МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное

образовательное учреждение высшего образования

«Самарский национальный исследовательский университет

имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики

Кафедра информационных систем и технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«АВТОМАТИЗИРОВАННАЯ СИСТЕМА НАВИГАЦИИ ВНУТРИ ПОМЕЩЕНИЙ С ПОМОЩЬЮ ИНЕРЦИАЛЬНОЙ ТЕХНОЛОГИИ ЛОКАЛЬНОГО ПОЗИЦИОНИРОВАНИЯ МОБИЛЬНЫХ УСТРОЙСТВ»

по направлению подготовки

09.04.01 Информатика и вычислительная техника

(уровень магистратуры)

профиль «Программное обеспечение мобильных устройств»

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.Д. Мавлютов

(подпись, дата)

Руководитель ВКР,

к.т.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.К. Головнин

(подпись, дата)

Нормоконтролер\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Я.В. Соловьева

(подпись, дата)

Самара 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное

образовательное учреждение высшего образования

«Самарский национальный исследовательский университет

имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики

Кафедра информационных систем и технологий

УТВЕРЖДАЮ

Заведующий кафедрой ИСТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.А. Прохоров

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ МАГИСТРА

студенту Мавлютову Владимиру Дмитриевичу

группа № 6223-090401D .

Тема работы: «Автоматизированная система навигации внутри помещений с помощью инерциальной технологии локального позиционирования мобильных устройств»

утверждена приказом по университету №25-Т от «20» января 2020 г.

Исходные данные к работе:

* объект автоматизации: расчет интенсивности движения транспортных потоков;
* типы операционных систем: Android;
* среда программирования: Android Studio;
* техническое обеспечение: Смартфон на базе Android;
* манипулятор – сенсорный экран.

Перечень вопросов, подлежащих разработке в работе:

* анализ предметной области «Анализ навигации в помещении»;
* аналитический обзор систем-аналогов;
* разработка системы построения маршрута внутри помещения с помощью датчиков телефона;
* разработка и исследование алгоритмов инерциальной навигационной системы;
* разработка, отладка и тестирование автоматизированной системы.

Руководитель работы

доцент кафедры ИСТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.К. Головнин

(подпись)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_\_г.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.Д. Мавлютов

(подпись)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_\_г.

РЕФЕРАТ

Пояснительная записка 90 с, 42 рисунка, 4 таблицы, 72 источника, 3 приложения.

Презентация: 23 слайда Microsoft PowerPoint.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА, ИНЕРЦИЛЬНАЯ НАВИГАЦИОННАЯ СИСТЕМА, МАРШРУТ, КАРТА, ДАТЧИКИ СМАРТФОНА, ПАТТЕРН, ШАБЛОН.

Целью выпускной квалификационной работы магистра является разработка автоматизированной системы построения маршрута с помощью датчиков смартфона, таких как акселерометр, гироскоп и магнитометр.

Проведено исследование предметной области «Анализ навигации», рассмотрены задачи навигации внутри помещения и особенности работы датчиков и алгоритмы построения маршрута, применяемые в решении аналогичных задач. Выполнен аналитический обзор существующих систем-аналогов. Разработан метод построения маршрута, с помощью датчиков Android смартфона. Построены диаграммы по методологии UML, создана модель данных, разработаны алгоритмы функционирования и описана архитектура системы. Разработана автоматизированная система навигации внутри помещений с помощью инерциальной технологии локального позиционирования мобильных устройств. Проведены исследования эффективности разработанной системы для решения поставленной задачи. Программное обеспечение системы разработано на языках Java в среде Android Studio.

СОДЕРЖАНИЕ

[Определения, обозначения и сокращения 8](#_Toc41306003)

[Введение 9](#_Toc41306004)

[1 Анализ исследуемой задачи и направления ее решения 12](#_Toc41306005)

[1.1 Анализ предметной области 12](#_Toc41306006)

[1.1.1 Транспортный поток 12](#_Toc41306007)

[1.1.2 Состав транспортного потока 13](#_Toc41306008)

[1.1.3 Классификация транспортных средств 13](#_Toc41306009)

[1.1.4 Задача детектирования признаков 16](#_Toc41306010)

[1.1.5 Акустическое излучение 18](#_Toc41306011)

[1.2 Нейросетевые технологии 22](#_Toc41306012)

[1.2.1 Классификация нейронных сетей 22](#_Toc41306013)

[1.2.2 Модель нейрона 23](#_Toc41306014)

[1.2.3 Функция активации нейрона 24](#_Toc41306015)

[1.2.4 Сверточные нейронные сети и их особенности 25](#_Toc41306016)

[1.3 Алгоритмы дискретного преобразования сигналов 26](#_Toc41306017)

[1.3.1 Дискретное преобразование Фурье 26](#_Toc41306018)

[1.3.2 Быстрое преобразование Фурье 28](#_Toc41306019)

[1.3.3 Дискретное вейвлет-преобразование 29](#_Toc41306020)

[1.4 Мел-частотные кепстральные коэффициенты 31](#_Toc41306021)

[1.4.1 Мел 31](#_Toc41306022)

[1.4.2 Кепстр 33](#_Toc41306023)

[1.4.3 Вычисление мел-частотных кепстральных коэффициентов 34](#_Toc41306024)

[1.5 Описание систем-аналогов 36](#_Toc41306025)

[1.5.1 ShotSpotter 36](#_Toc41306026)

[1.5.2 Shazam 38](#_Toc41306027)

[1.5.3 AudioAnalytics 40](#_Toc41306028)

[1.6 Постановка задачи 41](#_Toc41306029)

[1.7 Описание методологии проектирования системы 42](#_Toc41306030)

[1.8 Выбор комплекса программных средств 42](#_Toc41306031)

[1.8.1 Выбор среды проектирования 43](#_Toc41306032)

[1.8.2 Выбор языка программирования 43](#_Toc41306033)

[1.8.3 Выбор среды программирования 43](#_Toc41306034)

[1.8.4 Выбор системы управления базами данных 44](#_Toc41306035)

[2 Проектирование и программная реализация автоматизированной системы 46](#_Toc41306036)

[2.1 Разработка метода детектирования признаков (паттернов) в аудиофайлах 46](#_Toc41306037)

[2.2 Архитектура системы 49](#_Toc41306038)

[2.3 Проектирование информационно-логической модели системы 51](#_Toc41306039)

[2.3.1 Диаграмма вариантов использования 51](#_Toc41306040)

[2.3.2 Диаграмма классов 53](#_Toc41306041)

[2.3.3 Диаграмма состояний 54](#_Toc41306042)

[2.3.4 Диаграмма взаимодействий 55](#_Toc41306043)

[2.3.5 Диаграмма деятельности 58](#_Toc41306044)

[2.3.6 Разработка логической модели данных 59](#_Toc41306045)

[2.3.7 Разработка физической модели данных 59](#_Toc41306046)

[2.4 Выбор комплекса технических средств 60](#_Toc41306047)

[2.4.1 Расчет объема оперативной памяти 60](#_Toc41306048)

[2.4.2 Расчет объема дискового пространства 61](#_Toc41306049)

[2.4.3 Рекомендованные характеристики технических средств 61](#_Toc41306050)

[2.5 Разработка алгоритмов функционирования системы 62](#_Toc41306051)

[2.6 Разработка и описание интерфейса пользователя 62](#_Toc41306052)

[3 Экспериментальные исследования автоматизированной системы 69](#_Toc41306053)

[3.1 Исследование и визуализация данных 69](#_Toc41306054)

[3.1.1 Набор данных 69](#_Toc41306055)

[3.1.2 Преобразование данных 71](#_Toc41306056)

[3.2 Проведение исследования 72](#_Toc41306057)

[3.2.1 Методика исследования 72](#_Toc41306058)

[3.2.2 Инструменты получения признаков 73](#_Toc41306059)

[3.2.3 Обучение многослойного персептрона 74](#_Toc41306060)

[3.2.4 Обучение сверточной нейронной сети 75](#_Toc41306061)

[3.3 Результаты исследования 77](#_Toc41306062)

[3.4 Описание контрольного примера работы системы 77](#_Toc41306063)

[3.4.1 Назначение 77](#_Toc41306064)

[3.4.2 Исходные данные 78](#_Toc41306065)

[3.4.3 Результаты расчета 79](#_Toc41306066)

[Заключение 81](#_Toc41306067)

[Список использованных источников 83](#_Toc41306068)

[Приложение А Функции активации нейронов 91](#_Toc41306069)

[Приложение Б Руководство пользователя 92](#_Toc41306070)

[Приложение В Листинг программы 97](#_Toc41306071)

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных;

ИТС – интеллектуальная транспортная система;

ИНС – инерциальная навигационная система;

МЛП – многослойный персептрон;

МЧКК – мел-частотные кепстральные коэффициенты;

СНС – сверточная нейронная сеть;

СУБД – системы управления базами данных;

ТП – транспортный поток;

ТС – транспортное средство;

CD – Compact Disc;

GPS – Global Positioning System;

IBM – International Business Machines;

PDF – Portable Document Format;

RBF – Radial Basis Function;

ReLU – Rectified Linear Units;

REST – Representational State Transfer;

SVM – Support Vector Machine;

UML – Unified Modeling Language.

ВВЕДЕНИЕ

Актуальность. На сегодняшний день можно с уверенностью сказать, что информационные технологии «умного города» изменили многое в нашей жизни, и эти положительные изменения с каждым днем все заметней. Одним из примеров развития технологий в сфере управления транспортом является интеллектуальная транспортная система (ИТС), которая позволяет строить безопасную, эффективную и экологичную инфраструктуру современной городской агломерации [1]. Для построения ИТС необходим глубокий анализ различных характеристик транспортного потока (ТП).

Сбор и анализ характеристик ТП возможен лишь при наличии таких технических возможностей [2]. Сбор информации о текущем состоянии ТП осуществляется с помощью различных технических устройств: петлевых датчиков, видеокамер, детекторов. Существует множество алгоритмов обработки видеоинформации в мониторинге транспортных потоков [3]. Такой подход не лишен недостатков, поскольку обработка видеофайлов обладает огромной избыточностью информации, которая требует значительных затрат для обработки, хранения и анализа данных. Стоит отметить, что огромную часть расходов включают в себя такие пункты, как монтаж, эксплуатация и поддержка сети видеонаблюдения. Таким образом, анализ менее избыточного по информации аудиосигнала видится рациональным.

В задаче сбора и анализа аудиоданных накоплен огромный опыт. Больших успехов в области преобразования и анализа сигналов, в том числе и аудиосигналов, достигли такие ученые, как К.Ф. Гаусс, С.А. Прохоров, В.А Котельников. Выполненные исследования в области детектирования признаков в сигналах опираются на разработки ученых и практиков: Е.М. Миркеса, С. Дэвиса, П. Мермельшейтна. Значительный опыт накоплен в области интеллектуальных информационных технологий и разработки систем: Д. Кнут, Д.А. Поспелов и др. Однако данные исследования и труды не концентрировали внимание на применении нейронной сети в качестве классификатора для признаков, полученных из аудиосигналов.

Объектом исследования является процесс извлечения данных из аудиосигналов.

Предметом исследования является методы, модели, алгоритмы и средства детектирования и классификации транспортных средств (ТС) на аудиозаписях.

Целью выпускной квалификационной работы магистра является разработка автоматизированной системы детектирования признаков акустического излучения ТС на аудиозаписях с использованием сверточных нейронных сетей (СНС).

