

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ

КАЗАХСТАН СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра «Программная инженерия»

Ихсанова-Солод Жаннель Владимировна

Музыкальный плейлист основанный на эмоциях

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и
программное обеспечение

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ


Институт информационных и телекоммуникационных технологий

Кафедра «Программная инженерия»

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой ПИ

Доктор Ph. D.

 М.Турдалыулы

«06» июня 2021 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

На тему: «Музыкальный плейлист основанный на эмоциях»

по специальности 5В070400 – Вычислительная техника и программное
обеспечение

Выполнил

Ихсанова-Солод Ж.В.

Научный руководитель
Магистр технических наук,

Сениор-лектор

 А.Д.Куникеев

«06» июня 2021 г.

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра «Программная инженерия»


5B070400 – Вычислительная техника и программное обеспечение

УТВЕРЖДАЮ

Заведующий кафедрой ПИ

Доктор Ph/D.

М.Турдалыулы

 «06» июня 2021 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Ихсановой-Солод Жаннель Владимировне

Тема: Музыкальный плейлист основанный на эмоциях

Исходные данные к дипломному проекту: Описание необходимых функций проекта.

Перечень подлежащих разработке в дипломном проекте вопросов:

а) реализация связи мобильного приложения с бэкенд-сервером;

б) проектирование и разработка мобильного приложения;

в) разработка K-means алгоритма;

г) интеграция HAAR cascade face detection;

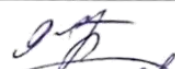
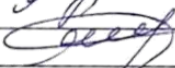
Перечень графического материала (с точным указанием обязательных чертежей): представлены в 28 слайдах презентации.


ГРАФИК
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области, разработка технического задания	21. 01. 2021	Выполнено
2. Выбор базового компонента	21. 02. 2021	Выполнено
3. Разработка дизайна интерфейса	10. 03. 2021	Выполнено
4. Разработка функционала мобильного приложения	15. 05. 2021	Выполнено
5. Тестирование мобильного приложения	17. 05. 2021	Выполнено
6. Написание пояснительной записки к дипломному проекту	28. 05. 2021	Выполнено

Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	Рамазан А.Б.	31. 05. 2021	
Нормоконтролер	Марғұлан Қ.	31. 05. 2021	

Научный руководитель _____ А.Д.Куникеев
Задание принял к исполнению обучающийся  Ж.В.Ихсанова-Солод
Дата _____ «06» июня 2021 г.

АННОТАЦИЯ

Emotion-based music player – инновационное мобильное приложение для ОС Android. Данное уникальное приложение умеет распознавать эмоции пользователя, при помощи которых создается соответствующий список музыкальных композиций. Приложение направлено на ускорение и автоматизирование процесса подбора музыкальных плейлистов при помощи рекомендательной системы на основе распознавания эмоций, отражающихся на лице пользователя. Пользователь после аутентификации в мобильном приложении, наводит камеру на уровень лица, после чего модель распознавания эмоций считывает текущее настроение пользователя и выдает готовый плейлист с правильно подобранными музыкальными композициями.

В этой работе мы рассматриваем решение автоматизации подбора музыкального плейлиста пользователя. Для решения был создан проект, состоящий из трех частей: мобильное приложение, построенное при помощи Android Studio и языка программирования Kotlin, модель распознавания эмоций, реализованная на языке программирования Python, и бэкенд сервер при помощи фреймворка Flask на языке программирования Python.

АНДАТПА

Emotion-based music player - Android ОЖ арналған инновациялық мобильді қосымша. Бұл бірегей қосымша пайдаланушының эмоцияларын тани отыра, олардың көмегімен музыкалық композициялардың тиісті тізімін жасақтайды. Қосымша пайдаланушының бетінде көрінетін эмоцияларды тануға негізделген ұсыныс жүйесін қолдана отырып, музыкалық ойнату тізімдерін құрастыру процесін жеделдетуге және автоматтандыруға бағытталған. Пайдаланушы мобильді қосымшада аутентификациядан кейін камераны бет деңгейіне бағыттайды, содан кейін эмоцияны тану моделі пайдаланушының қазіргі көңіл-күйін оқиды және сәйкес таңдалған музыкалық композициялармен дайын ойнату тізімін береді.

Бұл жұмыста біз пайдаланушының музыкалық ойнату тізімін таңдауды автоматтандыру шешімін қарастырамыз. Берілген мәселені шешу үшін үш бөліктен тұратын жоба құрылды: Android Studio және Kotlin бағдарламалау тілінің көмегімен жасалған мобильді қосымша, Python бағдарламалау тілінде іске асырылған эмоцияны тану моделі және Python бағдарламалау тілінде Flask жақтауының көмегімен бэкенд сервер.

ANNOTATION

Emotion-based music player is an innovative mobile application for the Android OS. This unique application is able to recognize the emotions of the user, through which a corresponding list of musical compositions is created. The application is aimed at speeding up and automating the process of selecting music playlists by recommendation system based on the recognition of emotions reflected on the user's face. After authentication in the mobile app, the user points the camera at the face level, after which the emotion recognition model detects the current mood of the user and outputs a ready-made playlist with correctly selected musical compositions.

In this article, we consider a solution for automating the selection of a user's music playlist. For the solution, we created a project consisting of three parts: a mobile application created by using Android Studio and the Kotlin programming language, an emotion recognition model implemented in the Python programming language and a backend server using the Flask framework in the Python programming language.

СОДЕРЖАНИЕ

	Введение	10
1	Исследовательский раздел	11
1.1	Цель разработки	11
1.2	Термины и сокращения	11
1.3	Предметная область	12
1.3.1	AI	12
1.3.2	Основные преимущества CV	13
1.3.3	Основные функции CV	13
1.3.4	Android	13
1.3.5	Основные преимущества Android	14
1.3.6	Основные функции Android	14
2	Технологический раздел	16
2.1	Фреймворк Flask	16
2.2	Среда разработки	16
3	Проектная часть	17
3.1	Архитектура системы взаимодействия	17
3.2	Описание диаграммы деятельности	18
3.3	Разработка пользовательского интерфейса	19
4	Экспериментальный раздел	28
4.1	Реализация связи мобильного приложения с бэкенд-сервером	28
4.2	Разработка K-means алгоритма	29
4.3	Интеграция HAAR cascade face detection	30
	Заключение	31
	Список использованной литературы	32
	Приложение А. Техническое задание	33

ВВЕДЕНИЕ

Недавние исследования подтверждают, что люди отзываются и реагируют на музыку, и что музыка оказывает сильное влияние на мозговую деятельность человека. Среднестатистический казахстанец слушает до четырех часов музыки каждый день. Люди склонны слушать музыку в зависимости от настроения и интересов. Этот проект направлен на создание приложения, предлагающего песни для пользователей в зависимости от настроения, распознавая выражения лица.

