

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,  
профессор

\_\_\_\_\_ Л.Б. Соколинский

« \_\_\_\_ » \_\_\_\_\_ 2024 г.

**Разработка веб-приложения для подборки музыки  
с использованием нейросетевых технологий**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 02.03.02.2024.308-552.ВКР

Научные руководители:  
ст. преподаватель кафедры СП  
\_\_\_\_\_ Н.С. Силкина,

м.н.с. кафедры СП  
\_\_\_\_\_ А.Е. Старков

Автор работы,  
студент группы КЭ-401  
\_\_\_\_\_ Д.А. Стахеев

Ученый секретарь  
(нормоконтролер)  
\_\_\_\_\_ И.Д. Володченко  
« \_\_\_\_ » \_\_\_\_\_ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_ Л.Б. Соколинский

29.01.2024 г.

### **ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра  
студенту группы КЭ-401**

Стахееву Дмитрию Андреевичу,  
обучающемуся по направлению

02.03.02 «Фундаментальная информатика и информационные технологии»

**1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)

Разработка веб-приложения для подборки музыки с использованием  
нейросетевых технологий.

**2. Срок сдачи студентом законченной работы:** 03.06.2024 г.

**3. Исходные данные к работе**

3.1. Алекс Б. Ева П. React и Redux. Функциональная веб-разработка. // Изда-  
тельский дом «Питер», 2017. – 336 с.

3.2. Миша Г. Йен О. Высокопроизводительные Python-приложения. // Бом-  
бора, 2022. – 528 с.

3.3. Джейк В. Python для сложных задач. Наука о данных и машинное обуче-  
ние. // O'REILLY, 2017. – 576 с.

3.4. Анатолий П. Основы искусственного интеллекта в примерах на Py-  
thon. // БХВ-Петербург, 2021. – 444 с.

**4. Перечень подлежащих разработке вопросов**

4.1. Создание дизайна веб-приложения.

4.2. Разработка frontend части веб-приложения.

4.3. Разработка backend части веб-приложения.

4.4. Создание и обучение нейронной сети для распознавания эмоций человека по фотографии лица.

**5. Дата выдачи задания:** 29.01.2024 г.

**Научные руководители:**

ст. преподаватель кафедры СП

Н.С. Силкина

м.н.с. кафедры СП

А.Е. Старков

**Задание принял к исполнению**

Д.А. Стахеев

## ГЛОССАРИЙ

1. *Нейронная сеть* – это тип технологии искусственного интеллекта, которая учит компьютеры обрабатывать данные также, как человеческий мозг. Это процесс машинного обучения, известного как глубокое обучение, в котором используются узлы и нейроны, соединенные друг с другом в иерархическую структуру, подобную человеческому мозгу [25].

2. *Глубокое обучение (Deep Learning)* – совокупность методов машинного обучения, основанных на обучении представлениям, а не специализированных алгоритмах [25].

3. *VGG (Very Deep Convolutional Network)* – стандартная архитектура сверточной нейронной сети с несколькими уровнями, которая была разработана для повышения точности классификации за счет увеличения глубины CNN. «Глубина» относится к количеству слоев свертки с VGG16 или VGG19, состоящих из 16 и 19 слоев свертки соответственно [14].

4. *Алгоритм SDM* – алгоритм, позволяющий выделять отметки на лице, для дальнейшего выравнивания граней в соответствии с полученными отметками [14].

5. *DeepID2* – алгоритм для классификации лиц, использующий объединенную байесовскую модель в качестве классификатора. Алгоритм основан на выборе семи наборов функций по отдельности. Полученные результаты объединяются с использованием SVM [14].

6. *SVM (Support Vector Machines)* – линейный алгоритм используемый в задачах классификации и регрессии. Основой алгоритма является процесс создания линий или гиперплоскостей, которые разделяют данные на классы авто [39].

7. *DCNN (Deep Convolution Neural Network)* – класс искусственных нейронных сетей использующий алгоритм свертки с использованием ядра свертки. Чаще всего применяются для анализа визуальных образов [14].

## **ОГЛАВЛЕНИЕ**

ГЛОССАРИЙ.....	2
ВВЕДЕНИЕ.....	6
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	9
1.1. Сравнительный анализ аналогов .....	9
1.2. Алгоритм определения эмоций по фотографии.....	11
1.3. Алгоритм подборки музыки.....	11
2. ПРОЕКТИРОВАНИЕ .....	13
2.1. Определение требований.....	13
2.2. Проектирование веб-приложения .....	16
2.3. Проектирование нейронной сети .....	19
3. РЕАЛИЗАЦИЯ .....	25
3.1. Программные средства реализации .....	25
3.2. Разработка веб-приложения .....	26
3.3. Подготовка и предобработка набора данных .....	28
3.4. Построение и обучение нейронной сети .....	29
4. ТЕСТИРОВАНИЕ .....	33
ЗАКЛЮЧЕНИЕ .....	36

## **ВВЕДЕНИЕ**

### **Актуальность**

Музыка – основной источник вдохновения. Именно она занимает важную роль в жизни людей и способна прямо воздействовать на эмоции и настроение слушателя. Чтобы подбирать композиции для каждого в соответствии с его интересами, большинство музыкальных площадок постоянно совершенствуют свои методы подборки плейлистов и треков.

Недостатком подхода музыкальных площадок является то, что они основываются на истории прослушивания и подбирают музыку соответствующего жанра. Помимо данного метода существует способ классификации пользователя с определенной группой людей, объединенных общими интересами. Распределяя пользователей на подгруппы упрощается алгоритм выбора музыки для конкретной группы, так как можно учитывать мнение сразу нескольких пользователей.

Но при использовании любого метода из описанных выше, не всегда подобранные композиции интересны пользователям в момент прослушивания. Может случиться ситуация, когда пользователь хочет изменить жанр. В таком случае музыкальным площадкам потребуется постепенно менять подбираемую музыку, учитывая предыдущие интересы пользователя. Это приводит к тому, что пользователь может долго искать подходящие треки, которые ему будут интересны в данный момент.

Чтобы не полагаться исключительно на историю прослушивания и поисковые запросы пользователя, был разработан алгоритм для анализа эмоций человек в конкретный момент времени. Этот подход позволит выбирать музыку, которая должна наиболее точно соответствовать настроению пользователя в момент прослушивания.

Основанный на анализе эмоционального состояния человека, алгоритм может быть реализован с использованием сверточных нейронных сетей для поиска лица на фотографии и его анализа на соответствие определенной эмоции.

## **Постановка задачи**

Целью выпускной квалификационной работы является разработка веб-приложения для прослушивания музыки с использованием нейронных сетей, анализирующих эмоции человека по фотографии его лица.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) провести обзор аналогов и научной литературы;
- 2) изучить методы определения эмоций человека по фотографии с использованием нейросетевых технологий;
- 3) осуществить сбор и предобработку данных для обучения;
- 4) разработать нейронную сеть и оценить результаты ее работы;
- 5) реализовать веб-приложение для прослушивания музыки;
- 6) провести тестирование веб-приложения.

## **Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 40 страниц, объем списка литературы – 36 источников.

В первой главе описываются теоретические сведения о предметной области и осуществляется обзор существующих аналогов. Рассматриваются различные подходы к решению задачи определения эмоций по фотографии лица человека. Также проводится сравнительный анализ аналогов для выявления существующих проблем и способов их решения. Рассматриваются существующие алгоритмы подборки музыки, используемые современными музыкальными площадками на данный момент.

Во второй главе представлены функциональные и нефункциональные требования к разрабатываемому веб-приложению. Спроектированы диаграммы вариантов использования и компонентов системы, а также приведено описание данных диаграмм. Также описан дизайн интерфейса для веб-

приложения и логика получения данных от серверной части. Помимо получения данных, описана работа музыкального плеера на странице прослушивания музыки.

В третьей главе описываются концепция сверточных нейронных сетей и топологии нейронных сетей, которые могут использоваться для решения задачи определения эмоций по фотографии лица. Рассматриваются процессы предобработки данных, которые необходимы для повышения качества модели и улучшения ее способности к обобщению. Описано создание и обучение сверточной нейронной сети и используемые в данном процессе технологии и методы, включая выбор оптимальных алгоритмов и техник регуляризации для предотвращения переобучения. Также представлены результаты подборки гиперпараметров нейронной сети для поиска наилучшей конфигурации для дальнейшего обучения.

В четвертой главе описано тестирование полученного веб-приложения и нейронной сети для определения эмоций человека по фотографии. Приведены результаты тестирования на различных наборах данных. Оценивается точность определения эмоций, их стабильность и производительность системы при различных условиях эксплуатации.

В заключении описываются результаты выполненной работы, возможные направления дальнейших исследований и перспективы веб-приложения.



## **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

### **Предметная область проекта**

Подборка музыкальных композиций – важная часть работы всех музыкальных приложений и сайтов таких как: «Apple Music», «Яндекс Музыка», «Boom», «Spotify», «VK music», «Deezer», «SoundCloud», «YouTube music» и другие. От качества работы методов подборки зависит продолжительность использования приложения пользователем. Большинство современных музыкальных сервисов используют методы, основанные на ранее прослушанной музыке или истории поисковых запросов.

Сервисы имеют огромное количество данных о пользователе, но все они основаны на его предыдущих поисковых запросах. Исходя из данного подхода, все современные площадки выстраивают линию интересов пользователя, но не его настроение в конкретный момент времени. Чтобы сразу понять настроение им требуются вводные данные, а именно история прослушивания или поиска. Но пользователь может долго искать подходящие треки из-за чего качество такого подхода сильно снижается.