Для достижения цели в работе поставлены и решены следующие задачи:

* анализ исследуемой задачи и современных направлений ее решения;
* проектирование и программная реализация автоматизированной системы, позволяющей осуществлять подсчет интенсивности транспортных потоков за счет детектирования отдельных ТС на аудиозаписях;
* экспериментальные исследования разработанной автоматизированной системы детектирования признаков акустического излучения ТС.

Научная новизна содержится в следующих результатах:

* новый метод детектирования признаков (паттернов) акустического излучения транспортных средств в аудиофайлах путем получения мел-частотных кепстральных коэффициентов в качестве признаков, классифицируемых с помощью СНС;
* алгоритмы функционирования и программное обеспечение автоматизированной системы, реализующие предложенный метод детектирования признаков.

Основной практический результат работы – разработанная автоматизированная система детектирования акустического излучения ТС на аудиофайлах с использованием СНС, которая может применяться для решения задачи анализа характеристик ТП.

Теоретической и методологической основой исследования методы обработки сигналов и распознавания речи, методы и технологии искусственного интеллекта.

Положения, выносимые на защиту:

* новый метод детектирования признаков (паттернов) акустического излучения транспортных средств в аудиофайлах, использующий СНС для классификации по признакам, формируемым мел-частотными кепстральными коэффициентами;
* алгоритмы функционирования и программное обеспечение автоматизированной системы, реализующие предложенный метод детектирования признаков.

Апробация работы. Основные результаты работы докладывались и обсуждались на международных, всероссийских и региональных конференциях: XXVI Международная конференция «Математика. Компьютер. Образование» (Пущино, 2019); VII Всероссийская научная конференция с международным участием «Информационные технологии интеллектуальной поддержки принятия решений» (Уфа, 2019); LXIX Молодежная научная конференция, посвященная 85-летию со дня рождения первого космонавта Земли Ю.А. Гагарина (Самара, 2019); Международная научно-техническая конференция «Перспективные информационные технологии» (Самара, 2019); Международная мультидисциплинарная конференция по промышленному инжинирингу и современным технологиям (Владивосток, 2019); Международная молодёжная научная конференция «XV Королёвские чтения», посвящённая 100-летию со дня рождения Д.И. Козлова (Самара, 2019); Всероссийская научно-техническая конференция «Новые информационные технологии в научных исследованиях» (Рязань, 2019).

Публикации по теме работы. Основные положения и результаты работы изложены в 8 научных публикациях, включающих 2 статьи в изданиях, индексируемых Scopus и Web of Science.

1. Анализ исследуемой задачи и направления ее решения
   1. Анализ предметной области
      1. Способы навигации внутри помещений

Навигация (лат. navigatio, от лат. navigo — «плыву на судне»):

* Определение местоположения, скорости и ориентации движущихся объектов;
* Мореплавание, судоходство;
* Период времени в году, когда по местным климатическим условиям возможно судоходство;
* Основной раздел судовождения, в котором разрабатываются теоретические обоснования и практические приёмы вождения судов.

В течение многих веков термин навигация означал совокупность указанных значений. В XX веке, с развитием науки и техники, появлением воздушных судов, космических кораблей — новых объектов навигации, появились новые смысловые значения термина. Теперь, в общем смысле, навигация — процесс управления некоторым объектом (имеющим собственные методы передвижения) в определённом пространстве передвижения. Состоит из двух основных частей:

* теоретическое обоснование и практическое применение методов управления объектом;
* маршрутизация, выбор оптимального пути следования объекта в пространстве [1].
  + - 1. Навигационная система и её типы

Навигационная система — это совокупность приборов, алгоритмов и программного обеспечения, позволяющих произвести ориентирование объекта в пространстве (осуществить навигацию).

Навигационные системы обеспечивают ориентацию с помощью:

* карт, имеющих видео, графический или текстовый форматы;
* определения местоположения посредством датчиков или других внешних источников;
* спутниковой связи;
* информации от других объектов [2].

Типы навигационных систем:

* Спутниковая система навигации;
* Инерциальная навигация.

По структуре навигационное системы подразделяются на:

* 1. объектовые, к которым относятся:
* бортовые,
* пользовательские (персональные);
  1. распределенные:
* имеющие в своем составе один навигационно-информационный центр;
* многоуровневые (иерархические).

По принадлежности навигационно-информационные системы подразделяются на:

* региональные (районные, организаций, предприятий (корпоративные), подразделений ведомств).
* территориальные (областные, муниципальные, ведомственные);
* федеральные (окружные, федеральные органы исполнительной власти);
* международные.

По условиям применения навигационно-информационные системы подразделяются на:

* мобильные;
* со стационарно размещенными навигационно-информационными центрами.

По виду решаемых задач навигационно-информационные системы подразделяются на:

* навигационно-мониторинговые, предназначенные для контроля перемещений и состояний объектов навигации;
* точной навигации;
* навигационно-управляющие, предназначенные для выработки (формирования) навигационно-временных решений для управления движением и перевода в различные режимы работы отдельных объектов навигации и(или) их групп:
* диспетчерские, предназначенные для использования в системах диспетчерского управления,
* поддержки принятия решений.

По возможностям использования навигационных технологий навигационно-информационные системы подразделяются на:

* навигационно-информационные системы первого типа;
* навигационно-информационные системы второго типа;
* навигационно-информационные системы третьего типа [2].
  + - 1. GSM

GSM (Global System for Mobile Communications) – глобальный стандарт цифровой мобильной сотовой связи, с разделением каналов по времени и частоте. Был разработан в конце 1980-х годов.

Связь возможна на расстоянии не более 120 км от ближайшей базовой станции даже при использовании усилителей и направленных антенн. Поэтому для покрытия определённой площади необходимо большое количество передатчиков [3].

При отсутствии препятствий ослабление сигнала при распространении возрастает пропорционально квадрату расстояния, увеличиваясь, таким образом, на 6 дБ каждый раз, когда расстояние удваивается.

Следует учитывать не только потери при прохождении сигнала в свободном пространстве, но также и воздействия всякого рода препятствий, расположенных между станцией и сотовым телефоном. Например, железобетонные строения способны ослаблять сигналы, проходящие через них, в 100-1000 раз.

На сегодняшний день очень сложно использовать данную технологию для навигации внутри помещений, поскольку точность определения координат низкая, и нет возможности определить уровень над уровнем моря (устройство не сможет понять, на каком этаже здания оно находится), значительное ослабление сигнала из-за железобетонных перекрытий, а также прямая зависимость точности от количества дорогостоящих базовых станций.

* + - 1. Bluetooth/Wi-Fi

Первый способ использования таких данных – решение задачи типа трилатерации. в этом случае исходными данными для оценки координат наблюдаемого объекта служат измерения дальностей «объект-датчик», при этом дальность оценивается по уровню принимаемого мобильным устройством сигнала; такой способ аналогичен используемым, например, в гидроакустике. второй способ – определение местоположения объекта путём сравнения измеряемых уровней сигнала от окружающих датчиков с заранее измеренными значениями, привязанными к карте помещения (карта уровня сигналов, радиоотпечаток). К достоинствам подхода можно отнести возможность использования уже развёрнутых сетей передачи данных (Wi-Fi). недостатком являются высокая погрешность измерения уровня сигнала, дискретность данных карт уровня сигналов, принципиально ограничивающая точность навигации, и необходимость предварительной калибровки и настройки (обучения) системы [4].

* + - 1. GPS

Спутниковая система навигации — это комплексная электронно-техническая система, состоящая из совокупности наземного и космического оборудования, предназначенная для определения местоположения (географических координат и высоты) и точного времени, а также параметров движения (скорости и направления движения и т. д.) для наземных, водных и воздушных объектов.

Основные элементы спутниковой системы навигации:

* Орбитальная группировка, состоящая из нескольких (от 2 до 30) спутников, излучающих специальные радиосигналы;
* Наземная система управления и контроля (наземный сегмент), включающая блоки измерения текущего положения спутников и передачи на них полученной информации для корректировки информации об орбитах;
* Аппаратура потребителя спутниковых навигационных систем («спутниковые навигаторы»), используемая для определения координат;
* Опционально: наземная система радиомаяков, позволяющая значительно повысить точность определения координат;
* Опционально: информационная радиосистема для передачи пользователям поправок, позволяющих значительно повысить точность определения координат.

Принцип работы спутниковых систем навигации основан на измерении расстояния от антенны на объекте (координаты которого необходимо получить) до спутников, положение которых известно с большой точностью. Таблица положений всех спутников называется альманахом, которым должен располагать любой спутниковый приёмник до начала измерений. Обычно приёмник сохраняет альманах в памяти со времени последнего выключения и если он не устарел - мгновенно использует его. Каждый спутник передаёт в своём сигнале весь альманах. Таким образом, зная расстояния до нескольких спутников системы, с помощью обычных геометрических построений, на основе альманаха, можно вычислить положение объекта в пространстве.

Метод измерения расстояния от спутника до антенны приёмника основан на определённости скорости распространения радиоволн. Для осуществления возможности измерения времени распространяемого радиосигнала каждый спутник навигационной системы излучает сигналы точного времени, используя точно синхронизированные с системным временем атомные часы. При работе спутникового приёмника его часы синхронизируются с системным временем, и при дальнейшем приёме сигналов вычисляется задержка между временем излучения, содержащимся в самом сигнале, и временем приёма сигнала. Располагая этой информацией, навигационный приёмник вычисляет координаты антенны. Все остальные параметры движения (скорость, курс, пройденное расстояние) вычисляются на основе измерения времени, которое объект затратил на перемещение между двумя или более точками с определёнными координатами.

В реальности работа системы происходит значительно сложнее. Ниже перечислены некоторые проблемы, требующие специальных технических приёмов по их решению:

* Отсутствие атомных часов в большинстве навигационных приёмников. Этот недостаток обычно устраняется требованием получения информации не менее чем с трёх (2-мерная навигация при известной высоте) или четырёх (3-мерная навигация) спутников; (При наличии сигнала хотя бы с одного спутника можно определить текущее время с хорошей точностью).
* Неоднородность гравитационного поля Земли, влияющая на орбиты спутников;
* Неоднородность атмосферы, из-за которой скорость и направление распространения радиоволн может меняться в некоторых пределах;
* Отражения сигналов от наземных объектов, что особенно заметно в городе;
* Невозможность разместить на спутниках передатчики большой мощности, из-за чего приём их сигналов возможен только в прямой видимости на открытом воздухе.

Спутниковые системы навигации находят применение в следующих областях:

* Геодезия: с помощью систем навигации определяются точные координаты точек;
* Картография: системы навигации используется в гражданской и военной картографии;
* Навигация: с применением систем навигации осуществляется как морская, так и дорожная навигация;
* Спутниковый мониторинг транспорта: с помощью систем навигации ведётся мониторинг за положением, скоростью автомобилей и контроль их движения;
* Сотовая связь: первые мобильные телефоны с GPS появились в 90-х годах. В некоторых странах (например, США) это используется для оперативного определения местонахождения человека, звонящего 911. В России в 2010 году начата реализация аналогичного проекта - Эра-ГЛОНАСС;
* Тектоника, Тектоника плит: с помощью систем навигации ведутся наблюдения движений и колебаний плит;
* Активный отдых: существуют различные игры, где применяются системы навигации, например, Геокэшинг и др;
* Геотегинг: информация, например фотографии «привязываются» к координатам благодаря встроенным или внешним GPS-приёмникам.

В настоящее время работают или готовятся к развёртыванию следующие системы спутниковой навигации:

GPS. Принадлежит министерству обороны США. Этот факт, по мнению некоторых государств, является её главным недостатком. Устройства, поддерживающие навигацию по GPS, являются самыми распространёнными в мире. Также система известна под более ранним названием NAVSTAR.

