. Анализ снимков с использованием нейронной сети.
- Для клиента:
  1. Регистрация и авторизация пользователя на веб-сайте;
  2. Личный кабинет с возможностью просмотра загруженных исследований снимков стоп;
  3. Возможность загружать снимки стоп;
  4. Просмотр аналитики по результатам исследований.

**Подготовка данных.** Исходные данные были представлены в виде набора медицинских фотографий стоп, сделанных на плантоскопе. Всего представлено 50 снимков, разрешением от 1600x960 до 3840x2160. Для каждого снимка стопы была найдена ручным способом маска с использованием программы Labelme, после этого, для обучения нейросети был написан алгоритм аугментации данных для расширения входного датасета по следующему алгоритму:

1. Берем случайную точку на исходном изображении;
2. Обрезаем вокруг нее новое изображение размером 256x256;
3. Сохраняем полученный фрагмент, как новое изображение в папку.

**Разработка архитектуры и обучение сети U-net.** Архитектура нейрон-

ной сети U-net является сверточной, то есть имеет сужающийся путь, расширяющийся путь. На основе этого была спроектирована архитектура нейронной сети:

- Сужающийся путь — Encoder представлен 4 сверточными слоями с количеством фильтров 64, 128, 256, 512 соответственно с размером матрицы  $3 \times 3$ . Они представлены матрицами весов, которые применяются к локальным областям входных данных и позволяют выделить важные признаки. После свертки с фильтром происходит активация, результатом которой является выходная карта признаков. Использовался padding с параметром same. Он добавляет недостающие пиксели нулевого значения вокруг входных данных, чтобы размер выхода был таким же, как размер входа.
- После каждого сверточного слоя, необходимо сделать субдискретизацию — max pooling, с размером окна (2, 2), который уменьшает размерность входных данных. Для операции свертки была взята функция библиотеки TensorFlow — Conv2.
- Средний блок, который используется для сохранения и агрегации информации с высоким уровнем абстракции. Он выполняет следующие задачи:
  - Извлечение более абстрактных признаков, из-за большего размера фильтров, который равен 1024.
  - Улучшение представление признаков, удаляя ненужные шумы и усиливая значимые признаки.
  - Благодаря pooling и свертке в encoder, модель теряет некоторую контекстную информацию о положении объектов на изображении. Средний блок помогает сохранить эту информацию и предоставить ее декодеру для более точного восстановления объектов.
- расширяющийся путь — Decoder, который является обратным Encoder, то есть каждый слой свертки имеет количество фильтров 512, 256, 128, 64, а перед операцией свертки, идет объединение входных тензоров, получая на вход результат среднего блока и Encoder.

Обучение модели проходило на 25 эпохах, с размером пакета 10 — то, сколько образцов будет обрабатываться на каждой итерации обучения модели. Так же для определения оценки модели на каждой эпохе во время обучения



была использована функция `validation_data`. После обучения веса модели сохраняются в файл с разрешением `.h5`

По полученным результатам нейронной сети были написаны алгоритмы определения индекса Штритера и коэффициента продольного уплощения, а так же построение характеристических линий и точек на изображении.

**Разработка серверной части сайта.** Для хранения данных была разработана следующая архитектура базы данных:

1. `UserAccount` — таблица с данными пользователя, имеющие следующие поля:
  - `firstname` — логин пользователя;
  - `email` — почта;
  - `password` — пароль;
  - `pers_identifier` — персональный номер;
  - `secure_number` — секретный ключ;
2. `UserPhotos` — таблица с данными пользователя, имеющие следующие поля:
  - `user_name` — ID пользователя;
  - `user_photo` — загруженное фото для исследования;
  - `date_upload` — дата загрузки;
  - `photo_pers_identifier` — персональный номер снимка;
  - `photo_secure_number` — секретный ключ снимка;
3. `UserResultPhotos` — таблица с данными пользователя, имеющие следующие поля:
  - `user_photo_first` — 1 результирующий снимок;
  - `user_photo_second` — 2 результирующий снимок;
  - `user_photo_third` — 3 результирующий снимок;
  - `date_upload` — дата исследования;
  - `coeficientcommont` — коэффициент продольного уплощения;
  - `coeficientK` — коэффициент  $k$ ;
  - `coeficientI` — коэффициент  $I$ ;
  - `coefficient_alpha` — угол  $\alpha$ ;
  - `preview_result` — итоговый результат;
  - `result_view` — пояснения результата;
  - `photo_pers_identifier` — персональный номер снимка;