Реализация веб-приложения, для подборки и прослушивания музыки с использованием алгоритма, определяющего эмоции человека [7] по фотографии его лица позволит производить более точный и быстрый анализ предпочтений пользователя в данный момент времени. При данном подходе вводными данными является фотография, а не история прослушивания, что заметно ускоряет поиск подходящей музыки.

### **1.1. Сравнительный анализ аналогов**

При анализе аналогичных проектов было обнаружено, что на данный момент не существует подобных решений, имеющих полноценную реализацию интерфейса для пользователя [6]. В большей части из них представлены только методы и нет возможности для прослушивания музыки. Исходя из теоретического материала были выделены следующие преимущества и недостатки (таблица 1).

Таблица 1 – Преимущества и недостатки аналогичных систем

Система	Преимущества	Недостатки
S Metilda Florence, M Uma «Emotional Detection and Music Recommendation System based on User Facial Expression»	Получение фотографии лица реализовано при помощи веб-камеры в режиме онлайн	Определяются только две эмоции: веселье и грусть.
«Music Recommendation based on Facial Emotion Recognition»	Определяется семь видов эмоций.	Подбираемая музыка разбита только на четыре эмоции из семи.
«EMUSE – AN EMOTION BASED MUSIC RECOMMENDATION SYSTEM» A. Phaneendra, Madhusmitha Muduli, Siri Lakshmi Reddy 213	Определяется шесть видом эмоций. Имеется интерфейс пользователя.	Исходя из графиков, модель является переобученной и имеет низкую точность определения.
Kavyashree, Bhargavi, Chinmayi, Kanchana, Sudhir Srinivas «ML approach for musical therapy using facial expression»	Получении фотографии с веб-камеры. Сверточная сеть имеет топологию ResNet 50.	Определяются только три вида эмоций: веселье, злость, грусть.

Подборка музыки с использованием искусственного интеллекта для анализа фотографии пользователя и определения эмоций [12] не используется на современных музыкальных площадках. В большей степени это связано с малой применимостью данного метода из-за сложности в его реализации.

Однако разработка систем, реализующих данный подход позволит музыкальным сервисам более точно составлять плейлисты для каждого пользователя, тем самым повышая общий уровень удовлетворенности использования сервиса.

Например, если на фотографии человек выглядит счастливым и улыбается, система может подобрать музыку с высокими тембрами, мажорным звучанием и быстрым темпом, что создаст ощущение радости и бодрости. А если пользователь имеет грустные эмоции, то ему подойдет минорное звучание с медленным темпом. Основываясь на звучании музыки для пользователя будут подбираться композиции по настроению. Тем самым, пользователь сможет найти новые и интересные для него треки.

## **1.2. Алгоритм определения эмоций по фотографии**

Для анализа эмоций человека [26] распространены сверточные нейронные сети. На сегодняшний день существует множество готовых решений, предназначенных для работы с выражением лица человека: «Emotient», «Affectiva», «EmoVu», «Kairos», «Project Oxford by Microsoft», «Face Reader by Noldus» и другие, причем большая часть сетей являются сверточными [4]. Это объясняется тем, что полносвязные нейронные сети не масштабируются должным образом, особенно в случае больших изображений [13], тем самым классические решения сильно уступают сверточным нейронным сетям.

Существуют несколько подходов к распознаванию эмоций на фотографиях при помощи машинного обучения, в том числе:

- 1) классификация эмоций: распознавание на основе классификации фотографий по основным эмоциям: радость, грусть, злость, страх и другие;
- 2) распознавание черт лица: определение эмоций по геометрическим параметрам лица, таким как лицевые мимики, выражения глаз и рта;
- 3) распознавание сочетаний: анализ фотографии, который определяет целостные композиты, которые описывают различные эмоциональные состояния.

Важную роль в создании нейронных сетей играют наборы данных для обучения. Для обучения определения и классификации эмоций наиболее часто используют следующие наборы данных: «CK++», «CARER», «ISEAR», «EmotionLines», «EmoBank».

## **1.3. Алгоритм подборки музыки**

Для подборки музыки на основании данных, полученных от нейронной сети, определяющей эмоции, разработан алгоритм, который учитывает корреляцию параметров музыкальной композиции и настроение человека

определяет подходящие музыкальные композиции. Исходя из набора данных «Spotify Dataset» [9], определены параметры музыкальных композиций, который определяют эмоциональную принадлежность трека.

Важным фактором является мажорное или минорное звучание, а также темп и тембр. Важно отметить, что музыка имеет различные жанры, что усложняет процесс подборки. При подборке учитываются все эти факторы в совокупности, что позволяет выделять некоторые правила.

К примеру минорное звучание и медленный темп передают эмоции печали, в то время как мажорное звучание и медленный темп несут состояние покоя. Минорное звучание с высоким темпом можно отнести к эмоции гнева, а характер музыки можно отнести к напряженно-драматическому. Мажорная окраска и быстрый темп будут передавать уже эмоции радости. Таким образом выделяются некоторые правила поиска музыки.

В данной работе рассматривается упрощенная система подбора музыки, основанная исключительно на параметрах музыкальной композиции, без учета их жанровой или иной принадлежности. К примеру, если нейронная сеть определяет счастливые эмоции человека, то преимущественно будут подбираться музыкальные композиции с большим параметром энергичности, чем для человека, имеющего более грустное эмоциональное состояние.

### **Выводы по первой главе**

В данной главе были рассмотрены предметная область, аналогичные приложения, а также проведен обзор алгоритмов для определения эмоций человека по фотографии лица.

Было определено, что в данный момент алгоритмы подборки музыки основаны на истории прослушивания. Новый алгоритм позволит ориентироваться на состояние пользователя в данный момент, что в теории позволит улучшить качество подборки музыки.

## **2. ПРОЕКТИРОВАНИЕ**

### **2.1. Определение требований**

Для прослушивания музыки, подобранной при помощи нейронной сети по фотографии лица человека будет разработано соответствующее веб-приложение. На текущий момент веб-приложения обретают большую популярность из-за их возможностей. В ходе проектирования приложения были определены функциональные и нефункциональные требования к системе.

#### **Функциональные требования**

К разрабатываемой системе имеются следующие функциональные требования.

1. Система должна распознавать 4 вида эмоций: веселье, грусть, нейтральность, злость.
2. Система должна предоставлять доступ к прослушиванию подобранной музыки.
3. Система должна предоставлять возможность загружать фотографию.

#### **Нефункциональные требования**

К разрабатываемой системе имеются следующие нефункциональные требования.

1. Система должна иметь возможность загружать файлы с расширениями jpeg и png.
2. Система должна иметь возможность запуска внутри Docker [25] контейнеров.
3. Интерфейс должен быть разработан с использованием React [15].

Архитектуру веб-приложения можно разделить на несколько компонентов, взаимодействующих между собой для реализации требуемого функционала. Диаграмма вариантов использования веб-приложения представлена на рисунке 1.

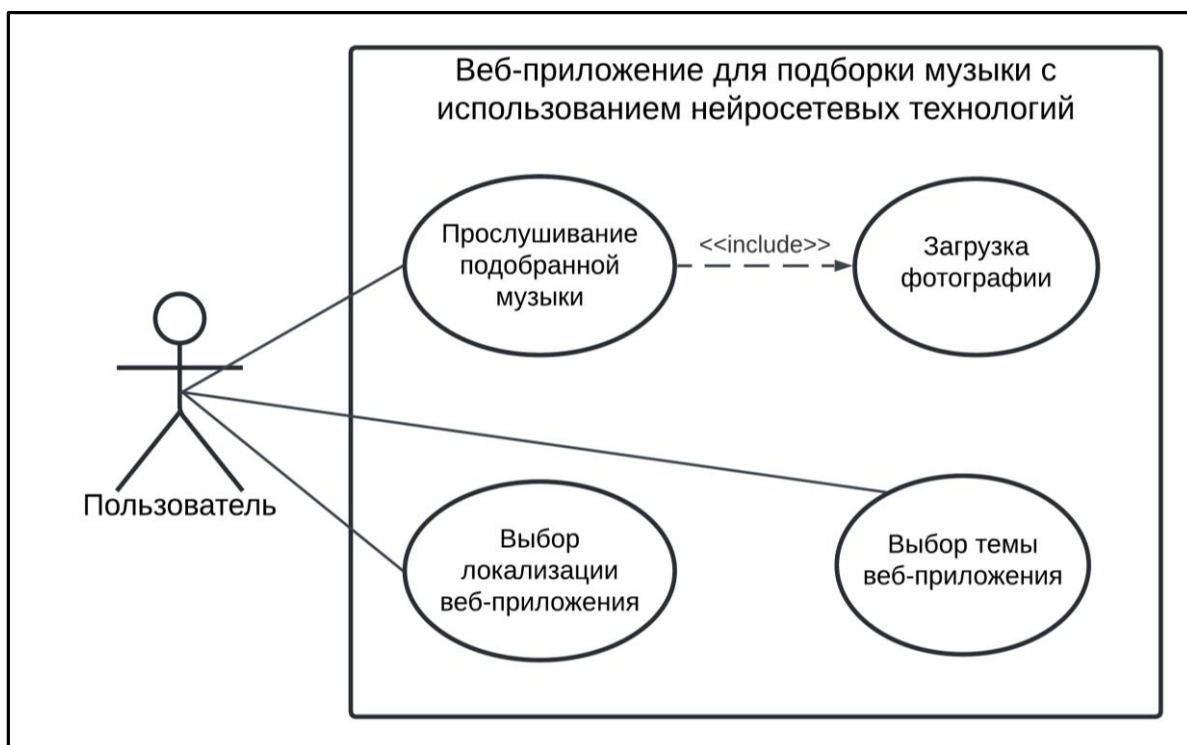


Рисунок 1 – Диаграмма вариантов использования веб-приложения

Актером данной системы является пользователь. Ему доступны следующие варианты использования системы.