Выражение лица - это форма невербального общения. Компьютерное зрение - это междисциплинарная область, которая помогает передать машинам понимание цифровых изображений или видео на высоком уровне. В этой системе компьютерного зрения компоненты используются для определения эмоций пользователя через мимику. Как только эмоция распознается, система предлагает плейлист для этой эмоции, позволяющий пользователю сэкономить время при выборе и воспроизведении песни вручную. Музыкальный проигрыватель на основе эмоций также отслеживает такие данные пользователя, как количество воспроизведений каждой песни, сортировка песен по категориям и уровню интереса, а также реорганизует каждый раз плейлист. Система также уведомляет пользователя о песнях, которые никогда не воспроизводятся, поэтому их можно удалить или изменить.

1 Исследовательский раздел

1.1 Цель разработки

Музыка - важное средство развлечения. С развитием технологий, большое внимание уделяется оптимизации ручной работы. В настоящее время есть много традиционных музыкальных плееров, которые требуют ручного выбора песен и организации. Пользователь, должен создать и обновлять плейлист для каждого настроения, что занимает некоторое время. Некоторые музыкальные плееры имеют расширенные функции, такие как распознавание текста песни и рекомендовать похожие песни в зависимости от исполнителя или жанра. Хотя некоторые из функций приятны пользователю, есть достаточное количество нюансов для улучшения в области автоматизации музыкальных плееров. Автоматический выбор песен и их организация на основе настроения пользователя дает пользователю лучший опыт использования приложения. Этого можно добиться, если система реагирует на эмоции пользователя, экономя время, которое было бы потрачено на ввод информации вручную. Эмоции можно выражать с помощью жестов, речи, мимики и т. д.

Чтобы система могла понять настроение пользователя, мы используем выражение лица. С помощью камеры мобильного устройства мы можем запечатлеть выражение лица пользователя. Существует множество систем распознавания эмоций, которые принимают захваченное изображение в качестве входных данных и определяют эмоцию. В этом приложении мы используем обученную модель распознавания лица на языке программирования Python. Система включает алгоритм CNN, который считывает изображения и делит эмоции на семь классов. Алгоритм рекомендательной системы кластеризует песни на классы эмоций. После чего, согласно полученной эмоции от алгоритма CNN, рекомендационная система создает плейлист.

1.2 Термины и сокращения

Термины и сокращения, использованные при разработке, а также связанные с реализацией и используемыми технологиями описаны в таблице 1.

Таблица 1 - Термины, сокращения, и их определения

Сокращение или термин	Определение
CNN	(сокр. от англ. convolutional neural network, CNN) — это сверточная нейронная сеть
IDE	(сокр. от англ. Integrated development environment) Интегрированная среда разработки — это программное обеспечение, позволяющее создавать приложения и объединяющее главные инструменты разработчика и графический пользовательский интерфейс (GUI)..
UI	(сокр. от англ. User interface) Пользовательский интерфейс.
HTTP	(сокр. от англ. Hypertext Transfer Protocol) Протокол передачи гипертекста — это прикладной протокол для распределенных, совместных, гипермедиа информационных систем.
AI	(сокр. от англ. Artificial Intelligence) – это искусственный интеллект
CV	(сокр. от англ. Computer Vision) – это компьютерное зрение
JVM	(сокр. от англ. Java Virtual Machine) - виртуальная машина Java

1.3 Предметная область

1.3.1 AI

Перед началом разработки необходимо изучить предметную область и дать определение понятию AI. В процессе исследования выяснилось, что AI (сокр. от англ. Artificial Intelligence) – это искусственный интеллект, другими словами, это возможность для информационных систем обрабатывать информацию с творческой стороны. Искусственный интеллект, в теории, должен иметь возможность самостоятельно принимать решения, думать, основываясь не только на предоставленных данных, но и с помощью дополнительных неявных связей. Благодаря искусственному интеллекту, человечество сделало огромный шаг в развитии технологий: компьютер теперь способен анализировать данные, оптимизировать процессы во многих отраслях промышленности и не только. Искусственный интеллект – большое направление, у которого много областей исследования. Одной из таких областей является CV (сокр. от англ. Computer Vision) – это компьютерное зрение, способность машин обрабатывать и анализировать изображение, другими словами, —видеть картинку

1.3.2 Основные преимущества CV

CV (или компьютерное зрение) помогает в трех основных направлениях:

- наиболее быстрый и простой способ автоматизации процессов реализации монотонных задач;
- улучшенные технологические решения: благодаря системам компьютерного зрения, ошибки минимизированы до 0%, это позволяет создавать продукты наилучшего качества;
- снижение затрат: компаниям не нужно будет платить за дефекты в производстве продуктов - системы компьютерного зрения предотвращают любые дефекты.

1.3.3 Основные функции CV

Основная цель CV - повысить быстроту и эффективность пользования приложением за счет сокращения ручных операций.

Проект предусматривает собой создание плейлиста, основанного на эмоциях, поэтому для быстроты и точности было выбрано компьютерное зрение, которое с помощью модели CNN и HAAR cascade face recognition

быстро и точно определяет эмоцию пользователя, по которой, впоследствии, создается плейлист музыкальных композиций.

1.3.4 Android

Android - операционная система, представленная компанией Android Inc. в 2008 году, позднее была куплена компанией Google, разработанная для смартфонов, планшетов и прочих смарт-устройств. Система построена на ядре Linux и реализована с помощью JVM. В данный момент, для создания приложений на Android используют два языка программирования: Java и Kotlin. Java - это объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems в 1995 году. Kotlin - более молодая версия Java (данный язык программирования также типизированный и объектно-ориентированный), разработанная компанией JetBrains в 2011 году.

1.3.5 Основные преимущества Android

На сегодняшний день технологии развиваются не по дням, а по часам. Любые элементы, помогающие облегчить человеку жизнь, оптимизируют, делая из них смарт-устройства. Например, смарт-часы, которые подключаются к смартфону и имеют ряд функций, облегчающих пользование устройством, или смарт-телевизор, который имеет ограниченный функционал, помогающий найти любой фильм в Интернете, записать серию тв-шоу и прочее. Такие умные устройства в 75% работают на Android, что делает данную операционную систему наиболее востребованной на рынке разработки приложений. Рассмотрим некоторые преимущества выбора Android:

- поддержка Google. Являясь одной из самой прогрессивной компаний в мире, Google активно внедряет новые функции и продукты в устройства на основе Android, что способствует удобному применению новых технологий в приложениях;
- большой спектр устройств. Разрабатывая приложение на Android - означает большой охват пользователей различных устройств;
- быстрая разработка приложений. Android - платформа с открытым исходным кодом, которая включает в себя множество фреймворков, библиотек и плагинов, которые, в свою очередь, значительно упрощают процесс разработки приложений.