ГЛОНАСС. Принадлежит министерству обороны России. Система, по заявлениям разработчиков наземного оборудования, будет обладать некоторыми техническими преимуществами по сравнению с GPS. После 1996 года спутниковая группировка сокращалась и к 2002 году практически полностью пришла в упадок. Была полностью восстановлена только в конце 2011 года. К 2025 году предполагается глубокая модернизация системы.

Бэйдоу(Beidou). Развёртываемая Китаем подсистема GNSS предназначена для использования только в этой стране. Особенность - небольшое количество спутников, находящихся на геостационарной орбите.

На 28 декабря 2012 года выведено на орбиту Земли шестнадцать навигационных спутников, из них по предназначению используется.

Согласно планам, к 2012 году она сможет покрывать Азиатско-Тихоокеанский регион, а к 2020 году, когда количество спутников будет увеличено до 35, система «Бэйдоу» сможет работать как глобальная. Реализация данной программы началась в 2000 году. Первый спутник вышел на орбиту в 2007-ом.

Galileo. Европейская система, находящаяся на этапе создания спутниковой группировки. Планируется полностью развернуть спутниковую группировку к 2022 году.

IRNSS. Индийская навигационная спутниковая система, в состоянии разработки. Предполагается для использования только в Индии. Первый спутник был запущен в 2008 году. Общее количество спутников системы IRNSS - 7.

QZSS. Первоначально японская QZSS была задумана в 2002 г. как коммерческая система с набором услуг для подвижной связи, вещания и широкого использования для навигации в Японии и соседних районах Юго-Восточной Азии. Первый запуск спутника для QZSS был запланирован на 2008 г. В марте 2006 японское правительство объявило, что первый спутник не будет предназначен для коммерческого использования и будет запущен целиком на бюджетные средства для отработки принятых решений в интересах обеспечения решения навигационных задач. Только после удачного завершения испытаний первого спутника начнётся второй этап и следующие спутники будут в полной мере обеспечивать запланированный ранее объём услуг [5].

* + - 1. ИНС

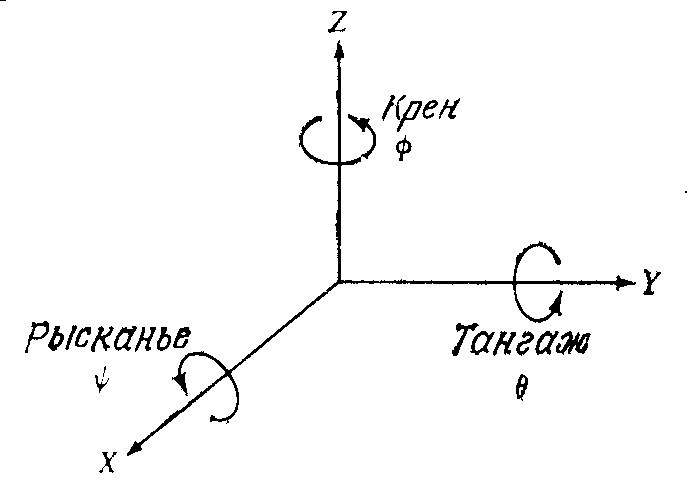
Инерциальная навигация – это метод навигации (определения координат и параметров движения различных объектов – судов, самолётов, ракет и др.) и управления их движением, основанный на свойствах инерции тел, являющийся автономным, т. е. не требующим наличия внешних ориентиров или поступающих извне сигналов. Неавтономные методы решения задач навигации основываются на использовании внешних ориентиров или сигналов (например, звёзд, маяков, радиосигналов и т. п.). Эти методы в принципе достаточно просты, но в ряде случаев не могут быть осуществлены из-за отсутствия видимости или наличия помех для радиосигналов и т. п. Необходимость создания автономных навигационных систем явилась причиной возникновения инерциальной навигации.

Сущность инерциальной навигации состоит в определении ускорения объекта и его угловых скоростей с помощью приборов и устройств, установленных на движущемся объекте, а по этим данным – местоположения (координат) этого объекта, его курса, скорости, пройденного пути и др., а также в определении параметров, необходимых для стабилизации объекта и автоматического управления его движением. Это осуществляется с помощью:

* датчиков линейного ускорения (акселерометров);
* гироскопических устройств, воспроизводящих на объекте систему отсчёта (например, с помощью гиростабилизированной платформы) и позволяющих определять углы поворота и наклона объекта, используемые для его стабилизации и управления движением.
* вычислительных устройств (ЭВМ), которые по ускорениям (путём их интегрирования) находят скорость объекта, его координаты и др. параметры движения;

Преимущества методов инерциальной навигации состоят в автономности, помехозащищенности и возможности полной автоматизации всех процессов навигации. Благодаря этому методы инерциальной навигации получают всё более широкое применение при решении проблем навигации надводных судов, подводных лодок, самолётов, космических аппаратов и других движущихся объектов.

Инерциальные навигационные системы (ИНС) имеют в своём составе датчики линейного ускорения (акселерометры) и угловой скорости (гироскопы или пары акселерометров, измеряющих центробежное ускорение). С их помощью можно определить отклонение связанной с корпусом прибора системы координат от системы координат, связанной с Землёй, получив углы ориентации: рыскание (курс), тангаж и крен (рисунок). Угловое отклонение координат в виде широты, долготы и высоты определяется путём интегрирования показаний акселерометров. Алгоритмически ИНС состоит из курсовертикали и системы определения координат. Курсовертикаль обеспечивает возможность определения ориентации в географической системе координат, что позволяет правильно определить положение объекта. При этом в неё постоянно должны поступать данные о положении объекта. Однако технически система, как правило, не разделяется и акселерометры, например, могут использоваться при выставке курсовертикальной части.

  
Рисунок 1.1 – ИНС в системах координатах

Инерциальные навигационные системы делятся на имеющие гиростабилизированную платформу (ПИНС) и бесплатформенные (БИНС). В платформенных ИНС взаимная связь блока измерителей ускорений и гироскопических устройств, обеспечивающих ориентацию акселерометров в пространстве, определяет тип инерциальной системы. Известны три основных типа платформенных инерциальных систем.

Инерциальная система геометрического типа имеет две платформы. Одна платформа с гироскопами ориентирована и стабилизирована в инерциальном пространстве, а вторая с акселерометрами – относительно плоскости горизонта. Координаты самолета определяются в вычислителе с использованием данных о взаимном расположении платформ.

В инерциальных системах аналитического типа и акселерометры, и гироскопы неподвижны в инерциальном пространстве (относительно сколь угодно далёких звёзд). Координаты объекта получаются в счетно-решающем устройстве, в котором обрабатываются сигналы, снимаемые с акселерометров и устройств, определяющих поворот самого объекта относительно гироскопов и акселерометров.

Полуаналитическая система имеет платформу, которая непрерывно стабилизируется по местному горизонту. На платформе имеются гироскопы и акселерометры. Координаты самолета или иного летательного аппарата определяются в вычислителе, расположенном вне платформы.

В БИНС акселерометры и гироскопы жестко связаны с корпусом прибора. Передовой технологией в производстве БИНС является технология волоконно-оптических гироскопов (ВОГ), принцип действия которых основан на эффекте Саньяка. БИНС на базе таких гироскопов не имеет подвижных частей, абсолютно бесшумна, не требует специального обслуживания и имеет хорошие показатели наработки на отказ (до 80 000 часов у некоторых моделей) и малое энергопотребление (десятки Ватт). Технологии ВОГ пришли на смену лазерно-кольцевым гироскопам (ЛКГ), имеющим подвижные части и требующим периодического обслуживания по калибровке и замене износившихся узлов и деталей, а также с относительно высоким уровнем энергопотребления.

* + 1. Инерциальные технологии позиционирования

Инерциальная технология основана на оценке текущей позиции объекта с учетом его ранее известной позиции, скорости и направления движения. Указанная технология строится на базе цифровых инерциальных датчиков (акселерометров, гироскопов, датчиков углового положения и др.). Для определения положения объекта инерциальная технология использует метод навигационного счисления пути, также называемый методом инерциальной навигации.

Основная задача метода заключается в определении среднего шага, так как длина человеческого шага непостоянна: она зависит от скорости движения объекта, его роста и т. д. Также необходимо распознавать начала каждого нового шага, возникающего в процессе человеческой ходьбы. Человеческая ходьба имеет циклическую природу, и ее можно разделить на 4 основные фазы:

* отталкивание опорной ногой от земли;
* выпрямление переносной ноги;
* наклон голени опорной ноги вперед и начало переноса центра тяжести на переносную ногу;
* отталкивание опорной ноги от пола (после этого она становится переносной) и окончание переноса центра тяжести на переносную ногу, которая, в свою очередь, становится опорной.

Первая и третья фазы цикла ходьбы характеризуются повышением опорной реакции, вторая и четвертая – ее понижением. Факт шага определяется при распознавании каждой из двух этих групп фаз. Для расчета средней длины шага используются различные методы: эмпирические, аппроксимационные, методы, основанные на интегрировании ускорения пройденного шага, и пр.

Также для определения положения объекта необходимо знать направление его движения, которое можно определить на основе показаний, полученных от акселерометра и гироскопа.

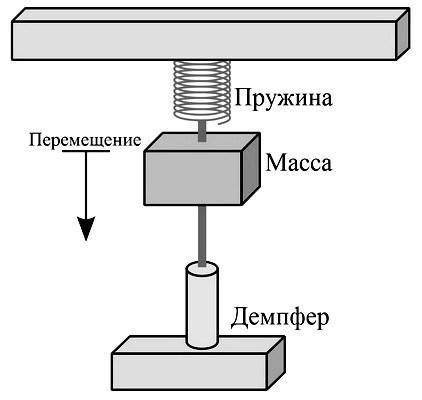
Использование indoor-позиционирования на базе инерциальных датчиков обладает рядом преимуществ:

* автономностью и простотой развертывания. Для определения положения объекта достаточно только инерциальных датчиков и не требуется никакой дополнительной инфраструктуры и оборудования. Учитывая, что большинство современных мобильных устройств связи оборудованы встроенными акселерометрами и гироскопами, нет необходимости в разработке отдельных инерциальных датчиков;
* помехоустойчивостью. Существующие виды помех не влияют на точность измерений.

Главным недостатком инерциальной технологии является накопление погрешности измерений и, как следствие, уменьшение точности позиционирования с течением времени. Указанный недостаток является основной проблемой, возникающей при разработке инерциальных IPS. Для ее устранения применяются различные программные фильтры, но этого, зачастую, оказывается недостаточно. Вследствие этого в системах indoor-позиционирования только инерциальная технология используется редко. Чаще всего для повышения точности позиционирования инерциальную технологию совмещают с другими технологиями позиционирования[13].

* + 1. Датчики мобильных устройств
       1. Акселерометр

Акселерометр – это измерительный прибор позволяющий определить проекцию кажущегося ускорения. В простейшем исполнении он представляет собой грузик, закрепленный на упругом подвесе. При его отклонении от первоначального положения на упругом подвесе можно определить направление изменения положения, а также величину ускорения.

  
Рисунок 1.2 – Схема работы акселерометра

Существует три разновидности акселерометров. Они бывают одноосные, двуосные и трехосные. Наиболее часто используемыми являются трехкомпонентные устройства. Они имеют возможность измерять проекцию кажущегося ускорения в 3-х плоскостях.

Данное оборудование бывает:

* Механическим.
* Электронным.
* Пьезоэлектрическим.
* Термальным.