1. Прослушивание музыки подобранной системой для пользователя.
2. Загрузка фотографии для получения музыкальных композиций.

Всю серверную часть можно изобразить в диаграмме компонентов (рисунок 2), которая показывает, как будет происходить взаимодействие между составляющими веб-приложения.

На диаграмме можно увидеть основные модули, работа которых обеспечивает обработку изображения, анализ эмоционального состояния человека на фотографии, а также подбор и отправка музыкальных треков для пользователя. Важную роль в системе играет Docker [32], так как он позволяет упрощать развертывание и управление масштабированием приложений.

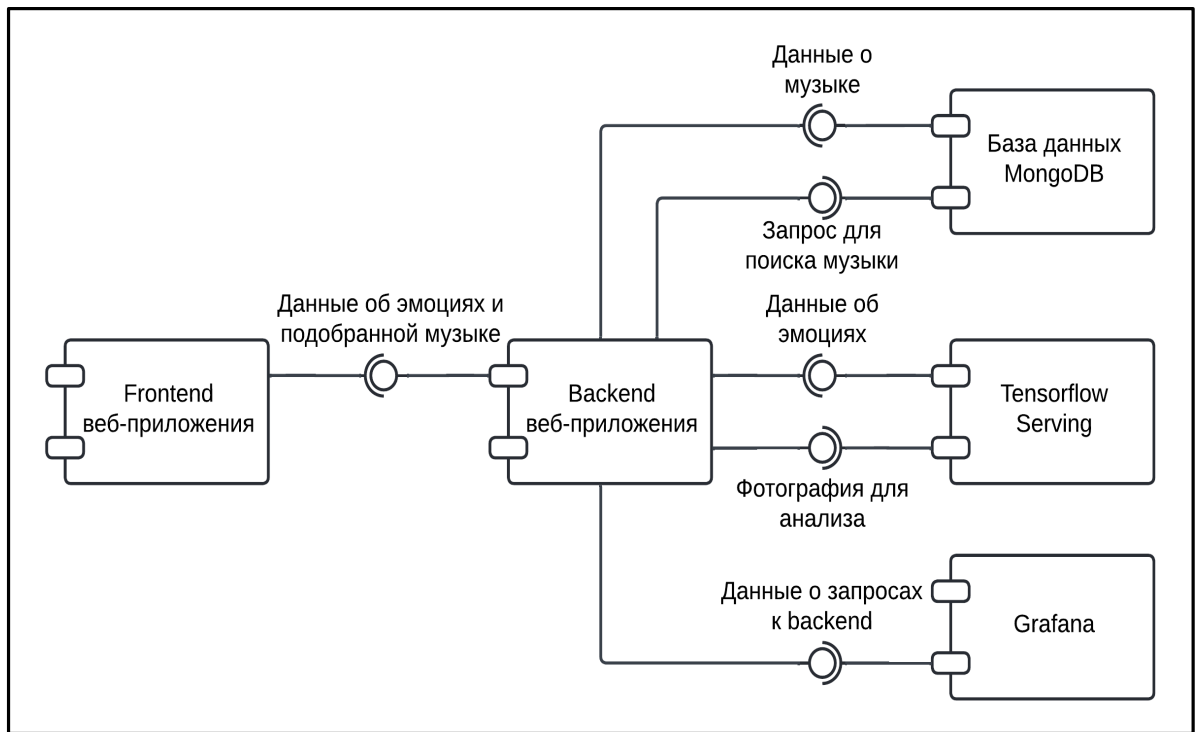


Рисунок 2 – Диаграмма компонентов

Данная диаграмма включает следующие программные модули.

1. Frontend веб-приложения – отвечает за отображение данных для пользователя, воспроизведение музыкальных композиций и отправку запросов к backend.

2. Backend веб-приложения – получаем запросы от frontend, проверяет корректность полученных данных и отправляет запросы на получение данных к модулям Tensorflow Serving и MongoDB, а также отправляет информацию о своей работе Grafana.

3. MongoDB – хранит информацию о музыкальных композициях и их размеченных параметрах, которые будут использоваться при поиске подходящих треков.

4. Tensorflow Serving – хранит внутри себя сохраненную сверточную нейронную сеть для определения эмоций человека по фотографии его лица и предоставляет REST API для получения данных.

5. Grafana – визуализирует информацию о работе компонента «Backend веб-приложения».

## 2.2. Проектирование веб-приложения

### Интерфейс веб-приложения

Проектирование интерфейса веб-приложения является критически важной фазой в процессе разработки. Это не только определяет визуальное впечатление, которое пользователи получают от использования приложения, но и то, насколько они смогут легко и эффективно его использовать.

Интерфейс веб-приложения состоит из двух страниц. Главная страница, макет которой представлена на рисунке 3, позволяет пользователю ознакомиться с основной информацией о функциях веб-приложения и загрузить фотографию. Также предусмотрена светлая и темная тема и поддержка английской локализации приложения.

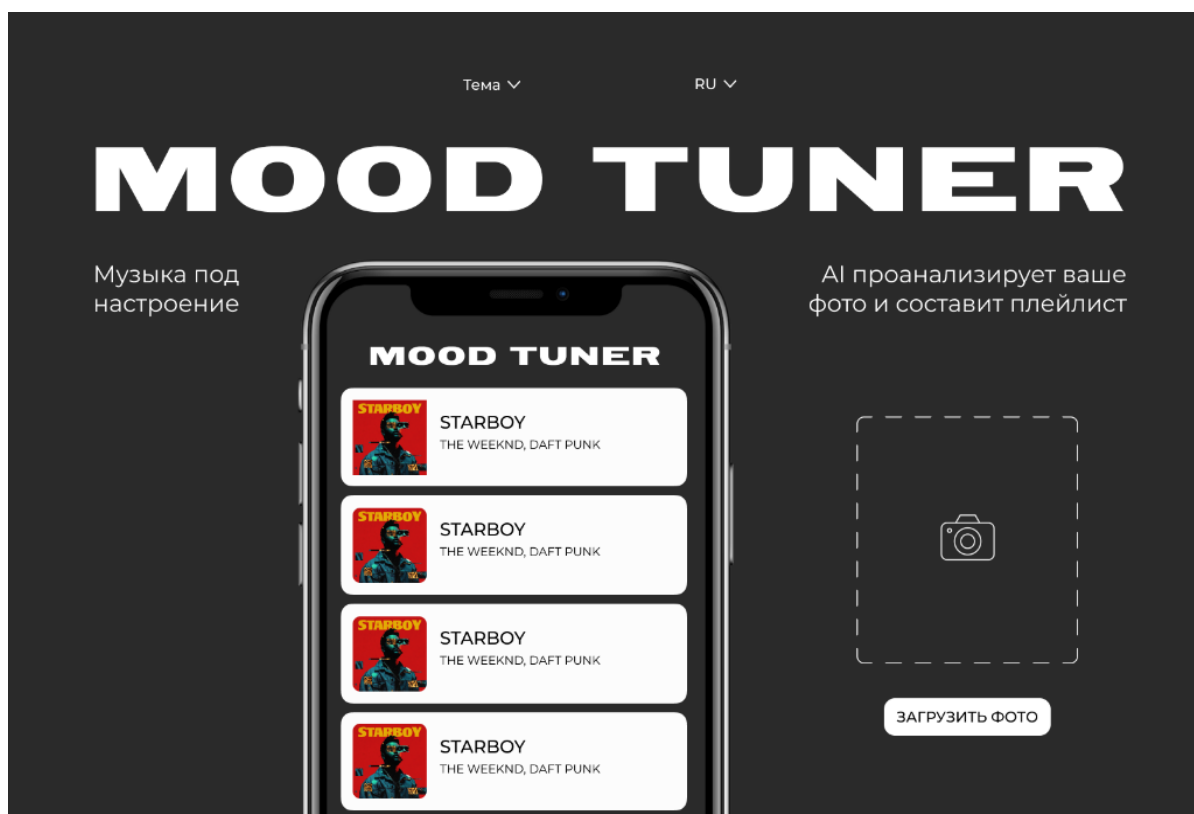


Рисунок 3 – Макет главной страницы веб-приложения

После загрузки фотографии пользователь попадает на страницу с музыкальными композициями (рисунок 4), где он может прослушать подобранную музыку, посмотреть информацию об эмоциях человека на загруженном фото или загрузить другое. Если лицо на загруженной фотографии



не было найдено, то для пользователя будет отображено соответствующее диалоговое окно.