— `photo_secure_number` — секретный ключ снимка;

Для программной разработки базы данных был использован язык Python и фреймворк Django с базой данных SQLite. Были прописаны обработчики методов получения или изменения данных формата GET, POST, PUT, DELETE, а так же прописаны URL маршруты:

- `register` — регистрация пользователя;
- `me` — получение информации о пользователе;
- `takephoto` — получение снимков пользователя;
- `postphoto` — загрузка нового исследования;
- `resultphoto` — получение подробно отчета об исследовании.

**Разработка клиентской части сайта.** Клиентская часть сайта, или Front часть была написана на языке JavaScript, с использованием библиотеки React на платформе Node.js. В качестве языка описания компонентов использовался декларативный язык разметки HTML с языком стилизации CSS, а так же была подключена библиотека Bootstrap, имеющая в своем наличии готовые компоненты, которые можно быстро интегрировать в код проекта.

Для навигации по сайту, была использована библиотека `react-router-dom` и прописаны следующие страницы:

- `Home` — главная страница;
- `Login` — страница авторизации;
- `Register` — страница регистрации;
- `ProfilePage` — личный кабинет пользователя;
- `PostPhoto` — загрузка нового снимка;
- `ResultPhoto` — просмотр исследования.

Для организации хранения была использована библиотека `react-redux`, с помощью нее была разработана структура хранения информации об пользователе.

## ЗАКЛЮЧЕНИЕ

В результате работы была изученная нейронная сеть U-net. По итогам изучения была спроектирована архитектура и обучена модель, которая способна определить маски на стопе. Также были написаны алгоритмы построения контрольных точек на полученных масках соединенные линиями. Данные линии характеризуют стопу и могут говорить о различных ее медицинских отклонениях. Датасет, использованный для обучения модели, можно улучшать, используя цветные изображения, другие архитектуры нейронной сети и более большой объем данных, тем самым качество модели возрастет, чтобы получить более точные результаты подоскопии. Для пользователей был разработан удобный в использовании интернет сервис, в котором они смогут зарегистрировать свой личный кабинет и загружать свои снимки стоп для получения результатов нейронной сети и генетических алгоритмов, а так же просматривать результаты сквозь время, для проведения сравнительно анализа.

**Отдельные части бакалаврской работы были представлены на конференции:**

- участие в международной конференции молодых ученых фундаментальная и прикладная медицина «Biomeeting», СГУ Саратов, 2023 Пантелеев Д.С., Забоев М.В., «Система диагностики плоскостопия средствами компьютерного зрения»;
- Участие в IV международной научно-практической конференции «Анализ данных в медицине» ИСП РАН, 17 Мая 2024г., Великий Новгород;
- Участие в ежегодной всероссийской школесеминар «Метода компьютерной диагностики в биологии и медицине», СГУ, СГМУ, 16 ноября 2023, «Проблемы подготовки данных для анализа медицинских снимков».

**Основные источники информации:**

1. Коул, А. Искусственный интеллект и компьютерное зрение. / А. Коул. — O'Reilly Media: П., 2023. — С. 608.
2. Szeliski, R. Computer Vision: Algorithms and Applications / R. Szeliski. — Springer: Springer, 2010. — С. 979.
3. Красикова, И. Плоскостопие. Профилактика и лечение / И. Красикова. — Корона-Принт: Корона-Принт, 2021. — С. 128.
4. Васильева, А. Плоскостопие. Самые эффективные методы лечения / А. Васильева. — Крылов: Крылов, 2019. — С. 160.

5. *Стефанов, С.* React. Быстрый старт / С. Стефанов. — Питер: Питер, 2023. — С. 304
6. *Маккинни, У.* Python и анализ данных / У. Маккинни. — М.: ДМК Пресс, 2015. — С. 482.