1.3.6 Основные функции Android

Основной целью Android - это разработка мобильного приложения, которое имеет функции:

- снимок фотографии или выбор готового изображения из галереи;
- аутентификация с помощью Spotify;
- просмотр профиля пользователя, главной страницы, проигрывание песен и поиск по музыкальным композициям, вывод плейлистов;
- передача и получение данных.

2 Технологический раздел

2.1 Фреймворк Flask

Flask - это кроссплатформенный фреймворк для создания веб-приложений, разработанный в 2010 году австрийским программистом Армином Ронахером. Фреймворк также позволяет интегрироваться с внешними библиотеками и легко расширяется. Flask относится к микрофреймворкам, т.е. приложения, созданные при помощи данного фреймворка, сознательно выполняют базовые функции.

На Flask приложения создаются на языке программирования Python, использующие инструменты Werkzeug и шаблонизатор Jinja2.

2.2 Среда разработки

Интегрированная среда разработки (сокр. от англ. Integrated Development Environment) — это программное обеспечение, позволяющее создавать приложения и объединяющее главные инструменты разработчика и графический пользовательский интерфейс (GUI). Такие IDE значительно упрощают процесс разработки.

Рассмотрим лучшие IDE для Flask, которые необходимо использовать и которые могут помочь в разработке веб-приложений.

PyCharm - это редактор кода, созданный на основе аналогичного продукта IntelliJ IDEA, представленный компанией JetBrains в 2013 году, отлично подходит для разработки веб-приложений на фреймворках Flask и Django. Он включает в себя автозаполнение кода, переработка кода, имеется отладчик кода при помощи PyDev и поддержка Git, SVN, Mercurial и другие. У PyCharm есть две версии: Community Edition (бесплатная, которая имеет ограниченный функционал) и Professional Edition (платная, которая представлена в полной функциональности).

Sublime Text — это бесплатный текстовый редактор для кода и разметки. Отлично подходит для ознакомления со структурой фреймворка, знакомства с языком программирования Python, удобно создавать в нем HTML-страницы, но нет инструментов для отладки и переработки кода, ошибки выявляются лишь на моменте запуска программы. Имеет довольно-таки приятный интерфейс, но на этом преимущества заканчиваются.

Рассмотрев все вышеперечисленные IDE, для разработки на фреймворке Flask выбор был сделан в пользу PyCharm. Для разработки на Android был выбран Android Studio.

3 Проектная часть

3.1 Архитектура системы взаимодействия

Разрабатываемый проект предназначен для открытого пользования и основан на клиент-серверной архитектуре, состоит из трех частей: мобильное приложение, серверная часть и модель машинного обучения. Первая часть проекта - это мобильное приложение на Android, написанное на языке программирования Kotlin. Таким образом, взаимодействие между частями выглядит следующим образом: мобильное приложение, посредством request-запроса, передает изображение на серверную часть, которая реализована на языке программирования Python при помощи фреймворка Flask. Полученное изображение сохраняется в главной директории, и модель машинного обучения CNN, ожидающая фотографию по этому пути, обрабатывает полученное изображение и на выходе выдает ответ в виде строки с выявленным настроением. Далее эту строку принимает классификатор K-means algorithm, где происходит кластеризация музыкальных композиций по определенному настроению, и формируется плейлист. Полученный плейлист сохраняется в виде JSON и передается обратно на сторону мобильного приложения. Архитектура проекта отражена на рисунке 3.1.



Рисунок 3.1 – Архитектура системы взаимодействия

3.2 Описание диаграммы деятельности

Диаграмма деятельности (диаграмма активностей) – одна из важнейших UML диаграмм, показывающая динамические аспекты системы, т.е. отражает последовательность действий, можно считать, что диаграмма активностей является расширенной версией блок-схем, которая моделирует переход от одного действия к другому. Действия, которые указаны в диаграмме, могут

быть как последовательными, так и параллельными. Важно указать начало действий (т.е начальное состояние) и конец (т.е конечное состояние).

Для создания диаграммы деятельности для проекта было выбрано бесплатное приложение draw.io, созданное компанией Google.

Работа приложения начинается с того, что пользователь аутентифицируется с помощью Spotify API или Google API. После удачной аутентификации, пользователю открываются страницы мобильного приложения, такие как: Главная страница, Профиль пользователя, страница камеры и страница Поиска.

Меню располагается снизу и является навигацией для существующих страниц.

Главная страница представляет собой список песен, плейлистов. Профиль пользователя представляет собой данные о пользователе (имя, фамилия, статистика работы приложения и подписки на артистов).

Страница камеры реализована, как обычный функционал, существующий в современных смартфонах, отличие лишь в том, что полученное изображение (сфотографированное или выбранное из галереи) отправляется на Flask-сервер.

Страница поиска представляет собой поле, в которое вписывается название песни, имя артиста или название альбома, и списком выводится результат.

Далее пользователь нажимает на кнопку с иконкой камеры, которая расположена в меню, и попадает на страницу камеры, где от него требуется два действия: сделать снимок своего лица с текущей эмоцией или же выбрать из галереи уже готовое изображение с соответствующим настроением. Готовое изображение пользователь отправляет на сервер и ждет ответа.

В это время Flask-сервер ждет запрос от мобильного приложения. После успешного получения изображения, файл сохраняется в главной директории, которую, в свою очередь, просматривает модель машинного обучения CNN.

CNN модель получает изображение и начинает обработку, с помощью алгоритма HAAR cascade модель определяет лицо и применяет обученную 4-х слойную нейронную сеть, которая, в результате, дает на выход в виде строки настроение, определенное на изображении. Этот результат передается в модель K-mean algorithm.

K-mean algorithm - это кластеризатор, который принимает аутентифицированного пользователя и настроение, полученное от CNN модели и генерирует плейлисты, основанные на 5-ти признаках: танцевальность, громкость, наличие речи, акустичность и живость музыкальных композиций. После обработки песен и формирования плейлиста, генерируется JSON-файл, состоящий из списка музыкальных композиций, названия плейлиста и прочее. Такой файл передается обратно к ожидающему мобильному приложению и выводится на главной странице.