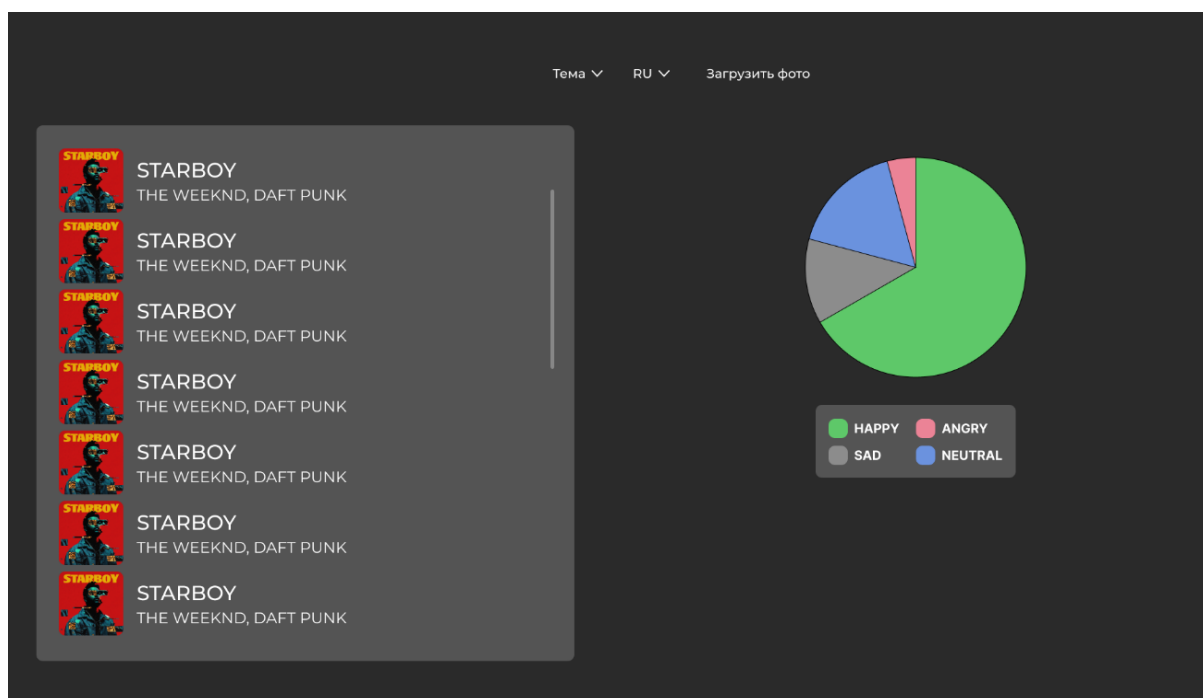


Рисунок 4 – Макет страницы веб-приложения с музыкальными треками

Если пользователь хочет прослушать музыкальную композицию, то он просто нажимает на понравившийся вариант. В нижней части страницы открывается музыкальный плеер (рисунок 5), в котором для пользователя отображаются основные кнопки для управления.

Пользователь может менять текущий момент песни, а также ставить на паузу и возобновлять прослушивания. После окончания текущей музыкальной композиции начинает проигрываться следующая. В случае если была прослушана последняя музыкальная композиция из списка, то воспроизведение музыкальных композиций начнется с начала списка.

Пользователь имеет возможность переключить композицию, нажав на соответствующую кнопку или выбрав трек из списка. После переключения текущая композиция останавливается и начинает загружаться новая. После загрузки она автоматически воспроизводится.

Если пользователь загружает новую фотографию во время прослушивания, то текущая проигрываемая музыкальная композиция останавливается.

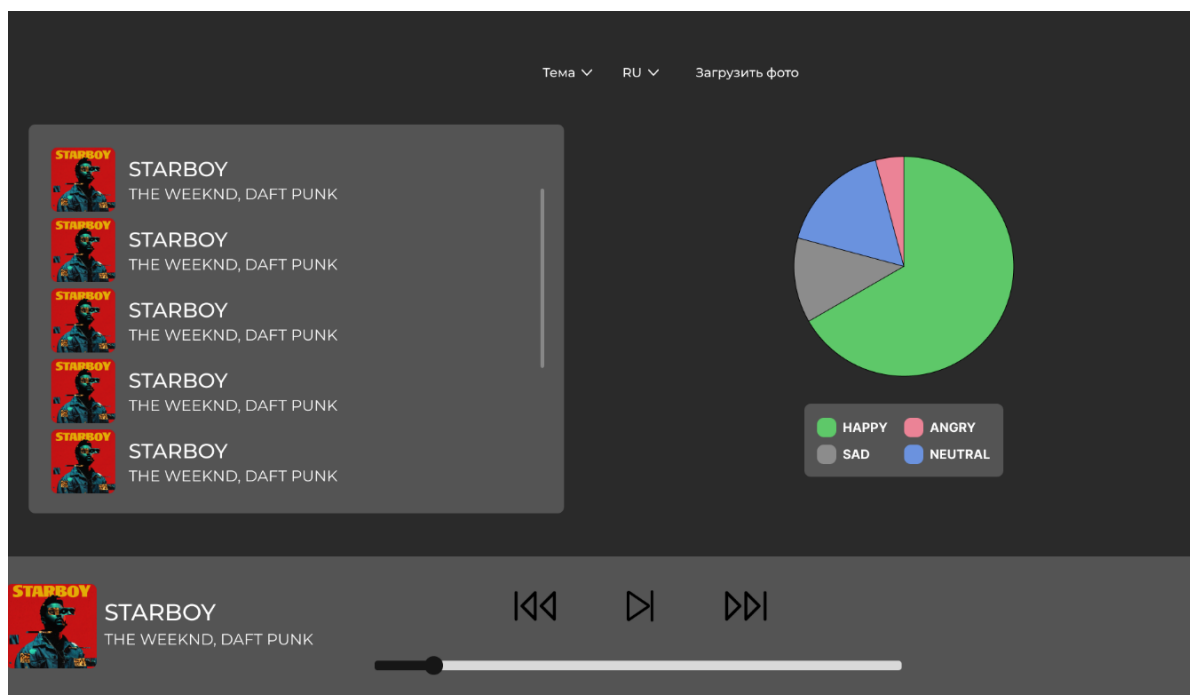


Рисунок 5 – Макет плеера

### Серверная часть веб-приложения

Второй важной частью веб-приложения является его бизнес логика. Для реализации был создан сервер с использованием FastAPI [36], который отвечает за предобработку, анализ и выдачу результатов пользователю.

На этапе предобработки сервер получает фотографию и при помощи нейронной сети Face recognition [29] определяется наличие лиц на фотографии.

После удачного определения лица, сервер выделяет лицо из фотографии и отправляет его в сверточную нейронную сеть, развернутую в Docker [32] контейнере, для определения эмоций.

При помощи полученных от нейронной сети результатов происходит поиск подходящей музыки в базе данных и ответ пользователю с музыкальными композициями.

Помимо получения фотографий сервер имеет конечные точки для отправки пользователю обложки музыкального трека и самого файла с треком. Данные файлы хранятся в S3 хранилище, куда сервер отправляет запросы на получение конкретных файлов. Данный подход позволяет значительно уменьшить размер backend части веб-приложения, что благополучно сказывается при развертывании сервера на виртуальной машине.

Все запросы к конечным точкам сервера логируются при помощи Prometheus версии 7.0.0 [34] и отображаются в Grafana [31] для удобного просмотра и анализа. Также предусмотрены инструменты для логирования данных внутри FastAPI приложения, что является полезным при отладке.

### **2.3. Проектирование нейронной сети**

Проектирование нейронной сети начинается с выбора топологии, которая определяет функцию сети. Для решения задач анализа фотографии используют сверточные нейронные сети.

Сверточная нейронная сеть (с англ. Convolutional Neural Network) – глубокая искусственная нейронная сеть с прямой связью. В ней сигналы между нейронами передаются с одним направлением. Выход последующего нейрона не передается на входы предыдущих.

Основная идея сверточных нейронных сетей заключается в исследовании Д. Хьюбела и Т. Визера. В 1962 г. ученые подтвердили свою гипотезу о том, что зрительная кора головного мозга чувствительна к определенным областям поля зрения. Было продемонстрировано, что отдельные нейроны в мозге начинают активизироваться только при наличии краев определенной ориентации.

Простейшая нейронная сеть состоит из следующих слоев:

- 1) входной слой – принимает данные о изображении в нормализованном формате от 0 до 1;

2) сверточный слой – состоит из матрицы свертки, которая проходит по изображению и применяет к нему операцию свертки, получая на выходе новую карту признаков;

3) полносвязный слой – проводит классификацию объектов, полученных на карте признаков;

4) слой субдискретизации (подвыборки) – позволяет снижать риск переобучения модели путем уменьшения размерности карт признаков;

5) выходной слой – используя функцию активации, предсказывает метки класса для изображения.

### Алгоритм свертки

Операция свертки проводится при помощи специального ядра, называемого фильтром. Фильтр является матрицей весовых коэффициентов, значения которых устанавливаются и изменяются в процессе обучения модели. Количество фильтров и карт признаков определяется задачей. Увеличение числа фильтров повышает эффективность модели в распознавании образов, но требует больше вычислительных ресурсов.

Сканирование карт фильтром происходит за счет перемещения окна определенного размера по всей области карты (рисунок 6), полученной с предыдущего слоя. Наиболее эффективными размерами окна варьируется от 3x3 до 7x7, так как меньшие размеры не выделяют достаточное количество признаков, а большие размеры уменьшают скорость обучения из-за увеличения количества нейронных связей.