Механический акселерометр является самой простой и полностью соответствует классической конструкции, которая была придумана изначально. У нее подвешенный груз закрепляется на эластичном подвесе. При изменении положения корпуса прибора под воздействием инерции подвешенное тело компенсирует перекос, тем самым воздействия на пружину на которой оно крепится. В результате специальный механизм определяет подобные колебания и переводит их в показатель линейного ускорения.

Электронные предусматривают совмещение механических частей прибора с датчиками. Они позволяют осуществить более точное и быстрое измерение параметров перемещения положения закрепленной массы. Подобные устройства в разы более компактные, и внешне могут представлять собой миниатюрный чип для микросхемы, габариты которого не превышают размер ногтя на мизинце.

Пьезоэлектрические имеют внутри твердый стержень, который постоянно находится под давлением и воздействует на пьезокристалл. В результате вибрации осуществляется выработка электрического тока. Измеряя параметры напряжения, проводится определение фактических показателей ускорения.

Термальные имеют в своей конструкции миниатюрный пузырек воздуха. При ускорении он отклоняется от своего положения, что фиксируется чувствительными датчиками.

Развитие технологий привело к внедрению акселерометра в различные виды оборудования, позволяя расширить их технические возможности. Если сразу после изобретения подобные датчики применялись только на паровозах с целью определения скорости их движения, то сейчас такие приборы можно встретить повсеместно.

Долгое время акселерометры относились к оборудованию, которое не интересно окружающим. С развитием электронных технологий подобная тенденция пошла на убыль, сделав этот прибор известным среди широких масс. В первую очередь этому поспособствовало появление современных смартфонов, в корпусе которых имеется такое устройство.

Именно благодаря акселерометрам при изменении положения экрана смартфон переводит ориентацию изображения с книжной на альбомную. Впервые данный прибор был применен в мобильном телефоне компанией Nokia. Устройство было установлено в телефон Nokia 5500. Помимо переключения ориентации экрана, акселерометры обеспечивают возможность управления в играх, в частности гонках, где для управления транспортом нужно делать уклоны смартфоном.

При изучении инструкции телефонов, планшетов и прочей мобильной компьютерной техники можно увидеть информацию о наличие так называемого G-датчика. Он и есть тот самый акселерометр.

Именно акселерометр позволяет с помощью специального приложения использовать смартфоном в качестве строительного уровня [6].

* + - 1. Датчик Холла или Магнитометр

Такое приспособление, как датчик Холла – это магнитоэлектрический механизм, принцип которого был впервые открыт физиком Холлом, в честь которого он и был впоследствии назван.

Принцип работы прибора заключается в наличии названого элемента, который связывается с электрической схемой.

 Сам датчик Холла – это такая микросхема, способная на выходе создавать тот или иной информационный сигнал. Зафиксированное магнитное поле – это и есть основа принципа работы данного механизма. Чтобы определить скорость перемещения неподвижных элементов той или иной конструкции, к ней прикрепляют датчик Холла и магниты к подвижной части.

Также движущиеся контакты и части могут просто намагничиваться и не будет необходимости дополнять конструкцию чем-либо. А с целью измерения скорости вращения потребуется несколько постоянных магнитов и сам датчик Холла. В таком случае принцип работы такого приспособления будет следующим:

* в свободном состоянии пластинка будет перемещаться между двумя полюсами;
* также она будет экранировать магнитное поле;
* во время каждого оборота будет происходить электрический импульс в тахометр.

А чтобы увеличить точность показателей нужно увеличить количество используемых магнитов.

Прибор может выдавать такие состояния сигнала:

* единица при наличии сигнала (горящий светодиод);
* ноль при его отсутствии и потухший светодиод.

Эта особенность выдачи сигнала сделала данное устройство незаменимым элементом для мобильных телефонов и прочей цифровой техники. Датчики можно размещать в одном корпусе с логическими элементами и микроконтроллерами, а также цеплять прямо к ним.

Примеры использования данного устройства для телефонов и техники такие:



Рисунок 1.3 – Датчик Холла

Также еще очень широко используются эти приборы в электрических двигателях и в автомобилях в качестве датчиков положения.

Это устройство работает в паре с магнитным полем. Оно подразделяется на две основные категории:

* аналоговое;
* цифровое.

Аналоговый прибор способен перерабатывать индукцию в напряжение, а величина, которую он способен показать, зависит от полярности магнитного поля и силы. Также учитывайте и дистанцию его установки.

Благодаря цифровым приборам можно определить есть ли вообще поле.

Работают они так:

1. Датчик выдаст логическую единицу при условии достижения индукции нужного порога.
2. Если порог не достигается, то выдается логический ноль.
3. Если индукция слабая, а прибор имеет низкую чувствительность, то можно и не зафиксировать поле.

Огромный недостаток цифрового прибора – это наличие нечувствительной меж пороговой зоны.

 Цифровые приборы, в свою очередь, подразделяются биполярные на две категории:

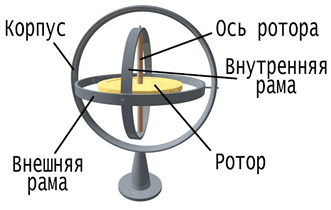
* биполярные датчики Холла, способные реагировать на изменение полярности магнитного поля. Так, одна полярность отключает прибор, а вторая наоборот включает;
* униполярные могут включаться лишь при наличии определенной полярности и выключаются они при снижении индукции.

В машине датчик Холла работает по принципу обычного ключа – замыкателя и размыкателя. Магнит при этом вращается в трамблере и влияет на закрепленный стационарным способом сам датчик. Когда последний начинает «чувствовать» магнитное поле, он начинает подавать импульсы, которые, в свою очередь, вызывают искру для зажигания.

Ключевое преимущество датчиков Холла заключается в том, что при соблюдении допустимых рабочих значений тока и напряжении, его может хватить на огромное количество включений и выключений телефонов, смартфонов, ноутбуков и других приборов [7].

* + - 1. Гироскоп

Гироскоп в телефоне – это специальный датчик, предназначенный для определения положения устройства в пространстве. Его нельзя назвать новым изобретением, так как подобную технологию можно было встретить уже в 19-ом веке. В то время это был довольно громоздкий прибор, представляющий собой круг, вращающийся вокруг оси. Если объяснять более конкретно, то он был похож на детскую юлу или волчка.

  
Рисунок 1.4 – Гироскоп

Понятное дело, что в смартфоны устанавливается совершенно другая конструкция. Это небольшой датчик, который обладает длиной в 3-5 мм, высотой в 5 мм, а шириной в 4 мм. Даже несмотря на столь смешные габариты, многие производители не устанавливают его в свои аппараты, стремясь сделать телефон максимально тонким. Гироскоп вычисляет угол наклона устройства относительно земли, а после передаёт полученные данные операционной системе.

Без подобного датчика было бы сложно играть в игры, особенно гонки, где для управления требуется поворачивать смартфон. Качественные гироскопы настолько точные, что способны определять отклонения на 1-2 градуса. Этого достаточно, чтобы вовремя изменить ориентацию экрана телефона или повернуть игрового персонажа.

Как было сказано выше, главное предназначение гироскопа в телефоне – это определение положения устройства в пространстве. Но зачем системе знать, насколько градусов наклонён смартфон? Ответ на этот вопрос вы можете найти далее, ознакомившись со следующим списком:

* Просмотр видео в 360 градусов. Если у вас есть очки виртуальной реальности, то вы можете просматривать ролики и играть в игры без нажатий по экрану. Все повороты становятся возможными благодаря гироскопу.
* Встряхивание телефона. Без рассматриваемого датчика нельзя было бы использовать функцию, позволяющую разблокировать смартфон после встряхивания.
* Использование навигации. Без гироскопа практически невозможно пользоваться GPS и компасом. Этот датчик позволяет определять стороны горизонта и расположение человека относительно спутника.
* Управление персонажем в играх. Существует огромное количество мобильных игр, где для управления автомобилем или героем нужно поворачивать телефон. Без гироскопа система никак не смогла бы понять положение устройства.

Гироскоп и акселерометр – датчики, предназначенные для определения положения смартфона в пространстве. Самое главное и единственное отличие между ними кроется в принципе считывания данных. Первый компонент высчитывает угол наклона телефона относительно поверхности земли, а после передаёт полученную информацию операционной системе. А вот акселерометр вычисляет ускорение, причём очень точно.

Именно поэтому в качестве шагомера лучше использовать телефон с акселерометром. Полученные данные будут максимально точными, так как датчик учитывает отклонения даже на десятые части миллиметра. Современные производители стараются устанавливать в свои смартфоны как гироскоп, так и акселерометр. Подобное решение является правильным, что исключает случайные повороты экрана устройства при его перемещении [8].

* 1. Описание систем-аналогов

В последнее время всё более актуальной становится проблема навигации внутри помещений, а также предоставления посетителям услуг, основанных на их местоположении и предпочтениях. Здания становятся всё более объёмными и нередко имеют довольно сложную структуру, ориентироваться в которой могут лишь те, кто постоянно посещает такие здания, а для неподготовленного человека ориентирование в таких местах превращается в пытку.

* + 1. ГЛ-ВГ110 - БИНС МЭМС (Микроэлектромеханические системы)

БИНС МЭМС «ГЛ-ВГ110» в этом приборе применены передовые решения повышающие, прежде всего, функциональные возможности его применения, а также основные точностные параметры. Вместо 3-х одноосных акселерометров, применен кластер из 4-х трехосных акселерометров, рассчитанных на повышение точности измерений (за счет избыточности) и масштабируемость в диапазоне ускорений до ±40g.



Рисунок 1.11 – БИНС МЭМС

В состав БИНС введены два новых датчика – баровысотомер (для демпфирования неустойчивого вертикального канала, что особенно актуально для грубых датчиков МЭМС) и трехосный магнитометр (для демпфирования курсового канала).

Добавлен новый высокоскоростной и гальванически развязанный интерфейс Ethernet. Все указанные дополнения выполнены при сохранении габаритно-присоединительных размеров и минимальных изменениях схемы внешних подключений.

БИНС МЭМС "ГЛ-ВГ110" предназначен для наземного, морского и авиационного применения для решения задач навигации (в том числе при пропадании сигнала ГЛОНАСС / GPS — леса, тоннели, при глушении спутниковых сигналов), ориентации, стабилизации, а также контроля других многочисленных параметров (крены, тангажи, удары, горизонтирование и т.д.) [9].

Некоторые варианты применения:

* Обеспечение информацией о координатах объекта при отказе спутниковых систем: спецтранспорт (МВД, МО, МЧС и т.д.), перевозка людей, опасных и ценных грузов, автострахование, автоматическое определение аварийных и опасных ситуаций по характеру движения объекта;
* Автоматизация управления беспилотных систем и объектов / транспорта: наземного, водного, авиационного;
* Управление автоматизированными системами и робототехника для работы в условиях опасных для человека;
* Системы стабилизации линий визирования, подсветки, киносъемочного оборудования и т.д.;
* Управление морскими платформами, манипуляторами и механизмами подверженными внешним возмущениям (качка, течения);
* Системы высокоточного землепользования;
* Системы обнаружения «закладок» в дорожном (железнодорожном) полотне, а также функции предварительной диагностики железнодорожного полотна и контроль отклонения от норм;
* Системы стабилизации диаграммы направленности антенн мобильных систем спутникового телевидения, интернета, телефонной связи;
* Направленное бурение;
* Применение в диагностических внутритрубных снарядах контроля нефте и газопроводов;
* Обеспечение измерения перемещений сотрудника в любом положении тела, в том числе при перемещении ползком, обеспечение точного позиционирования сотрудника в помещениях при неработающей спутниковой системе, обнаружение «нехарактерных» движений сотрудника;
* Множество других вариантов применения, где требуется решение задач оперативного управления, навигации, ориентации и стабилизации.