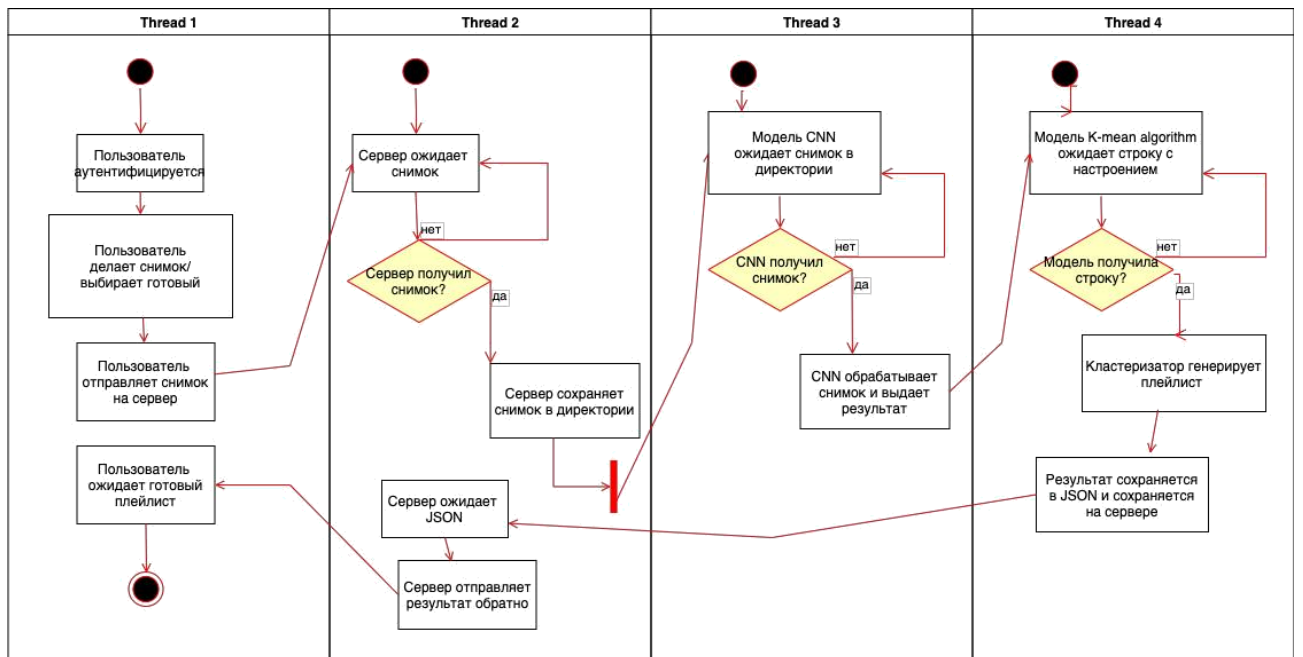


Рисунок 3.2.1 – Диаграмма деятельности

3.3 Разработка пользовательского интерфейса

Разработка пользовательского интерфейса — это разработка удобного и понятного дизайна веб-приложений, мобильных приложений и прочего. Ценность правильно разработанного пользовательского интерфейса велика, т.к именно через фронтальную сторону пользователь и использует продукт, поэтому важно сделать интерфейс дружелюбным, простым и эффективным.

Для разработки пользовательского интерфейса было выбрано приложение Figma. Figma - сервис для разработки дизайна пользовательского интерфейса, включающий расширенные инструменты и прочие дополнения.

В рамках данного проекта пользователь сталкивается лишь с мобильным приложением, которое реализовано на Android, и имеет четыре активных страницы. Рассмотрим страницы мобильного приложения подробнее:

При открытии приложения пользователя встречает приветственная начальная страница, на которой имеет кнопка «Начать». При нажатии на эту кнопку пользователь переходит на страницу аутентификации.

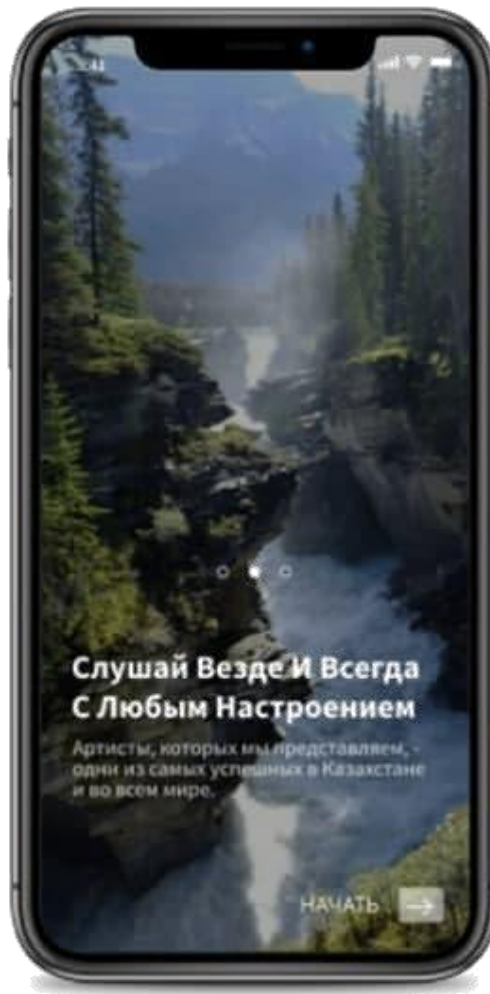


Рисунок 3.3.1 – Начальная страница приложения

На странице аутентификации пользователь видит три кнопки: —Get started, —Login, —Sign In.

Кнопка —Get started переносит пользователя на страницу аутентификации. где появляется два вида входа: при помощи Spotify аккаунта и посредством Google аккаунта.



Рисунок 3.3.2 – Страница аутентификации приложения

На главной странице пользователя встречает список музыкальных композиций, плейлисты. Также пользователь видит внизу меню, у которого имеется пять кнопок: —домой, —поиск, —камера, —голос, —профиль.