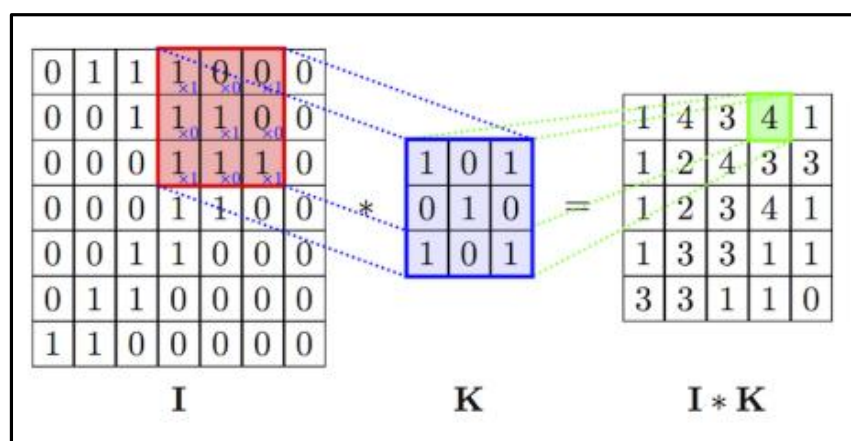


Рисунок 6 – Операция свертки

На этапе начала обучения модели значения карт сверточного слоя устанавливаются на ноль, а веса фильтров случайно задаются в диапазоне от  $[-0,5; 0,5]$ . В процессе обучения весовые коэффициенты изменяются для выделения характерных признаков объектов на изображении. Ядро перемещается по предыдущей карте, выполняя операцию свертки.

На каждом этапе фильтр покрывает область на карте, размерность которой совпадает с размерностью фильтра. Свертка осуществляется через поэлементное умножение элементов матрицы фильтра на элементы матрицы рассматриваемой области и последующее суммирование этих произведений. Результат записывается в соответствующий элемент выходной карты

### **Топология нейронной сети**

Существуют различные топологии сверточных нейронных сетей для обработки фотографии, музыки или видео. Наиболее популярные из них:

- 1) AlexNet;
- 2) ResNet;
- 3) VGGNet;
- 4) GoogleNet;
- 5) LeNet.

Каждая топология имеет свои особенности и недостатки. Проанализировав топологии, можно выделить «ResNet» как наиболее подходящий для задачи определения эмоций человека по фотографии его лица [12].

Ключевым фактором является то, что в данной топологии применен подход остаточного обучения, визуализация которого представлена на рисунке 7, что заметно отличает его от других топологий. Большинство моделей используют данную топологию для решения самых различным задач в области компьютерного зрения.

Особенности других топологий также заслуживают внимания. В частности, AlexNet, одна из первых глубоких сверточных нейронных сетей, подняла точность классификации изображений на новый уровень благодаря использованию нескольких сверточных слоев и нормализации.

VGGNet известна своей простой архитектурой, состоящей из небольших сверточных фильтров, что позволяет легче интерпретировать структуру сети. GoogleNet, или Inception-сеть, отличается уникальным дизайном модулей Inception, которые делают возможным захват характеристик на нескольких масштабах одновременно.

LeNet, одна из самых ранних сетей, предназначенных для распознавания рукописных цифр, заложила основы для дальнейшего развития в области сверточных нейронных сетей

Каждая из этих топологий имеет свои уникальные архитектурные элементы и сферы применения, что делает выбор подходящей архитектуры важной задачей в зависимости от конкретных требований и условий задачи.

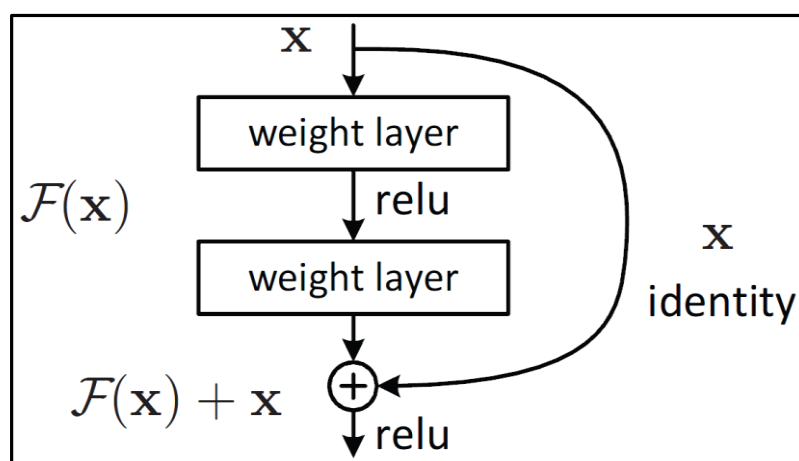


Рисунок 7 – Пример работы остаточного обучения

При обучении, глубокие нейронные сети имеют особенность быстрого увеличения точности на старте и заметного ухудшения в конце. Алгоритм обратного распространения ошибки (backpropagation) представляет собой основной метод для тренировки нейронных сетей позволяющий избежать данной проблемы.

После того как входные данные пройдут через сеть, вычисляется ошибка между фактическими и ожидаемыми результатами. Это значение ошибки затем проходит обратно через сеть для корректировки весов и смещений нейронов.

Шаги алгоритма обратного распространения ошибки описаны ниже.

1. Прямое распространение – входные данные проходят через сеть, каждый нейрон оценивает выходное значение на основе текущих весов и данных.

2. Вычисление ошибки – определяется разница между фактическими выходными данными и ожидаемыми результатами.

3. Обратное распространение ошибки – ошибка передается от последнего слоя к первому, причем каждый нейрон рассчитывает свою ошибку и передает ее предыдущему слою.

4. Обновление весов и смещений – после того как ошибка прошла через сеть, веса и смещения каждого нейрона корректируются на основе значения ошибки и заданной скорости обучения.

5. Повторение – эти четыре шага повторяются для каждой эпохи обучения до достижения желаемой точности работы сети.

Хотя алгоритм обратного распространения ошибки крайне эффективен для обучения нейронных сетей, он не лишен недостатков, таких как риск затухания или увеличения градиента и переобучение модели. Иногда это связано с тем, что не всегда нейронная сеть хорошо оптимизирована. Градиент представляет собой вектор частных производных функции ошибки по каждому весу и смещению в нейронной сети. Он также показывает направление наискорейшего уменьшения ошибки и используется для корректировки весов и смещений в процессе обучения.

Проблема затухания или разрастания градиента является значительным препятствием при обучении сети. Это может привести к тому, что после обновления значения весов и смещений станут слишком малыми или наоборот, слишком большими, что усложнит процесс обучения [3]. Возможные решения этой проблемы включают регуляризацию, оптимизацию скорости обучения, использование различных функций активации или изменение архитектуры нейронной сети

Проектирование нейронной сети включает в себя множество технических вызовов и сложностей. Прежде всего, необходимо обеспечить баланс между глубиной сети и вычислительными ресурсами, так как слишком сложные сети требуют значительных вычислительных затрат и могут привести к долгому времени тренировки [2]. Еще одна сложность связана с выбором и сбором качественных данных для обучения, так как некорректные или нерелевантные данные могут привести к неправильному обучению модели. Кроме того, проблема переобучения остается актуальной, так как сеть может слишком точно подстроиться под обучающий набор данных, теряя способность к корректному анализу новых данных [1].

Для получения наилучших результатов потребуется находить баланс между количеством слоев и их видом. Немаловажным является подбор гиперпараметров нейронной сети. Чем больше различных вариантов топологии будет рассмотрено, тем точнее получится нейронная сеть. Таким образом для поиска наилучшего решения требуется перебирать множество вариантов, что в достаточной мере усложняет задачу и требует использования мощных вычислительных ресурсов.

### **Выводы по второй главе**

В данном разделе были определены функциональные и нефункциональные требования к разрабатываемому веб-приложению. Были разработаны диаграммы вариантов использования и компонентов системы, а также спроектирован интерфейс двух страниц и серверная часть приложения.

В диаграмме компонентов были приведены основные микросервисы и составляющие серверной части веб-приложения, отвечающие за все этапы, начиная от получения фотографии от пользователя и заканчивая отправкой музыкальных композиций и информации об эмоциях на загруженном пользователем фото.



### 3. РЕАЛИЗАЦИЯ

#### 3.1. Программные средства реализации

Для реализации веб-приложения для подборки музыки с использованием нейросетевых технологий был использован язык программирования TypeScript версии 5.3.3 [35] и Python версии 3.10 [24]. Веб-интерфейс приложения разработан с помощью библиотеки React [15] для JavaScript. Серверная часть разработана при помощи Python библиотеки FastAPI [36]. Для написания frontend составляющей была использована среда разработки WebStorm, а для backend части PyCharm [8].

В процессе разработки frontend были использованы следующие библиотеки и фреймворки.

1. Material UI (7.2.96) [22] – для использования готовых компонентов.
2. Redux (4.2.1) – для управления состоянием пользователя.
3. FramerMotion (10.16.4) [21] – для реализации анимации.

При разработке backend использовались следующие библиотеки для Python.

1. Библиотека motor (3.3.2) [7] – для синхронного подключения к базе данных MongoDB.
2. Библиотека Pydantic (2.5.1) [18] – для построения и валидации структур данных.

Для разработки нейронной сети был выбран высокоуровневый язык программирования Python версии 3.10. Разработка велась в веб-приложении с открытым исходным кодом Jupiter Notebook [5].

Для работы с нейронными сетями были использованы следующие библиотеки: NumPy версии 1.24.3 [23], Tensorflow версии 2.10.0 [17], Keras версии 2.10.0 [16] и OpenCV версии 4.9.0 [20]. Для отслеживания хода обучения нейронной сети использовались программные средства TensorBoard версии 2.10.0 [10] и W&V версии 0.15.2 [11].

Tensorflow предоставляет обширное число различных слоев для построения сверточных нейронных сетей, а также помогает настраивать множество параметров для анализа обучения нейронных сетей.