Минус этой системы состоит в том, что это отдельный блок, который никак не интегрируешь в телефон и представляет интерес для компаний – разработчиков, интеграторов и производителей навигационного оборудования, систем и комплексов стабилизации и ориентации.

* + 1. Navigine Indoor

Navigine Indoor – геолокационный сервис для мониторинга местоположения в зданиях.

Возможности системы:

* Мониторинг во всех помещениях здания в реальном времени;
* Информирование о местоположении персонала в экстренных ситуациях;
* Автоматизированные уведомления о событиях в здании;
* Функция «свой-чужой», основанная на защищённых радиометках;
* Контроль проведения регламентных работ сотрудниками здания;
* Фиксация нарушителя в здании и определение его местоположения;
* Отчеты по перемещениям.

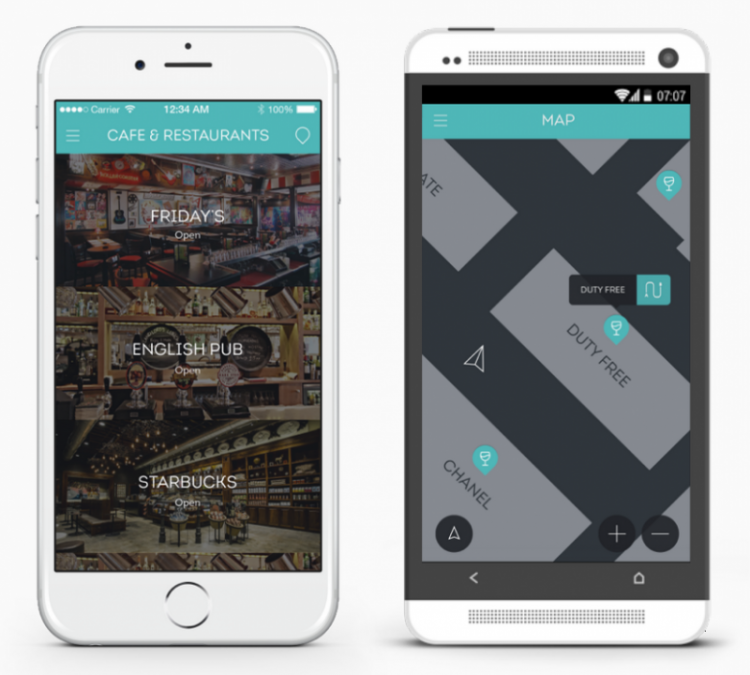


Рисунок 1.13 – Пример работы программы

Задачи, выполняемые мобильным приложением:

* Навигация, информирование и контроль перемещения сотрудников по территории здания с высокой точностью;
* Контроль местонахождения сотрудников на рабочих местах;
* Контроль соблюдения трудовой дисциплины и графиков работы;
* Автоматизированные уведомления о событиях в здании;
* Сигнализация о несанкционированных проникновениях посторонних в здание;
* Координация действий персонала путем направления им команд в режиме текущего времени, в т.ч. в экстренных ситуациях;
* Отчеты по перемещениям.

Главный недостаток системы заключается в том, что для навигации используются метки. Там, где заканчиваются метки, заканчивается, и определятся местоположение. Данный метод не подходит, если осуществлять навигацию в любой точке мира.

* + 1. DaRe

DaRe – это мобильное приложении индийской компании Inertial Element. Само приложение не ведет навигацию и не может определять пройденный маршрут. Для работы приложения требуется модуль, который вешается на ботинок и подключается к телефону с помощью Bluetooth.



Рисунок 1.16 – DaRe

В совокупности смартфона и модуля, система может:

* Позволяет отслеживать местоположение на Google Map;
* Доступ к истинно магнитному полю Земли с компенсацией наклона;
* Позволяет проводить калибровку компаса;
* Предоставляет информацию о трехмерном позиционировании.

Система, состоящая из датчика обуви, и сопрягаемого приложения Android DaRe. Данные из нескольких датчиков отбираются, объединяются, калибруются с компенсацией и подвергаются навигационным вычислениям для получения информации о смещении и курсе. Затем датчик передает запрошенные данные в ответ на соответствующую команду от DaRe. DaRe, в свою очередь, строит отслеживаемый путь, используя пошаговые данные от датчика положения.

К недостаткам данной системы необходимо отнести датчик обуви. Потому что это очень неудобно и не практично, так как отдельный датчик — это повышение стоимости и сложности системы.

* 1. Математическое описание объекта исследования

Существует несколько вариантов получения информации для построения пути: с помощью данных с акселерометра, магнитометра или гироскопа. Таким образом можно получить различные множества данных.

Так, с помощью акселерометра можно получить ускорение по трем осям. В итоге получается множество данных:

– это данные с акселерометра.

– это точка одного измерения в пространстве.

С помощью гироскопа можно получить гироскопический момент вращения ротора, возникающий при изменении направления оси. Он определяется следующим множеством:

– данные с гироскопа, а именно угловая скорость относительно оси симметрии.

– точка одного измерения для определения пространства.

Для определения магнитных полей используется магнитометр. В итоге получается множество данных:

– данные с гироскопа, а именно индукция магнитного поля Земли в определённой точке.

– точка одного измерения для определения пространств.

Результат обработки данных должен выглядеть как множество точек на карте, из которых строится маршрут. Так, например:

–это координаты точки на карте, где – это широта, а – долгота.

– это множество точек, из которых строится маршрут.

Из физики известно, что ускорение является первой производной от скорости, то есть характеризует быстроту ее изменения. Соответственно, скорость – это первая производная расстояния. Операцией, обратной дифференцированию (взятию производной), является интегрирование. Следовательно, если значение производной (измеренное ускорение) известно, то после его интегрирования получим скорость, а после интегрирования скорости получим пройденное расстояние.

Пусть  и  – измеренные ускорения по направлениям на север и восток,  и  – составляющие путевой скорости,  и  – пройденные расстояния по этим же направлениям. Тогда

Современные ИНС осуществляют счисление в географической системе координат, то есть определяют широту и долготу. Если принять Землю за сферу, то текущие широта и долгота (в радианах) могут быть определены как:

где  – радиус Земли,

, – начальные координаты, получаемые с магнитометра.

Все расчеты выполняются цифровыми вычислителями, а именно процессором телефона.

* 1. Постановка задачи

Основным преимуществом инерциальных навигационных систем (ИНС) является их автономность, делающая возможным определение пространственного положения в условиях отсутствия спутникового сигнала, что особо актуально в современной городской плотной застройке, а также внутри помещений. Использование технологии ИНС для решения бытовых задач резонно выполнять с использованием смартфонов, получивших широкое распространение. Разрабатывается бесплатформенная ИНС, функционирующая с использованием аппаратных средств смартфона (акселерометр, гироскоп, магнетометр) и обладающая следующими функциональными возможностями:

* определение и отображение направления, в котором осуществляется движение;
* осуществление замеров расстояния, на которое переместилось мобильное устройство;
* определение местоположения при указании координат начальной точки.

Каждое изменение положения смартфона регистрируется за счет обработки непрерывного потока данных с датчиков смартфона, а именно углов ориентации. Используемый принцип навигации основан на измерении ускорения смартфона по осям системы координат. Ускорение измеряется с помощь акселерометра. Для геомагнитного позиционирования используется магнетометр, позволяющий определять направление. Датчик гироскопа позволяет определить, насколько смартфон отклонился относительно силы тяжести. Интеграция в единую измерительную сеть магнитометра, акселерометра и гироскопа позволяет сформировать полноценную ИНС, определяющую местоположение в пространстве. Бесплатформенная ИНС разрабатывается на языке Java под операционную систему Android для апробации математического аппарата, заложенного в основу решения. Таким образом, предлагаемый подход позволит вести навигацию без доступа к сети интернет и спутниковой связи, например, в зданиях

* 1. Описание методологии проектирования системы

UML (сокр. от англ. Unified Modeling Language – унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML моделью. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем [10].

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графический конструкций – диаграмм. Процесс объектно-ориентированного анализа и проектирования неразрывно связан с процессом построения диаграмм. При этом совокупность построенных диаграмм является самодостаточной в том смысле, что в них содержится вся информация, которая необходима для реализации проекта сложной системы.

В процессе разработки UML-проекта осуществляется постепенный переход от анализа и проектирования к описанию реализации программы.

* 1. Выбор комплекса программных средств

Комплекс программных средств для разработки системы включает в себя среду проектирования, язык программирования, интегрированную среду разработки, набор тестовых и обучающих данных.

* + 1. Выбор среды проектирования

В качестве среды проектирования информационно-логической модели выбран инструмент VisualParadirmOnline. Он является векторным графическим редактором диаграмм и блок-схем и поддерживает большое количество различных нотаций. Основное преимущество данного инструмента является его открытый исходный код, что делает инструмент бесплатным и надежным, а также возможность получения диаграмм онлайн.

* + 1. Выбор языка программирования

В качестве языка программирования был выбран универсальный объектно-ориентированный язык со строгой типизацией – Java. Основным преимуществом является то, что на нем пишут мобильные приложения на Android, а также самый востребованный язык программирования [11]. Программировать можно практически на всех платформах, язык хорошо спроектирован, логичен и лаконичен.

* + 1. Выбор среды программирования

В качестве среды программирования был выбрана Android Studio. Android Studio, основанная на программном обеспечении [IntelliJ IDEA](https://ru.wikipedia.org/wiki/IntelliJ_IDEA" \o "IntelliJ IDEA) от компании [JetBrains](https://ru.wikipedia.org/wiki/JetBrains" \o "JetBrains) является официальным средством разработки Android приложений. А в 2017 году компания Google рекомендовала ее как оптимальное средство для создания приложения под Android. Почти все Android-девелоперы используют Android Studio. Если у вас возникнут проблемы в работе, сможете обратиться к коллегам на форумах или тематических сайтах — вам помогут.

* + 1. Выбор системы управления базами данных

Выбор базы данных (БД) является важным этапом в проектировании системы, потому что БД предопределяет множество нюансов на процессе разработки, а также дальнейшем

В качестве системы управления базами данных (СУБД) была выбрана SQLite. SQLite – это компактная встраиваемая СУБД с открытым исходным кодом.

SQLite — это внутрипроцессная библиотека, которая реализует автономный, безсерверный, транзакционный механизм базы данных SQL с нулевой конфигурацией. Это база данных с нулевой конфигурацией, что означает, что, как и другие базы данных, вам не нужно настраивать ее в вашей системе [12].

1. Проектирование и программная реализация автоматизированной системы
   1. Разработка метода детектирования признаков (паттернов) в аудиофайлах

Для того, чтобы эффективно и точно распознавать необходимые аудиособытия, был предложен следующий метод, состоящий из нескольких шагов, схема которого представлена на рисунке 2.1.



Рисунок 2.1 – Алгоритм детектирования признаков

На первом шаге происходит преобразование исходного аудиосигнала в набор фреймов с перекрытием. На рисунке 2.2 представлен пример буферизации с перекрытием.