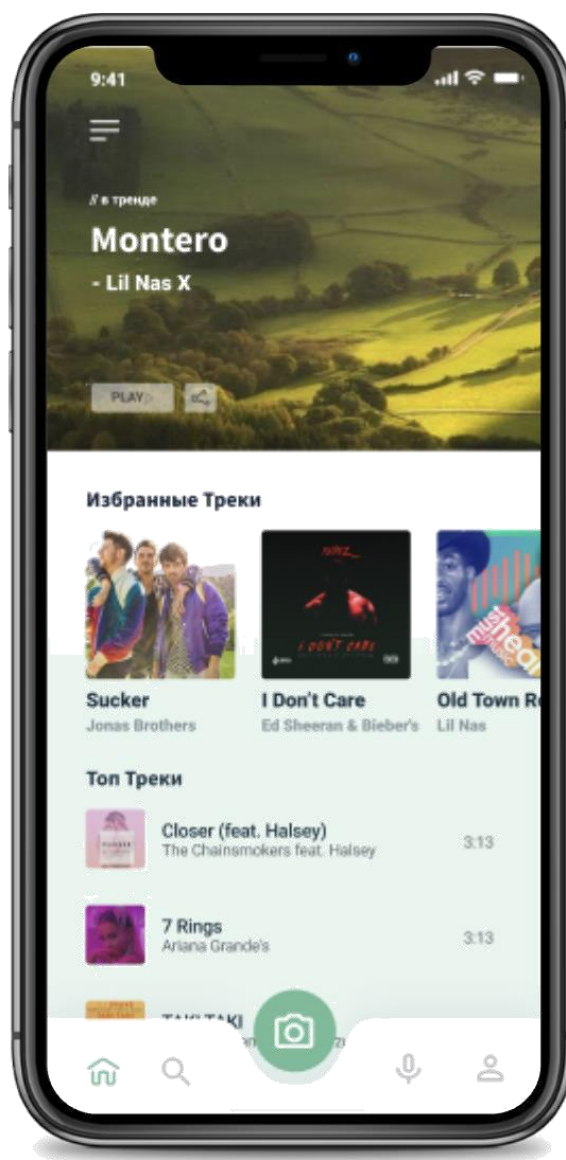


Рисунок 3.3.3 – Главная страница приложения

Кнопка —профиль переносит пользователя на его страницу в приложении, где указаны его данные, выведена статистику пользования приложением за неделю, и имеются функции редактирования профиля и просмотр подписок на артистов.

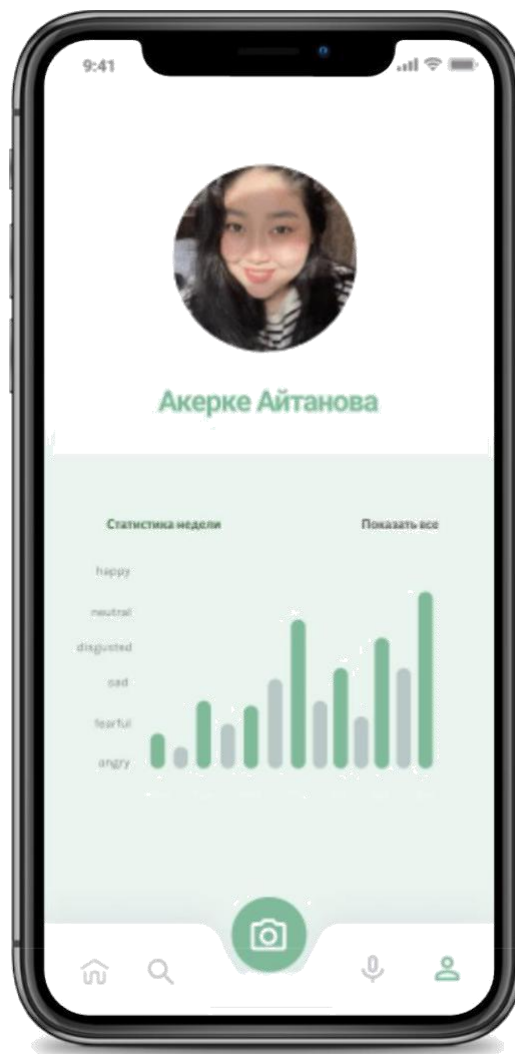


Рисунок 3.3.4 – Страница профиля в приложении

Если нажать на кнопку —камера, то пользователь перейдет на стандартную камеру смартфона и встанет перед выбором: сделать снимок с текущей эмоцией или же выбрать готовое изображение из галереи.



Рисунок 3.3.5 – Страница камеры в приложении



Рисунок 3.3.6 – Выбор изображения в приложении

После того, как пользователь сделал выбор, изображение нужно отправить серверной части проекта, поэтому тут реализована только кнопка —выбрать, которая и отправляет снимок. После нажатия кнопки, пользователь ожидает в среднем одну минуту и получает обратно снимок с выделенным лицом и определенным настроением. Пользователь должен нажать кнопку —готово, чтобы получить сгенерированный плейлист.

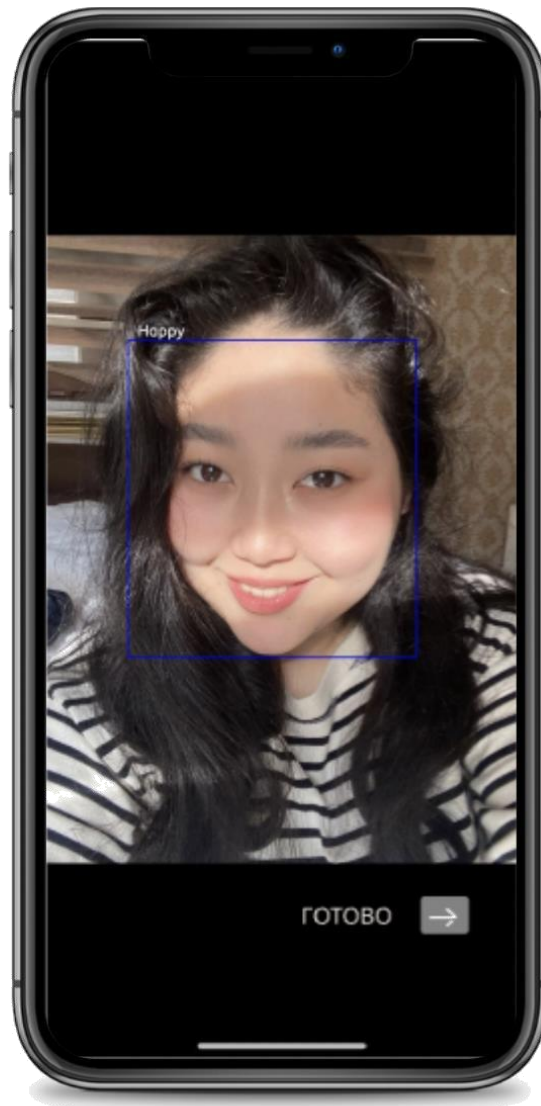


Рисунок 3.3.7 – Получение снимка обратно

Далее пользователь попадает на главную страницу приложения и видит новый плейлист по его текущему настроению и прослушивает подобранные музыкальные композиции.