Для анализа использовался веб-сервис W&V позволяющий выводить любую статистику по данным об обучении нейронной сети, сравнивать процесс обучения различных конфигураций и удобно вести мониторинг процесса обучения.

## **3.2. Разработка веб-приложения**

### **Реализация frontend части веб-приложения**

При разработке интерфейса веб-приложения на React, основная задача заключалась в создании пользовательского интерфейса, который бы отражал дизайн и функциональные требования проекта. Был использован компонентный подход, позволяющий создавать переиспользуемые элементы интерфейса для обеспечения легкости сопровождения и масштабирования кода. Именно он является основой в React.

Для эффективного управления состоянием приложения, включая данные пользователя, применяется React Redux. Сначала определяются «reducers», которые определяют изменения состояния приложения в ответ на действия пользователя или события. Затем с помощью библиотеки Redux Toolkit создается и управляется хранилище Redux.

Для загрузки фотографии добавлен компонент «input» с настройкой, позволяющий загружать только фотографии. После загрузки файл отправляется на сервер и обрабатывается. Полученный ответ записывается в Redux, и пользователь перенаправляется на страницу с музыкальными композициями (рисунок 3). На данной странице он может прослушать интересующие его музыкальные композиции или загрузить новое фото.

Прослушивание музыки осуществляется при помощи хука «useAudio», который получает файл в формате «.mp3» и предоставляет

функции для запуска и остановки музыкальной композиции. Управление состоянием прослушивания осуществляется при помощи плеера (рисунок 4), который также отображает основную информацию о прослушиваемом треке.

### **Реализация backend части веб-приложения**

Серверная часть реализована при помощи библиотеки FastAPI для Python и развернута в Docker контейнере. Данный подход позволяет быстро разворачивать сервер с минимальной настройкой и связывать между собой различные сервисы, сохраняя при этом дисковое пространство.

Сервер реализует конечные точки для получения данных по REST API. Каждая точка производит валидацию входных данных и в асинхронном режиме обрабатывает запрос.

Так как для хранения фотографий музыкальных композиций и сами исходные файлы этих композиций используется S3 хранилище данных, то на стороне серверной части реализовано обращение к S3 хранилищу при помощи библиотеки boto3 версии 1.34.44 [19]. Когда к серверу поступает запрос на получение фотографии, то имея токены доступа, сервер отправляет запрос к S3 хранилищу, откуда получает файл и перенаправляет его на frontend.

При получении фотографии, сервер предобработывает полученное фото, переводя его в NumPy массив и запускает поиск лиц при помощи нейронной сети Face Recognition. Если лица найдены, то на frontend отправляет соответствующий ответ. В противном случае создается запрос к Tensorflow Serving [17], развернутом в Docker контейнере. Полученный ответ используется для поиска музыкальных композиций в базе данных MongoDB. Информация о треках и эмоциях на фотографии отправляется пользователю.

### 3.3. Подготовка и предобработка набора данных

Для обучения нейронной сети были выбраны следующие наборы данных: FER2013 и CK+. Их выбор обоснован тем, что они имеют схожие наборы эмоций и содержат только лица людей без посторонних предметов на заднем плане. Особенностью данных наборов является то, что они содержат лица людей разной этнической расы. Также присутствуют лица людей под разным углом обзора, что позволит обучить нейронную сеть работать с сложными фотографиями лица.

Для качественного обучения нейронной сети, детектирующей эмоции по лицу человека, критически важно, чтобы в наборе данных для обучения не было фотографий с множеством лишних объектов (рисунок 8). Качества фотографии лица также имеет большое значение, недопустимы фотографии лиц в профиль.

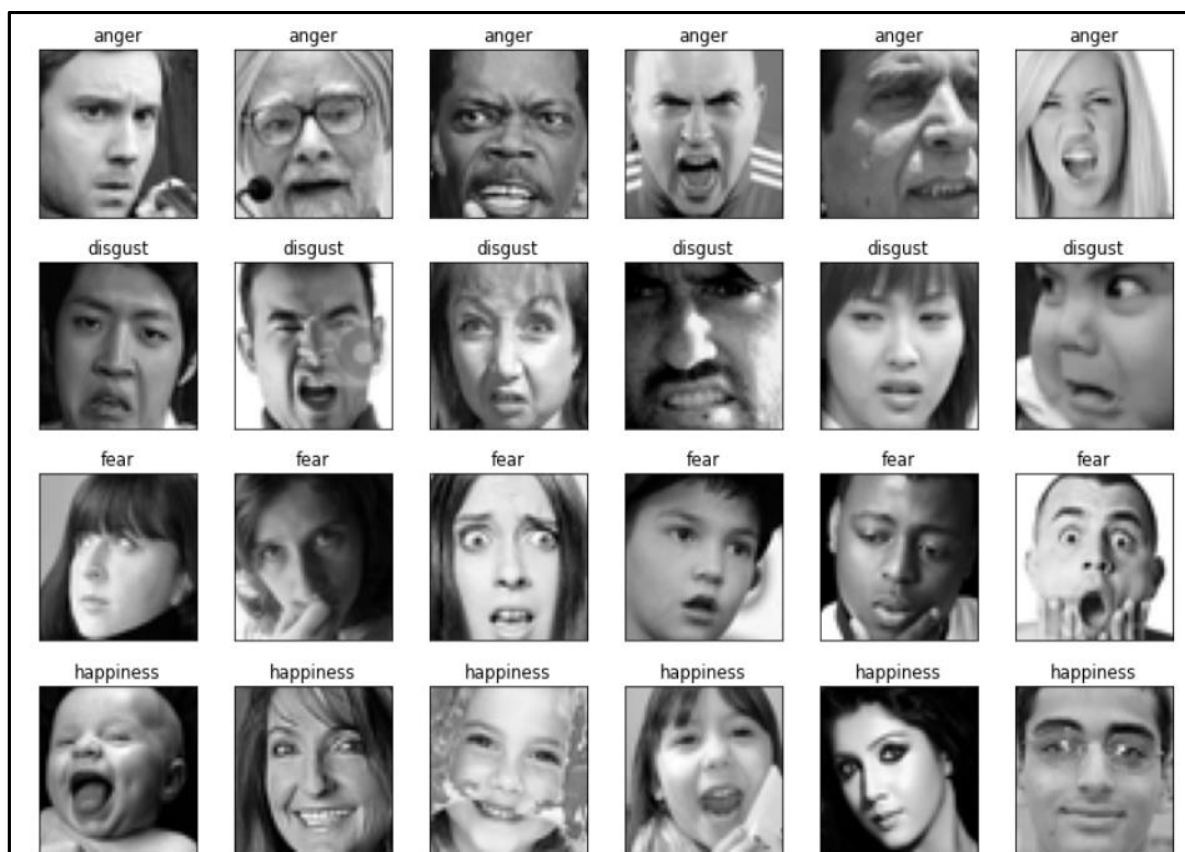


Рисунок 8 – Пример фотографий в наборе данных

Некачественные данные заметно ухудшают скорость обучения и точность определения эмоций. Все изображения имеют размер 60 на 60 пикселей, так как входной слой нейронной сети имеет размерность 60x60x3, что означает что он принимает изображения 60 на 60 пикселей и 3 канала цвета. Также из датасета были убраны эмоции страха, удивления и смущения, так как они путались с другими эмоциями сильно ухудшая качество работы модели.

### 3.4. Построение и обучение нейронной сети

#### Нейронная сеть для определения эмоции

Для определения эмоций была выбрана топология «ResNet50», позволяющую не терять в точности определения эмоции, при этом улучшая распознавание различных типов лиц (рисунок 9).

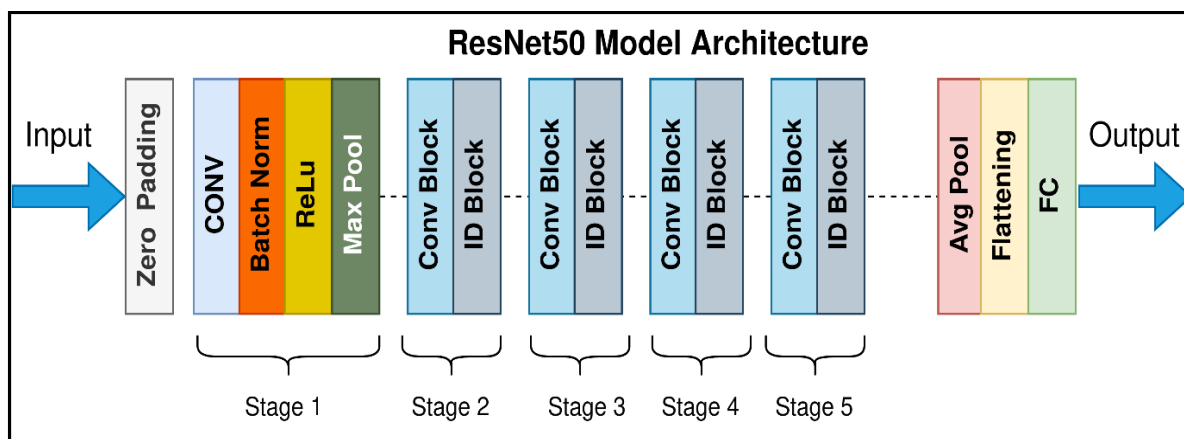


Рисунок 9 – Топология нейронной сети ResNet50

На приведенном выше рисунке схематично изображена топология нейронной сети, которая будет является основой сверточной нейронной сети для решения задачи определения эмоционального отклика человека по его фотографии.