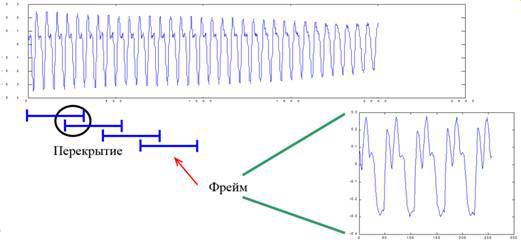


Рисунок 2.2 – Буферизация с перекрытием

Второй шаг представляет из себя предобработку. Обычно, эта стадия включает в себя фильтрацию и оконное взвешивание. Данный шаг необходим для спектрального сглаживания сигнала. В таком случае сигнал становится менее восприимчивым к различным шумам, возникающим в процессе обработки.

В разработанной системе в качестве предобработки производится передискретизация аудиозаписи, а также преобразование к одному к каналу. Также происходит нормализация разрядности сигнала, так что значения варьируются от -1 до 1. Это убирает сложность обработки данных, потому что различные аудиофайлы могут иметь различный диапазон разрядности.

Третий шаг – извлечение необходимых признаков. Существует множество подходов к извлечению признаков из аудиосигнала. Главная цель данных методов – уменьшение избыточности сигнала и выделение наиболее релевантной информации, и избавление от нерелевантной. Обычно, признаки, описывающие аудиосигнал для получения той или иной характеристики комбинируются в один вектор, на основе которого и происходит процесс обучения и дальнейшая классификация с использованием выбранной обученной модели. Далее, на рисунке 2.3 представлены наиболее популярные признаки, выделяемые из аудиосигнала:

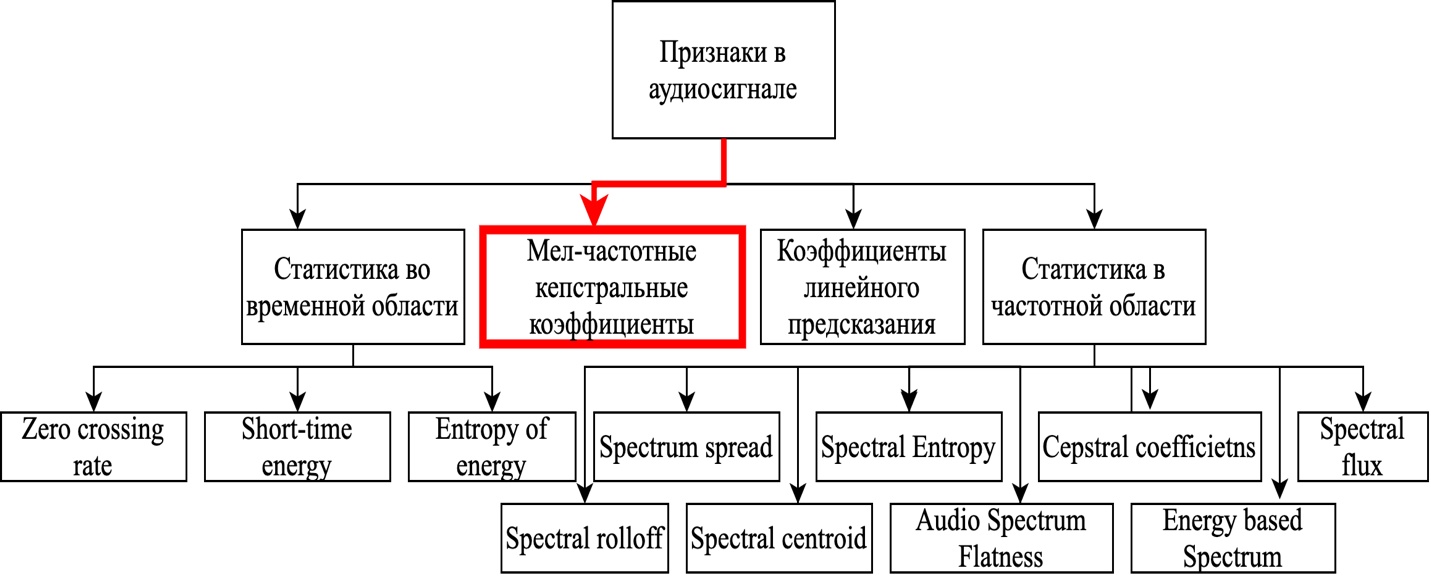


Рисунок 2.3 – Классификация методов получения   
признаков в аудиосигнале

В качестве инструмента извлечения признаков предложено использовать извлечение МЧКК из аудиофайлов. МЧКК суммируют распределение частот по размеру окна, поэтому можно анализировать как частотные, так и временные характеристики звука. Эти аудио представления позволят нам определить признаки, необходимые для классификации.

На четвертом шаге происходит постобработка признаков. После извлечения необходимых признаков сигнала для их дальнейшего использования производится нормализация признаков так, чтобы каждый компонент вектора признаков имел среднее значение и стандартное отклонение 1.

Одним из преимуществ использования методов сокращения размерности является способность существенно увеличить скорость и точность процесса обучения, и точность работы алгоритмов машинного обучения, за счет избавления от избыточного количества признаков и выделения существенных признаков. В разработанной системе на данном шаге используется метод главных компонент, так как он позволяет сократить размерность вектора признаков за счет выделения независимых компонент, максимальным образом покрывающий разброс по всем событиям.

На последнем пятом шаге предложенного метода происходит выбор обучающей модели. Для различных типов аудиособытий стоит подбирать конкретный классификатор, потому что это может обеспечить большой прирост точности в процессе классификации.

На рисунке 2.4 представлена схема классификаторов, применяемых для распознавания аудиособытий.

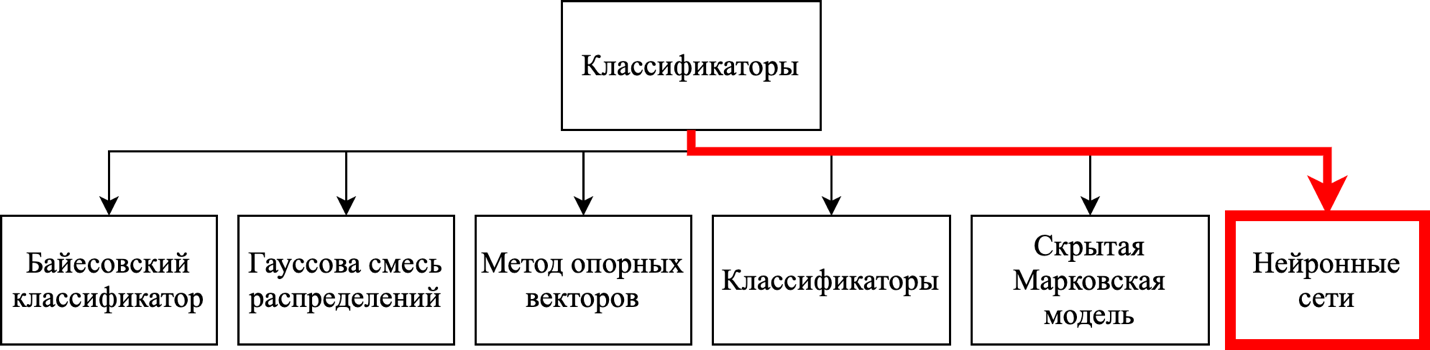


Рисунок 2.4 – Виды классификаторов

В качестве классификатора используются СНС. Слои сети организованы в трех измерениях: ширина, высота и глубина. Узлы в одного слоя не обязательно соединяются со всеми узлами следующего слоя, а зачастую лишь с какой-то его частью. Оптимизатором модели выступает Adam (adaptive moment estimation) – оптимизационный алгоритм, сочетающий в себе и идею накопления движения и идею более слабого обновления весов для типичных признаков [58]. Сверточные слои разработаны для обнаружения признаков. Алгоритм работы заключается в скольжении окна фильтра по входным данным, выполнении умножения матриц и сохранении результата в карте объектов. Данная операция называется операцией свертки. В качестве функции активации для сверточных слоев используется функция ReLU.

Каждый сверточный слой связан с субдискретизирующим слоем. Субдикретизирующий слой уменьшает размерность модели, а также сокращает количество параметров и последовательных вычислений. Данная оптимизация сокращает количество времени на обучение, а также сокращает переобучение.

* 1. Архитектура системы

Основой архитектуры системы являются микросервисы. Микросервисы – это новый архитектурный стиль, который пытается преодолеть недостатки централизованной монолитной архитектуры [56], в которой логика приложения заключена в большие развертываемые части файлы. Сами по себе микросервисы это подход к разработке отдельного приложения в виде набора небольших сервисов, каждый из которых работает в своем собственном процессе и обменивается данными с легковесными механизмами, часто под собой подразумевающий RESTful API [57]. В отличие от монолитов, микросервисы способствуют независимому развертыванию и масштабируемости и могут быть разработаны с использованием различных стеков технологий [58].

Хотя микросервисы можно рассматривать как эволюцию сервис-ориентированной архитектуры, однако они являются архитектурно различными. В отношении совместного использования и повторного использования: с учетом того, что повторное использование сервиса часто оказывалось не столь ожидаемого, вместо повторного использования уже существующего микросервиса для новых задач или вариантов использования, ведь они должны быть небольшими и достаточно независимыми, чтобы можно было быстро разработать новый, который может сосуществовать, развивать или заменять предыдущий в соответствии с потребностями задачи.

Схема архитектуры системы представлена на рисунке 2.5.

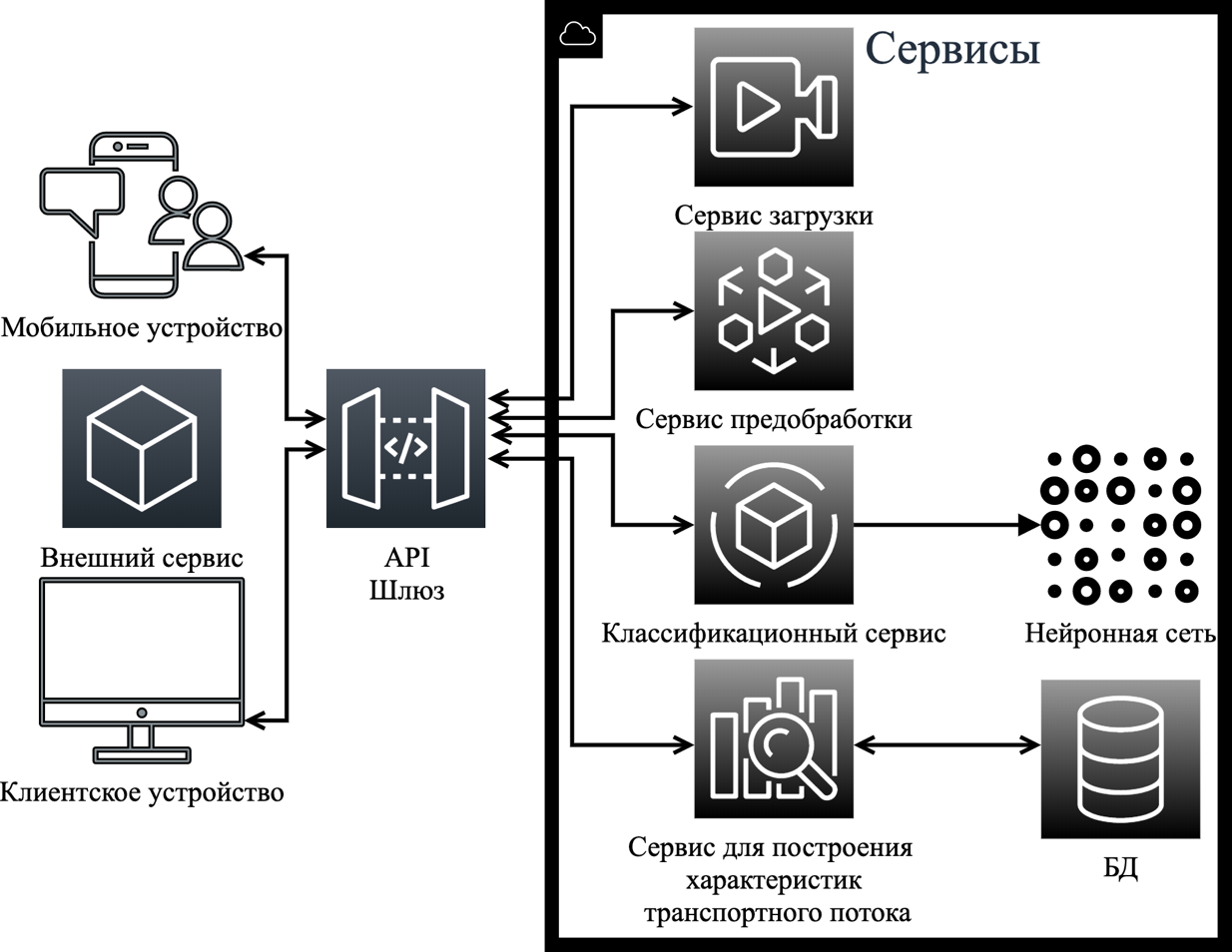


Рисунок 2.5 – Архитектура системы

Таким образом, микросервисная архитектура обеспечивает следующие преимущества:

* независимая разработка: маленькие независимые компоненты могут создаваться маленькими независимыми командами. Объём времени, необходимого для изучения компонента, значительно снижается, и разрабатывать новые функции становится проще.
* независимое развёртывание: каждый отдельный компонент можно разворачивать независимо. это позволяет выпускать новый функционал быстро и с меньшими рисками.
* независимая масштабируемость: каждый компонент можно масштабировать независимо от другого. Это позволяет обеспечить необходимую устойчивость системы меньшими затратами.
* возможность повторного использования: компоненты реализуют свою маленькую конкретную функцию. Это означает, что их проще адаптировать для использования в других системах, сервисах или приложениях.
  1. Проектирование информационно-логической модели системы
     1. Диаграмма вариантов использования

Диаграмма вариантов использования описывает функциональное назначение системы, то есть то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Разработка диаграммы вариантов использования преследует цели:

* определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
* сформулировать общие требования к функциональному поведению проектируемой системы;
* разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
* подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актантом (actor), взаимодействие которых с системой отображается в виде так называемых вариантов использования. При этом актантом (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая служит источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру или прецедентов использования системы. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актантом.

Помимо актантов и вариантов использования, данная диаграмма содержит отношения – описывающие взаимодействия экземпляров одних актантов и вариантов использования с экземплярами других актантов, и вариантов использования. В языке UML имеется несколько стандартных видов отношений между актантами и вариантами использования, такие как: ассоциации, расширения и включения.

Диаграмма вариантов использования представлена на рисунке 2.6

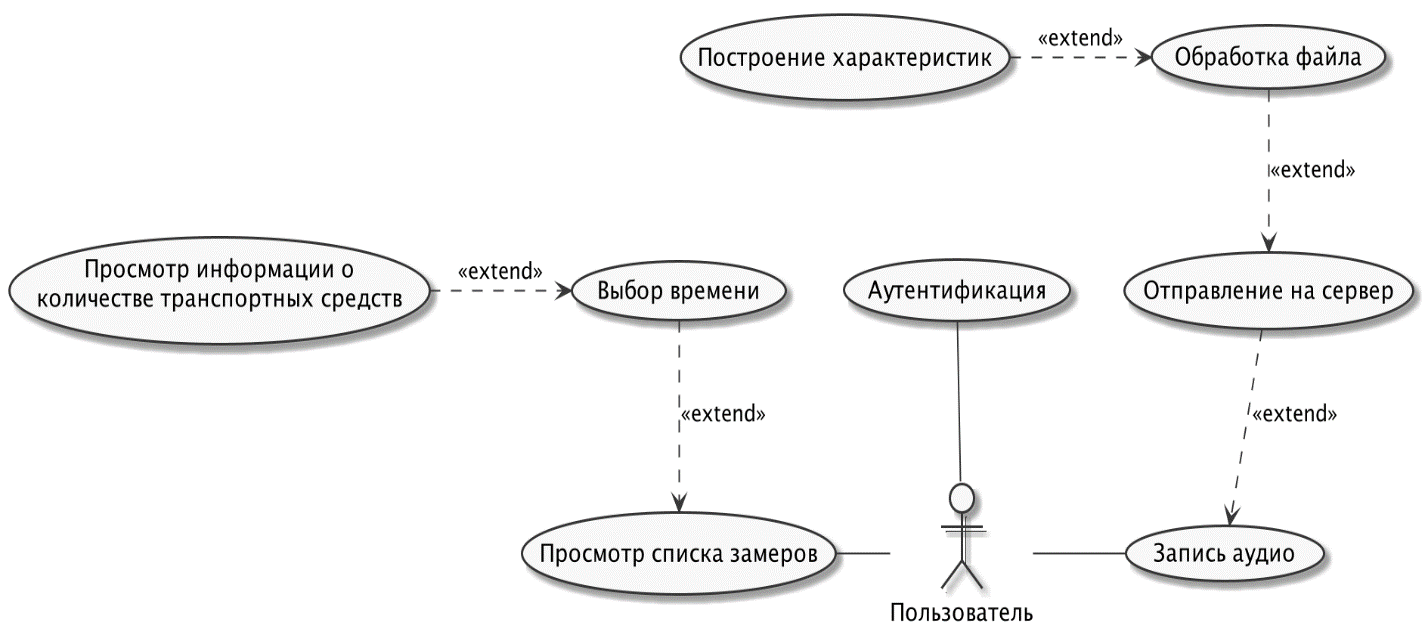


Рисунок 2.6 – Диаграмма вариантов использования

Пользователь имеет возможность загружать аудиозапись, чтобы далее она была обработана на сервере, с последующим вычленением всех необходимых характеристик и сохранении информации в базе данных. Пользователю также доступен выбор точек на карте, где происходила запись аудиофайла. При просмотре данных точек доступна информация о перечне ТС, которые проезжали в указанный промежуток времени.

* + 1. Диаграмма классов

Базовыми элементами автоматизированной системы являются классы. Класс – это шаблон, который описывает множество однотипных объектов, имеющих определенный набор параметров и характеристик – атрибуты, операции, отношения и семантику.

В нотации UML классы системы отображаются на диаграмме классов. Диаграмма классов представляет собой набор классов, интерфейсов, их атрибутов и методов, а также взаимосвязей между этими классами [59]. Данная диаграмма предназначена для определения типов объектов разрабатываемой системы и для описания их отношений.

Диаграмма классов системы представлена на рисунке 2.7.

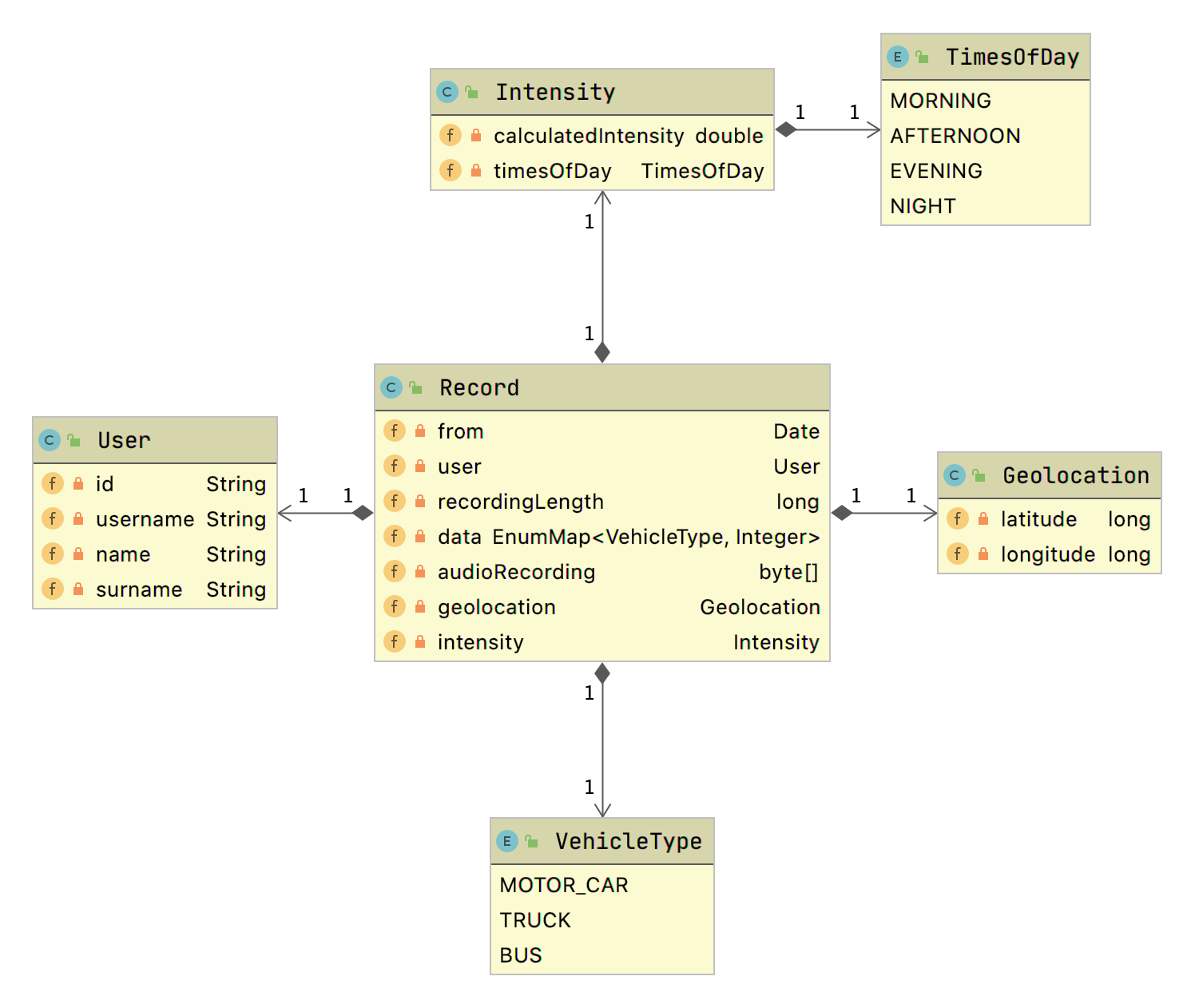


Рисунок 2.7 – Диаграмма классов

Наиболее важными среди связей диаграммы классов являются следующие:

* зависимость – отношение использования, при котором изменение объекта одного класса влияет на использующий его объект другого класса;
* обобщение, представляющее собой отношение между родительской сущностью и ее потомком;
* ассоциация, показывающая какую-либо связь между объектами различных классов;
* агрегация – это разновидность ассоциации, представляющая собой отношение типа «часть-целое»;
* композиция, являющаяся типом агрегации, при которой объект-часть принадлежит только единственному целому.
  + 1. Диаграмма состояний

Главное предназначение этой диаграммы – показать поведение одного объекта в течение его жизни, начиная от создания объекта и заканчивая его уничтожением. Каждая диаграмма состояний представляет некоторый автомат. С ее помощью представляется поведение сущностей во времени основываясь на восприятии ими каких-либо событий. Системы, которые реагируют на внешние действия от других систем или от пользователей, иногда называют реактивными. Если такие действия инициируются в произвольные случайные моменты времени, то говорят об асинхронном поведении модели.