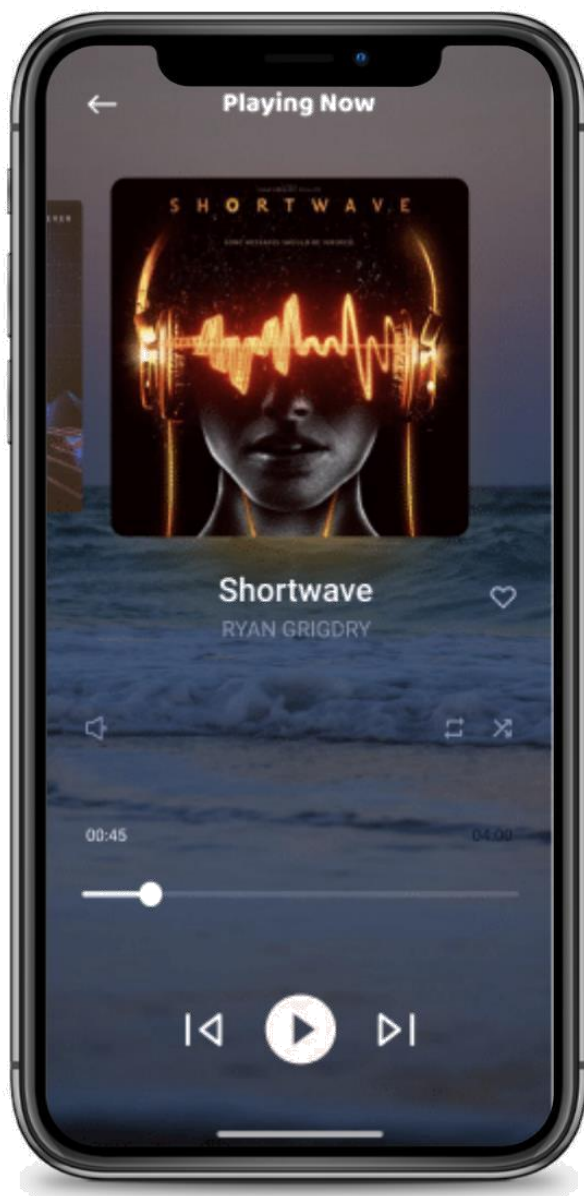


Рисунок 3.3.7 – Готовый плейлист

4 Экспериментальный раздел

4.1 Реализация связи мобильного приложения с бэкенд-сервером

Как известно, мобильные устройства ограничены в ресурсах и все еще проигрывают компьютерам в мощностях. С помощью смартфонов можно проделать множество вещей, однако, в проекте присутствует машинное обучение, а именно глубокое обучение с большим количеством данных для нейронной сети, поэтому было принято решение построить архитектуру проекта с помощью связи мобильного приложения и сервера. Для реализации такой архитектуры было разработано клиентское приложение на Android и сервер на фреймворке Flask.

Начнем настройку на стороне мобильного приложения. Для этого мы используем OkHttp. OkHttp - это HTTP/HTTP2 клиент, работающий на верхнем уровне с сокетами Java и не имеющий дополнительных зависимостей. Главное преимущество такого клиента - асинхронные вызовы с обратным вызовом, что, в нашем случае, очень важно.

После того, как мы успешно импортировали OkHttp в проект мобильного приложения, надо указать url локальной машины и прописать запрос, который будет передавать токен аутентификации и снимок.

Далее мы создаем обратный вызов, который будет ожидать JSON файл от сервера.

Теперь можно перейти и к настройке Flask сервера. Для этого мы должны указать в app хост и IP-адрес, по которому будет стучаться мобильное приложение, и реализовать метод `handle_request()`, который и будет принимать сообщение от клиента и сохранять снимок в нужной директории.

На каждом этапе нужно также прописать запросы, которые будут информировать об успешности или ошибке отправленных и полученных сообщений между клиентом и сервером.

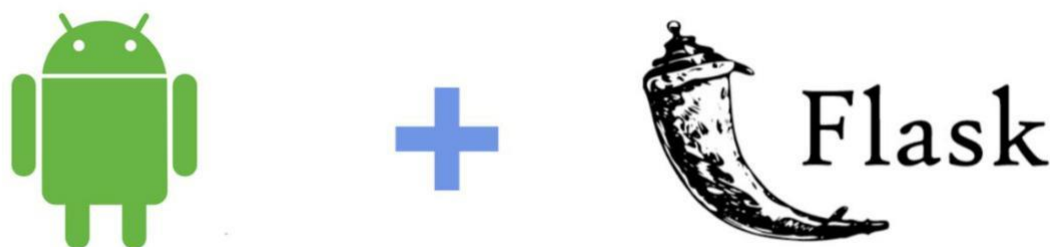


Рисунок 4.1.1 – Связь клиента и сервера

4.2 Разработка K-means алгоритма

Перед началом работы над проектом стоял вопрос того, как же все-таки реализовать рекомендательную систему, основываясь лишь на одном признаке, в нашем случае, на настроении.

Сначала надо было проверить какие данные можно будет использовать для поиска нужных музыкальных композиций, поэтому после изучения данных, которые предоставляет Spotify API, главными признаками были выбраны: танцевальность, громкость, наличие речи, акустичность и живость музыкальных композиций. Значения таких признаков варьируются от 0 до 1, где 0 - полное отсутствие, а 1 - полное преобладание признака в песне. После того, как определились с тем, как делать выборку, надо проверить оптимальность различных кластеров. Было выбрано три наиболее оптимальных и эффективных алгоритмов кластеризации: K-means алгоритм, алгоритм сдвига среднего значения и Гауссова смесь распределения. В процессе проверок было выявлено, что лучше всего справляется K-means с четырьмя кластерами.

Теперь можно перейти непосредственно к построению модели кластеризации при помощи K-means алгоритма. Для это передаем строку с настроением и применяем уже созданные кластеры по 5 признакам описанных выше. Далее отправляем запрос на выборку 50 музыкальных композиций, после чего происходит формирование плейлиста. Если плейлист готов, то генерируем JSON файл и сохраняем в нужной директории на сервере.

4.3 Интеграция HAAR cascade face detection

Для распознавания эмоций первым делом надо найти лицо на изображении, поэтому в проекте использован алгоритм HAAR cascade, который предоставляет библиотека компьютерного зрения opencv.

Изображение, полученное от мобильного приложения, сохраняется на сервере и передается в модель CNN, и с помощью функции detectMultiScale со значением scaleFactor равным 1,3. Далее мы устанавливаем границы лица и рисуем прямоугольник, обозначая границы, и передаем строку с настроением в модель кластеризации.