В данной топологии сверточная нейронная сеть состоит из нескольких повторяющихся слоев. Слои свертки позволяют выделять паттерны на фотографии и уменьшать общий размер данных. Для улучшения качества пат-

тернов используют слои подвыборки и пуллинга. Для предотвращения переобучения используются dropout слои, которые случайным образом выключают несколько нейронов, тем самым, не давая сети переобучаться. На выходе чистая ResNet50 имеет 1000 классов, которые потребуется свести к 4-м. Для этого к выходу ResNet50 добавлено несколько слоев подвыборки и нормализации батчей.

После выбора топологии и построения нейронной сети необходимо перейти к процессу ее обучения и подборки гиперпараметров. Для повышения качества обучения использовалась аугментация датасета, что повысило устойчивость к наклону лица и фотографиям с различной перспективой.

Для подборки гиперпараметров была выбрана библиотека Keras Tuner [16], которая позволила быстро и удобно настроить диапазон для подбираемых значений и способ выбора этих значений. Также Keras-Tuner удобно интегрировать с W&V и TensorBoard для их визуализации (рисунок 10).

<input type="checkbox"/> <input checked="" type="radio"/> Name (12 visualized)	epoch/loss	epoch/val_loss	epoch/accuracy	epoch/val_accuracy
• <input checked="" type="radio"/> <input checked="" type="radio"/> Search_2024-04-1...	0.8911	1.107	0.8338	0.7911
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	1.046	1.133	0.7965	0.784
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.96	1.125	0.8212	0.7921
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.942	1.038	0.8267	0.8048
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.9259	1.092	0.8268	0.7913
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.9598	1.082	0.8211	0.7992
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.9754	1.101	0.824	0.794
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.8391	1.113	0.844	0.794
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.8971	1.103	0.8339	0.7952
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.9246	1.107	0.8333	0.7925
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	1.002	1.068	0.811	0.7945
• <input checked="" type="radio"/> <input type="radio"/> Search_2024-04-1...	0.9726	1.083	0.8195	0.7951

Рисунок 10 – Визуализация процесса обучения в W&V

Для достижения наилучших результатов число эпох требуется увеличить, но в силу ограниченности вычислительных мощностей было принято решение ограничиться 30 эпохами на каждой итерации подбора. Размер подвыборки (batch size) при обучении составлял 64, как наиболее оптимальный по количеству фотографий в каждой подвыборке. В результате были получены следующие результаты (рисунок 11).

Name (12\	epoch/val_loss	epoch/val_accuracy ▾	Dense_1	Dense_2	Dense_3	Dropout	Activatio
● Sear...	0.838	0.8848	32	30	26	0.55	elu
● Sear...	0.982	0.8492	32	32	24	0.55	elu
● Sear...	1.103	0.8252	32	32	32	0.5	elu
● Sear...	1.083	0.7951	64	32	16	0.5	relu
● Sear...	1.068	0.7945	128	64	32	0.6	relu
● Sear...	1.113	0.794	64	32	32	0.5	elu
● Sear...	1.101	0.794	64	32	32	0.5	softmax
● Sear...	1.107	0.7925	64	64	32	0.5	softmax
● Sear...	1.125	0.7921	38	36	24	0.55	elu
● Sear...	1.092	0.7913	40	32	28	0.55	elu
● Sear...	1.107	0.7911	40	36	32	0.55	elu
● Sear...	1.133	0.784	38	36	32	0.55	softmax

Рисунок 11 – Подбор гиперпараметров нейронной сети

Проанализировав полученные данные, было принято решение использовать наилучшую конфигурацию гиперпараметров при заданной топологии нейронной сети. Из-за ограниченности в 40 эпох выбор наилучшей конфигурации не гарантирует абсолютного преимущества, но может способствовать скорейшему росту точности.

Вторая часть обучения заключалась в обучении наилучших конфигураций на большем количестве эпох. При обучении размер подвыборки

(batch size) составил 64 изображения как наиболее оптимальный при заданном объеме датасета. Обучение проводилось в течении 80 эпох со средним временем обучения в 50 секунд на эпоху.

Результатом обучения является готовая к использованию модель нейронной сети. Сохранив в формате Tensorflow [4] данную модель, она была загружена в Tensorflow Serving, который был развернут в Docker [32], для использования из серверной части веб-приложения.

### **Нейронная сеть для определения лица на фотографии**

Определение лиц на фотографии производится при помощи готовой нейронной сети «Face recognition» [29]. Эта сеть предназначена для идентификации и анализа лиц, присутствующих на изображении, загруженном пользователем. Когда на вход сети поступает фотография, происходит ее обработка на нескольких уровнях, начиная с предварительной подготовки изображения и заканчивая построением модели, которая позволяет определить ключевые точки лица.

Результатом работы является массив с точками, определяющими возможное лицо на фотографии. Используя данные о точках на основном изображении выделяются лица и отправляются в нейронную сеть для определения эмоций на них.

### **Выводы по третьей главе**

В соответствии с заданием был проведен сбор данных, построение и обучение нейронной сети для определения эмоций человека по фотографии лица. В соответствии с макетом были разработаны два окна веб-приложения и реализована серверная часть для обработки фотографий пользователя и поиска подходящих музыкальных композиций.



## 4. ТЕСТИРОВАНИЕ

### Тестирование веб-приложения

Тестирование полученного веб-приложения состоит из функционального тестирования и оценка качества построенной нейронной сети. Для проверки работы веб-приложения было проведено функциональное тестирование для frontend и backend частей отдельно для определения возможных ошибок при работе веб-приложения. Наборы функциональных тестов веб-приложения представлены в таблицах 2 и 3.

Таблица 2 – Функциональное тестирование frontend части

№	Название теста	Входные данные	Ожидаемый результат
1	Загрузка фотографии	Пользователь открывает главную страницу веб-приложение и загружает фотографию	Фотографию успешно загружается, отправляется запрос к backend и приходит ответ со списком музыкальных композиций. Пользователь автоматически переходит на страницу прослушивания
2	Загрузка файла некорректного формата	Пользователь открывает главную страницу веб-приложение и загружает файл некорректного формата	Веб-приложение отображает пользователю сообщение о некорректном формате файла
3	Прослушивание музыкальной композиции	Пользователь загрузил фотографию, автоматически перешел на страницу с музыкой и нажал на музыкальную композицию	Веб-приложение воспроизводит музыкальную композицию
4	Изменение текущей воспроизводимой музыкальной композиции	Пользователь прослушивает музыкальную композицию в веб-приложении и нажимаем на кнопку воспроизведения следующей композиции	Веб-приложение завершает воспроизведение текущей музыкальной композиции и начинает воспроизводить следующую музыкальную композицию
5	Быстрое переключение музыкальных композиций	Пользователь загрузил фотографию и получил список музыкальных композиций. После получение списка он начинает быстро переключаться между ними	Пользователь будет прослушивать только последний выбранный из списка музыкальный трек. Также нет задержек в получении каждой выбранной музыкальной композиции из списка
6	Отключение серверной части во время прослушивания музыки	Пользователь прослушиваем музыкальную композицию.	Для пользователя отображается уведомление о том, что серверная часть недоступна и он не может продолжать прослушивания музыки.

Таблица 3 – Функциональное тестирование backend части

№	Название теста	Входные данные	Ожидаемый результат
1	Получение обложки музыкальной композиции	GET запрос к конечной точке сервера «/api/v1/photo?fileId=test»	Сервер отправляет в ответ на запрос файл с обложкой в формате «png»
2	Получение музыкальной композиции	GET запрос к конечной точке сервера «/api/v1/music?musicId=test»	Сервер отправляет в ответ на запрос файл в формате «mp3»
3	Загрузка фотографии для анализа эмоций на ней	POST запрос к конечной точке сервера «/api/v1/uploadPhoto», где в теле запроса содержится фотография с лицом человека	Сервер отправляет JSON с данными о эмоциях человека на фотографии и музыкальных композициях
4	Загрузка фотографии без лица	POST запрос к конечной точке сервера «/api/v1/uploadPhoto», где в теле запроса содержится фотография без лица человека	Сервер отправляет ответ с кодом 204 и сообщение о том, что на фотографии не найдено лицо человека
5	Просмотр логов в «Grafana»	Пользователь используя конечную точку «/grafana» авторизуется и открывает дашборд «FastAPI»	Сервер «Grafana» отображает пользователю информацию, получаемую от «FastAPI»

### Тестирование нейронной сети для определения эмоций

Для оценки качества нейронной сети была сформирована контрольная выборка, состоящая из фотографий лиц людей, не вошедших в обучающую выборку.