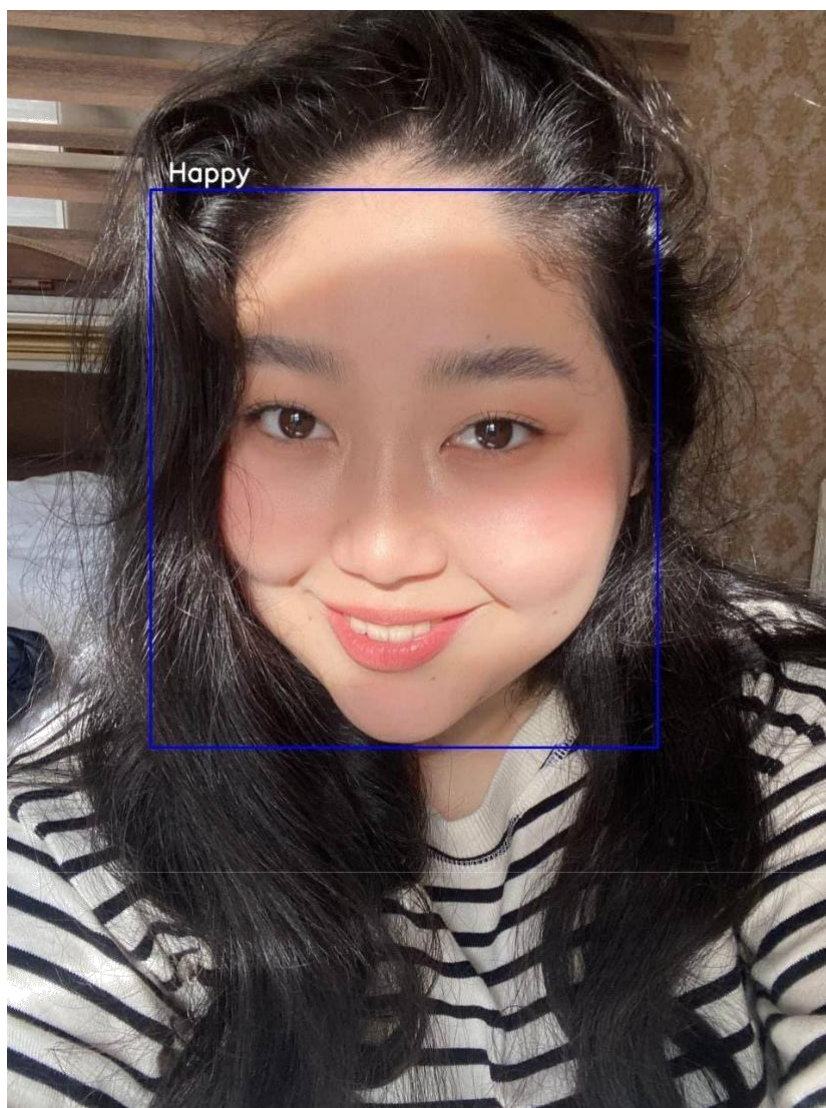


Рисунок 4.3.1 – Интеграция HAAR cascade в проект

ЗАКЛЮЧЕНИЕ

В рамках данного дипломного проекта поставленная цель была выполнена успешно. В результате реализовано клиент-серверное мобильное приложение на Android, которое, при помощи машинного обучения, определяет эмоцию пользователя и создает плейлист под настроение. Данный проект весомо сокращает время подбора музыкальных композиций и создает идеальный плейлист для текущего настроения.

В процессе подготовки к реализации проекта была подробно изучена предметная область распознавания лиц, эмоций посредством нейронных сетей. Было выявлено, что наличие модели распознавания эмоций значительно улучшает работу продуктов, особенно, если это напрямую связано с пользователями.

В процессе разработки и проектирования архитектуры проекта была проведена детальная работа над подбором оптимальных технологий для общения между частями проекта и выявлены наиболее удачные и быстрые.

Мобильное приложение —Музыкальный плейлист основанный на эмоциях|| представляет собой законченный продукт, который имеет удобный пользовательский интерфейс и оптимальную и эффективную производительность. Приложение поможет легко и быстро сгенерировать плейлист лишь по фотографии, где отражена текущая эмоция.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Flask's documentation // Электронная версия на сайте <https://flask.palletsprojects.com/en/2.0.x/>
2. Cascade classifier // Электронная версия на сайте https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
3. How do CNN works? // Электронная версия на сайте <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>
4. Kotlin and Android // Электронная версия на сайте <https://developer.android.com/kotlin/first>
5. Understanding K-means Clustering // Электронная версия на сайте <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>
6. Гундфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение // Электронная версия на сайте <https://liters.ru>
7. Адриан Р. Deep learning for Computer Vision with Python // Электронная версия на сайте <https://liters.ru>
8. Джурда П., The Book of Why // Электронная версия на сайте <https://liters.ru>
9. Клепман М., Высоконагруженные приложения // Электронная версия на сайте <https://liters.ru>
10. Гриффитс Д. Head First. Программирование для Android // Электронная версия на сайте <https://all-ebooks.com>

Приложение А (обязательное)

Техническое задание

А1.1 Техническое задание на разработку мобильного приложения с распознаванием эмоций

Настоящее техническое задание распространяется на разработку мобильного приложения с распознаванием эмоций. Предполагается что использовать данную систему будут пользователи смартфонов с Android. Мобильное приложение позволит значительно сократить процесс подбора музыкальных композиций.

А1.1.1 Основания для разработки

Редактор разрабатывается на основании устного распоряжения Заместителя Председателя Правления АО «Институт цифровой техники и технологий» по разработке программных продуктов.

А1.1.2 Назначение

Разрабатываемое мобильное приложение предназначено для генерации плейлистов, основанных на эмоциях, которые распознает модель машинного обучения.

А1.1.3 Требования к функциональным характеристикам

Мобильное приложение должно обеспечить возможность выполнения следующих функций:

- распознавание эмоций;
- аутентификация пользователя;
- вывод данных при помощи Spotify API;
- поиск по музыкальным композициям;

Продолжение приложения А

А1.1.4 Требования к надежности

Обеспечить конфиденциальность данных пользователя.

Приложение В (обязательное)

Текст программы

1. Реализация функции handle_request():

```
@app.route('/', methods = ['GET',
'POST']) def handle_request():
    files_ids = list(flask.request.files)
    print("\nNumber of Received Images : ", len(files_ids))
    image_num = 1
    global imagefile
    for file_id in files_ids:
        print("\nSaving Image ", str(image_num), "/static", len(files_ids))

        imagefile = flask.request.files[file_id]

        filename = werkzeug.utils.secure_filename(imagefile.filename)
        print("Image Filename : " + imagefile.filename)
        #timestr = time.strftime("%Y%m%d-%H%M%S")
        imagefile.save(filename)

        image_num = image_num + 1
    print("\n")
```