Нейронная сеть достигла по метрике «precision» значения 0,61, по метрике «ассигансу» значения 0,88, что показывает насколько точно она определяет верную эмоцию. Учитывая наилучшую существующую на данный момент модель, имеющую показатель «ассигансу» 0,90, можно утверждать, что данные показатели достаточно хорошие. На рисунке 12 представлены показатели нейронной сети, которые были получены при оценке результатов.

```

213/213 [=====] - 59s 274ms/step - loss: 0.8259 - accuracy: 0.8476 - precision: 0.763
ss: 1.0998 - val_accuracy: 0.8002 - val_precision: 0.6293 - val_recall: 0.4890 - val_auc: 0.8098 - lr: 0.0010
Epoch 53/100
213/213 [=====] - 59s 279ms/step - loss: 0.8132 - accuracy: 0.8544 - precision: 0.779
ss: 1.1115 - val_accuracy: 0.7999 - val_precision: 0.6248 - val_recall: 0.5000 - val_auc: 0.8101 - lr: 0.0010
Epoch 54/100
213/213 [=====] - 58s 275ms/step - loss: 0.7815 - accuracy: 0.8598 - precision: 0.790
ss: 1.1583 - val_accuracy: 0.7947 - val_precision: 0.6093 - val_recall: 0.4978 - val_auc: 0.8038 - lr: 0.0010
Epoch 55/100
213/213 [=====] - 58s 274ms/step - loss: 0.7932 - accuracy: 0.8550 - precision: 0.776
ss: 1.1117 - val_accuracy: 0.7947 - val_precision: 0.6225 - val_recall: 0.4538 - val_auc: 0.8042 - lr: 0.0010
Epoch 56/100
213/213 [=====] - 59s 276ms/step - loss: 0.7763 - accuracy: 0.8602 - precision: 0.783
ss: 1.1511 - val_accuracy: 0.7977 - val_precision: 0.6167 - val_recall: 0.5044 - val_auc: 0.8050 - lr: 0.0010
Epoch 57/100
213/213 [=====] - 58s 274ms/step - loss: 0.7724 - accuracy: 0.8623 - precision: 0.787
ss: 1.1651 - val_accuracy: 0.7934 - val_precision: 0.6087 - val_recall: 0.4858 - val_auc: 0.8012 - lr: 0.0010
Epoch 58/100
213/213 [=====] - 58s 274ms/step - loss: 0.7459 - accuracy: 0.8654 - precision: 0.795
ss: 1.1627 - val_accuracy: 0.7961 - val_precision: 0.6102 - val_recall: 0.5101 - val_auc: 0.8106 - lr: 0.0010
Epoch 59/100
213/213 [=====] - ETA: 0s - loss: 0.7523 - accuracy: 0.8643 - precision: 0.7912 - rec
Epoch 59: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
213/213 [=====] - 59s 276ms/step - loss: 0.7523 - accuracy: 0.8643 - precision: 0.791
ss: 1.1673 - val_accuracy: 0.7937 - val_precision: 0.6104 - val_recall: 0.4833 - val_auc: 0.8008 - lr: 0.0010
Epoch 59: early stopping

```

Рисунок 12 – Показатели нейронной сети на контрольной выборке

### Выводы по четвертой главе

Результаты функционального тестирования веб-приложения и тестирование качества нейронной сети совпадает с ожидаемыми. Все виды тестирования пройдены успешно.

## **ЗАКЛЮЧЕНИЕ**

В рамках выпускной квалификационной работы было разработано веб-приложение для определения эмоций и подборки музыки на основе нейронных сетей. При этом были решены следующие задачи:

- 1) проведен обзор аналогов и научной литературы;
- 2) изучены методы определения эмоций человека по фотографии с использованием нейросетевых технологий;
- 3) проведен сбор и предобработка данных для обучения;
- 4) разработана нейронная сеть и проведена оценка результатов ее работы;
- 5) разработано веб-приложение для прослушивания музыки;
- 6) проведено тестирование веб-приложения.

В будущем планируется, помимо улучшения количества музыкальных композиций, улучшить точность определения эмоций на фотографии при помощи изменения топологии и более глубокой подборки гиперпараметров.

Также можно расширить функционал веб-приложения, добавив функцию анализа текстов песен для определения их эмоциональной окраски и добавить авторизацию пользователей для сохранения истории прослушивания, так как данная информация очень важна для понимания интересов пользователей. Интеграция системы рекомендаций на основе предпочтений пользователя поможет значительно улучшить персонализацию сервиса и повысить удовлетворенность пользователей.

## ЛИТЕРАТУРА

1. Convolutional Neural Networks (CNN) Overview. [Электронный ресурс] URL: <https://encord.com/blog/convolutional-neural-networks-explained> (дата обращения: 13.02.2024 г.).
2. Deep Convolutional Neural Networks. [Электронный ресурс] URL: <https://www.run.ai/guides/deep-learning-for-computer-vision/deep-convolutional-neural-networks> (дата обращения: 13.02.2024 г.).
3. Filip R., Giorgos T., Ondrei C. CNN Image Retrieval Learns from BoW. // Unsupervised Fine-Tuning with Hard Example, 2016. – 17 с.
4. He K., Zhang X., Ren S., Sun J., Deep Residual Learning for Image Recognition // IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016. – 778 с.
5. Jupyter. [Электронный ресурс] URL: <https://jupyter.org> (дата обращения: 16.01.2024 г.).
6. Music Recommendation based on Facial Emotion Recognition. [Электронный ресурс] URL: <https://github.com/Dhruba59/Music-recommendation-based-on-facial-emotion-recognition> (дата обращения: 08.02.2024 г.).
7. Music Recommendation System using Spotify Dataset. [Электронный ресурс] URL: <https://www.kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset> (дата обращения: 25.02.2024 г.).
8. PyCharm. [Электронный ресурс] URL: <https://www.jetbrains.com/ru-ru/pycharm/> (дата обращения: 10.01.2024 г.).
9. Spotify Dataset. [Электронный ресурс] URL: <https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset> (дата обращения: 13.01.2024 г.).
10. TensorBoard. [Электронный ресурс] URL: <https://www.tensorflow.org/tensorboard> (дата обращения: 13.01.2024 г.).

11. W&B. (англ. Weight and Biases) [Электронный ресурс] URL: <https://wandb.ai> (дата обращения: 13.01.2024 г.).
12. Zepf S., Hernandez J., Schmitt A., Minker W., Picard R. Driver Emotion Recognition for Intelligent Vehicles // A Survey ACM Comput. Surv., 2020. – 30 с.
13. Алекс Б., Ева П. React и Redux. Функциональная веб-разработка // Издательский дом «Питер», 2017. – 336 с.
14. Анатолий П. Основы искусственного интеллекта в примерах на Python // БХВ-Петербург, 2021. – 444 с.
15. Библиотека JavaScript для создания внешних пользовательских интерфейсов React. [Электронный ресурс] URL: <https://react.dev> (дата обращения: 10.03.2024 г.).
16. Библиотека Python для машинного обучения Keras. [Электронный ресурс] URL: <https://keras.io> (дата обращения: 10.01.2024 г.).
17. Библиотека Python для машинного обучения TensorFlow. [Электронный ресурс] URL: <https://www.tensorflow.org> (дата обращения: 10.01.2024 г.).
18. Библиотека Python для проверки данных Pydantic. [Электронный ресурс] URL: <https://docs.pydantic.dev/latest> (дата обращения: 25.02.2024 г.).
19. Библиотека Python для работы с S3 хранилищем Boto3. [Электронный ресурс] URL: <https://pypi.org/project/boto3/> (дата обращения: 24.02.2024 г.).
20. Библиотека Python для работы с фотографиями OpenCV. [Электронный ресурс] URL: <https://opencv.org> (дата обращения: 12.01.2024 г.).
21. Библиотека React для создания анимированных компонентов Framer Motion. [Электронный ресурс] URL: <https://www.framer.com/motion/> (дата обращения: 10.03.2024 г.).
22. Библиотека React компонентов Material UI. [Электронный ресурс] URL: <https://mui.com/material-ui/> (дата обращения: 10.03.2024 г.).

23. Библиотека линейной алгебры, разработанная на Python NumPy. [Электронный ресурс] URL: <https://numpy.org> (дата обращения: 10.02.2024 г.).
24. Высокоуровневый язык программирования Python. [Электронный ресурс] URL: <https://www.python.org> (дата обращения: 08.01.2024 г.).
25. Джейк В., Python для сложных задач. Наука о данных и машинное обучение // O'REILLY, 2017. – 576 с.
26. Искусственный интеллект для эмоциональной классификации музыки. [Электронный ресурс] URL: <https://cyanite.ai> (дата обращения: 20.02.2024 г.).
27. Миша Г., Йен О. Высокопроизводительные Python-приложения // Бомбора, 2022. – 528 с.
28. Музыкальные предпочтения россиян. [Электронный ресурс] URL: <https://iom.anketolog.ru/2020/07/07/muzykal-nye-predpochteniya-rossiyan> (дата обращения: 15.12.2023 г.).
29. Нейронная сеть для определения лица на фотографии «Face recognition». [Электронный ресурс] URL: [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition) (дата обращения: 18.02.2024 г.).
30. Обзор форматов изображений. [Электронный ресурс] URL: <https://www.fotostoki.ru/articles/misc/image-file-formats.html> (дата обращения: 05.03.2024 г.).
31. Платформа мониторинга и анализа данных с интерактивной визуализацией Grafana. [Электронный ресурс] URL: <https://grafana.com> (дата обращения: 25.04.2024 г.).
32. Программное обеспечение для автоматизации развертывания приложения Docker. [Электронный ресурс] URL: <https://www.docker.com> (дата обращения: 25.04.2024 г.).

33. Система для развертывания Tensorflow моделей Tensorflow Serving. [Электронный ресурс] URL: <https://www.tensorflow.org/tfx/guide/serving> (дата обращения: 15.02.2024 г.).
34. Система мониторинга серверов Prometheus. [Электронный ресурс] URL: <https://prometheus.io> (дата обращения: 15.02.2024 г.).
35. Типизированный язык программирования, построенный на JavaScript – TypeScript. [Электронный ресурс] URL: <https://www.typescriptlang.org> (дата обращения: 15.03.2024 г.).
36. Фреймворк Python для создания быстрых HTTP API-серверов FastAPI. [Электронный ресурс] URL: <https://fastapi.tiangolo.com/ru/> (дата обращения: 12.02.2024 г.).