2. Использование модели машинного обучения

```
CNN: model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
input_shape=(48,48,1)))
model.add(Conv2D(64, kernel_size=(3, 3),
activation='relu')) model.add(MaxPooling2D(pool_size=(2,
2))) model.add(Dropout(0.25))

model.add(Conv2D(128, kernel_size=(3, 3),
activation='relu')) model.add(MaxPooling2D(pool_size=(2,
2))) model.add(Conv2D(128, kernel_size=(3, 3),
activation='relu')) model.add(MaxPooling2D(pool_size=(2,
2))) model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(1024, activation='relu'))
```

```

model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))

model.load_weights('model.h5')

imagePath =
"C:/Users/zhann/OneDrive/Pictures/edd4517f9f287b61e50b6f17b52860c9.j
pg"

# prevents openCL usage and unnecessary logging
messages cv2ocl.setUseOpenCL(False)

# dictionary which assigns each label an emotion (alphabetical order)
emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4:
"Neutral", 5: "Sad", 6: "Surprised"}

# start the webcam feed
# cap = cv2.VideoCapture(0)
while True:

    facecasc = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    image = cv2.imread(imagePath)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = facecasc.detectMultiScale(gray, scaleFactor=1.3,
minNeighbors=5)

    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y - 50), (x + w, y + h + 10), (255, 0, 0), 2)
        roi_gray = gray[y:y + h, x:x + w]
        cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray,
(48, 48)), -1), 0)
        prediction = model.predict(cropped_img)
        maxindex = int(np.argmax(prediction))
        cv2.putText(image, emotion_dict[maxindex], (x + 20, y - 60),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2,
cv2.LINE_AA)
        print(emotion_dict[maxindex])
        filename = 'savedImage.jpg'
        cv2.imwrite(filename, image)
        break

```

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Ихсанова-Солод Жаннель Владимировна

5B070400 – Вычислительная техника и программное обеспечение

ОТЗЫВ НАУЧНОГО РУКОВОДИТЕЛЯ

к дипломному проекту

Тема: «Музыкальный плейлист основанный на эмоциях»

Дипломный проект Ихсановой-Солод Жаннель выполнен самостоятельно, тема работы актуальна, технологии, выбранные для проекта новейшие и современные.

Для выполнения проектной работы было проведено исследование предметной области, для машинного обучения было собрано большое количество данных, которые поспособствовали обучению самой модели машинного обучения.

В первом разделе представлено описание проекта, цели выполняемой работы и также описан анализ предметной области.

Во втором разделе указаны технологии, при помощи которых был выполнен проект. Описан выбор технологий путем сравнения между собой.

В третьем разделе идет описание проектной части дипломного проекта. Представлена архитектура системы взаимодействия. Имеется диаграмма деятельности, которая визуально представляет последовательность действий. Также в разделе представлено проектирование пользовательского интерфейса.

В четвертом разделе был описан функционал проекта, и то, как данные функции были реализованы.

Дипломный проект выполнен с учетом всех требований, предъявляемых к дипломному проекту по специальности 5B070400 – «Вычислительная техника и программное обеспечение», студент Ихсанова-Солод Ж. В. рекомендована к защите дипломного проекта и заслуживает присвоения академической степени «бакалавра» по специальности 5B070400 – «Вычислительная техника и программное обеспечение».

Кушкеев А. Д.



Метаданные


Название
2021_БАК_Ихсанова_Солод_Жаннель_Владимировна (1).docx

Автор Научный руководитель
Ихсанова-Солод Жаннель Айдын Куникеев

Подразделение
ИКИИТ

Список возможных попыток манипуляций с текстом

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв		0
Интервалы		0
Микропробелы		1
Белые знаки		0
Парафразы (SmartMarks)		5

Объем найденных подоби

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



25

Длина фразы для коэффициента подобия 2



3824

Количество слов



30947

Количество символов

Подобия по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и обратите внимание на то, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз


Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	Разработка мероприятий по улучшению условий труда и снижению производственного травматизма в компании «Карачаганак Петролиум Оперейтинг Б. В.» Тастемирова Зульфия Ахметжановна 5/3/2019 Satbayev University (ИХИБТ)	23	0.60 %
2	Разработка мероприятий по улучшению условий труда и снижению производственного травматизма в компании «Карачаганак Петролиум Оперейтинг Б. В.» Тастемирова Зульфия Ахметжановна 5/3/2019 Satbayev University (ИХИБТ)	11	0.29 %

3	Разработка мероприятий по улучшению условий труда и снижению производственного травматизма в компании «Карачаганак Петролиум Оперейтинг Б. В.» Тастемирова Зульфия Ахметжановна 5/3/2019 Satbayev University (ИХИБТ)	11	0.29 %
4	Хибибуллаев Ш. Дипломная работа (для антиплагиата).docx Шухрат Хибибуллаев 5/3/2019 Satbayev University (ИКИИТ)	11	0.29 %
5	Разработка Smart протокола для оптимизации сети.docx Абуев Ерлан Тулегенович 5/17/2017 Satbayev University (ИКИИТ)	10	0.26 %
6	Разработка Smart протокола для оптимизации сети.docx Абуев Ерлан Тулегенович 5/17/2017 Satbayev University (ИКИИТ)	7	0.18 %
7	Хибибуллаев Ш. Дипломная работа (для антиплагиата).docx Шухрат Хибибуллаев 5/3/2019 Satbayev University (ИКИИТ)	6	0.16 %
8	Хибибуллаев Ш. Дипломная работа (для антиплагиата).docx Шухрат Хибибуллаев 5/3/2019 Satbayev University (ИКИИТ)	6	0.16 %
9	Разработка мероприятий по улучшению условий труда и снижению производственного травматизма в компании «Карачаганак Петролиум Оперейтинг Б. В.» Тастемирова Зульфия Ахметжановна 5/3/2019 Satbayev University (ИХИБТ)	5	0.13 %
10	Разработка Smart протокола для оптимизации сети.docx Абуев Ерлан Тулегенович 5/17/2017 Satbayev University (ИКИИТ)	5	0.13 %

из базы данных RefBooks (0.00 %) 

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из домашней базы данных (2.69 %) 

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	Разработка мероприятий по улучшению условий труда и снижению производственного травматизма в компании «Карачаганак Петролиум Оперейтинг Б. В.» Тастемирова Зульфия Ахметжановна 5/3/2019 Satbayev University (ИХИБТ)	50 (4)	1.31 %
2	Хибибуллаев Ш. Дипломная работа (для антиплагиата).docx Шухрат Хибибуллаев 5/3/2019 Satbayev University (ИКИИТ)	31 (5)	0.81 %
3	Разработка Smart протокола для оптимизации сети.docx Абуев Ерлан Тулегенович 5/17/2017 Satbayev University (ИКИИТ)	22 (3)	0.58 %

из программы обмена базами данных (0.00 %) 

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из интернета (0.00 %)



ПОРЯДКОВЫЙ НОМЕР

ИСТОЧНИК URL

КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)

Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР

СОДЕРЖАНИЕ

КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)

Протокол анализа Отчета подобию Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобию, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Ихсанова-Солод Жаннель

Название: 2021_БАК_Ихсанова_Солод_Жаннель_Владимировна.docx

Координатор: Айдын Куникеев

Коэффициент подобию 1: 2.69

Коэффициент подобию 2: 0.00

Замена букв: 0

Интервалы: 0

Микропробелы: 1

Белые знаки: 0

После анализа Отчета подобию констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....

27.05.2021

Дата



.....
Подпись Научного руководителя