Диаграмма состояний по существу является графом специального вида, который представляет некоторый автомат. Вершины этого графа являются состояния (или псевдосостояния). Дуги графа служат для обозначения переходов из состояния в состояние. Диаграммы состояний могут быть вложены друг в друга, образуя вложенные диаграммы более детального представления отдельных элементов модели. Диаграмма состояний разработанной системы представлена на рисунке 2.8.

Под состоянием понимается абстрактный метакласс, используемый для моделирования отдельной ситуации; при этом имеет место выполнение некоторого условия. Состояние может быть задано в виде набора конкретных значений атрибутов класса или объекта, которые отражают динамический или функциональный аспект его поведения. При этом изменение их отдельных значений будет отражать изменение состояния моделируемого класса или объекта.

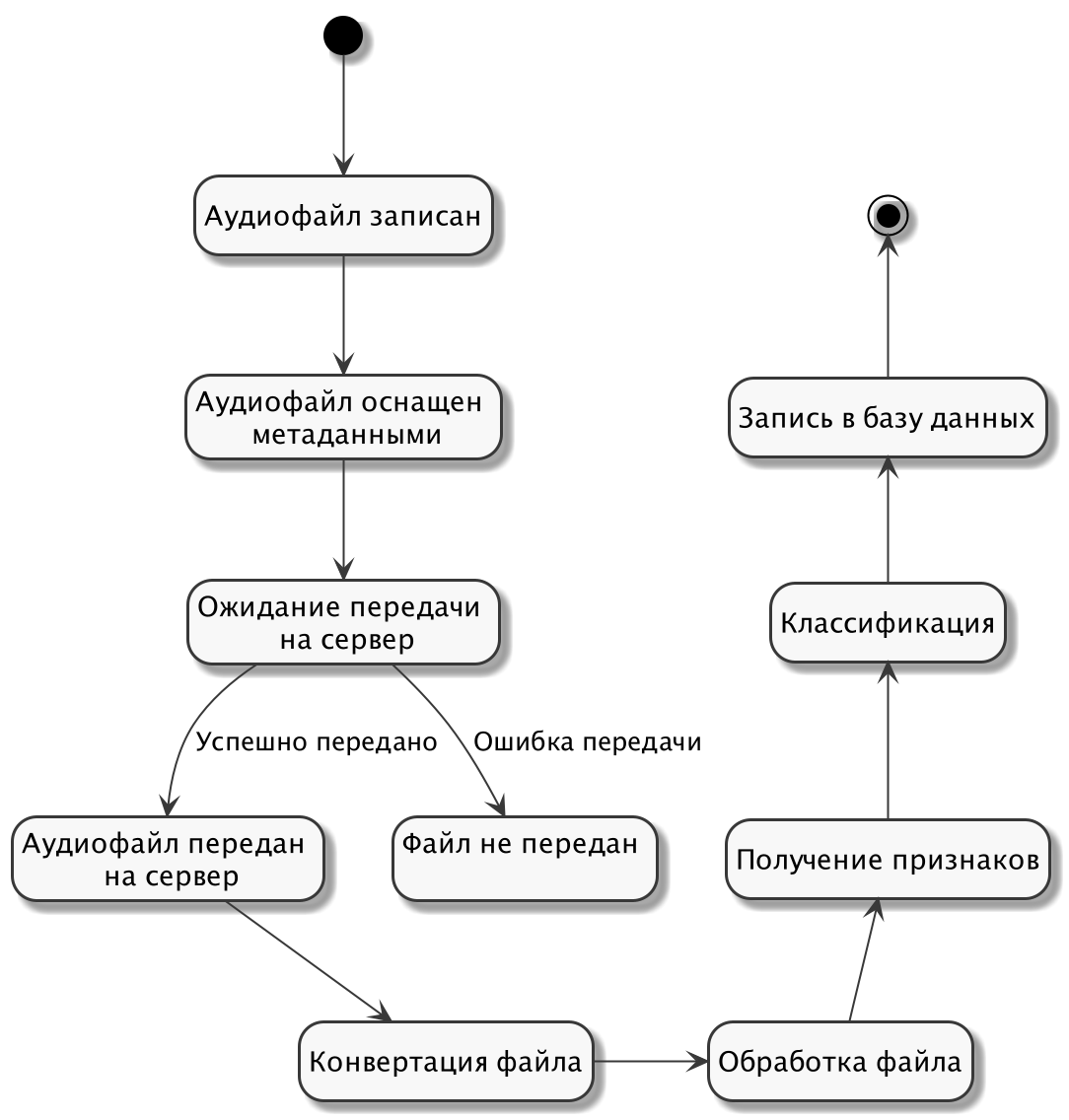


Рисунок 2.8 – Диаграмма состояний

* + 1. Диаграмма взаимодействий

Для моделирования взаимодействия объектов в языке UML используются советующие диаграммы взаимодействия. Так, взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

На диаграмме последовательности изображаются объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения.

Одно – слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии. Данный вид отображений называют swimlane или плавающая дорожка. Второе – вертикальная временная ось, направленная сверху вниз. При этом взаимодействие объектов реализуется посредством сообщений, которые посылаются от одних объектов к другим и также образуют порядок по времени своего возникновения. Удобно на диаграмме последовательности демонстрировать динамику сообщений и вызовов. При это видно: какой объект создается и в какой момент времени, к каким объектам адресуются вызовы и как они располагаются во времени [60]. На рисунке 2.9 по казан пример диаграммы последовательности.

* Для того, чтобы пользователь совершил анализ звуковой записи ТП, ему необходимо загрузить файл на сервер, где развернуто приложение. Если сервер работает в нормальном режиме и может принять аудиозапись, он посылает запрос пользователю и начинается загрузка файла. По истечении загрузки файла, происходит сохранение записи со всем необходимыми данными в базу данных, далее происходит предобработка и постобработка файла, получение необходимых признаков. После происходит классификация с использованием нейронной сети. Результаты классификации передаются в сервис подсчета интенсивности движения в соответствии с алгоритмами расчета интенсивности движения. Подсчитанная интенсивность сохраняется в базе данных, а полученная информация посылается пользователю. Пользователю представлена информация об интенсивности движения на том участке движения, где была произведена запись звука ТП.

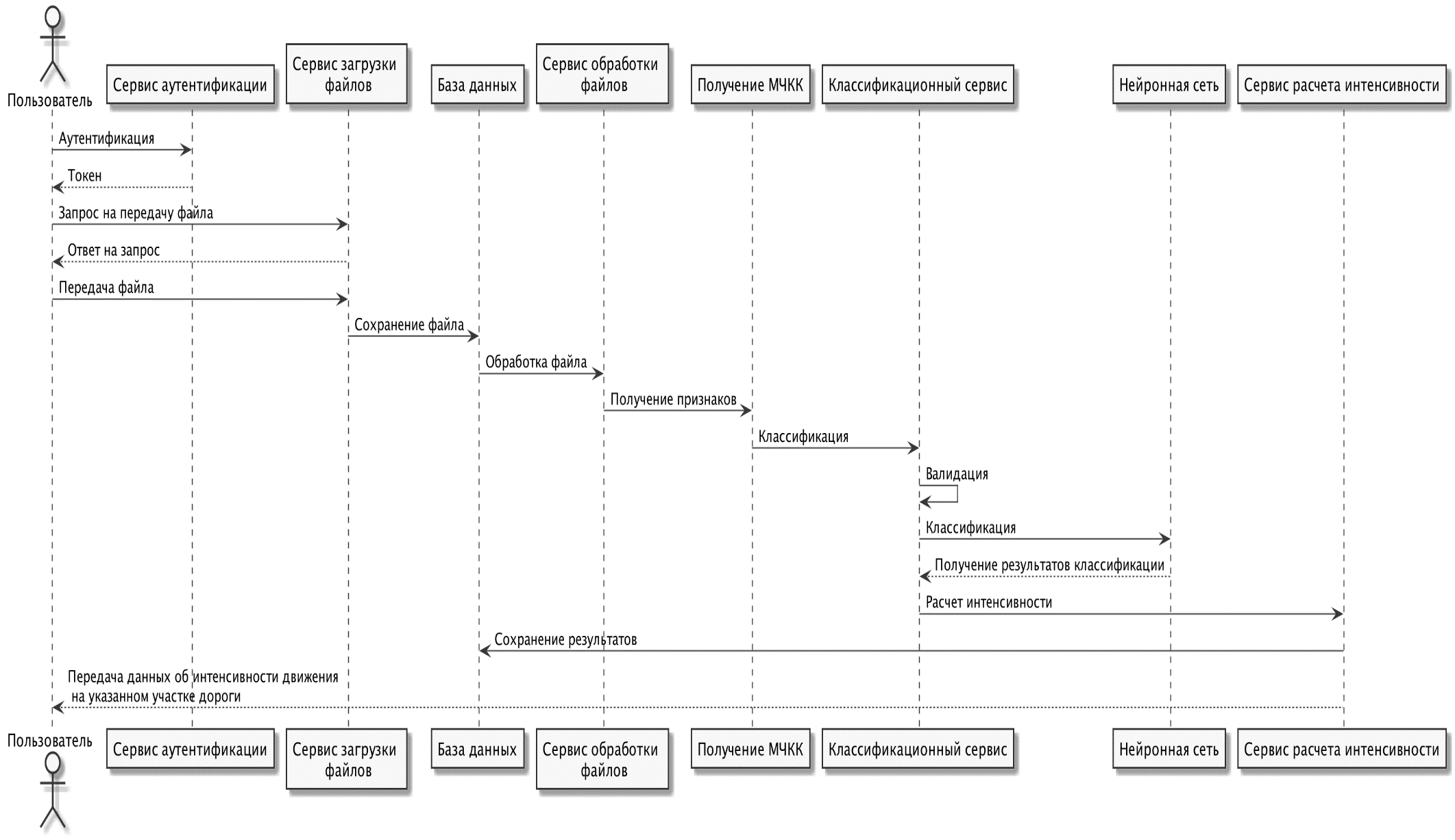


Рисунок 2.9 – Диаграмма последовательности

* + 1. Диаграмма деятельности

Диаграммы деятельности используются для моделирования процесса выполнения операций. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой операции в предыдущем состоянии. Состояние действия – специальный случай состояния с некоторым входным действием и, по крайней мере, одним выходящим из состояния переходом.

На рисунке 2.10 представлена диаграмма деятельности для разработанной системы.



Рисунок 2.10 – Диаграмма деятельности

Этот переход неявно предполагает, что входное действие уже завершилось. Состояние действия не может иметь внутренних переходов, поскольку оно является элементарными. Обычное использование состояния действия заключается в моделировании одного шага выполнения алгоритма (процедуры) или потока управления. Каждая диаграмма активностей должна иметь единственное начальное значение и единственное конечное состояния. При этом каждая деятельность начинается в начальном и заканчивается в конечном состоянии [61].

* + 1. Разработка логической модели данных

Логическая модель базы данных является моделью базы данных системы, представленной в виде совокупности сущностей, атрибутов и связей. Логическая модель позволяет графически представить структуру базы данных, которая не зависит от аппаратной платформы и конечной реализации базы данных.

В процессе проектирования системы была разработана логическая модель базы данных, для описания которой была выбрана методология IDEF1X. На рисунке 2.11 представлена логическая модель, содержащая основные сущности базы данных системы.

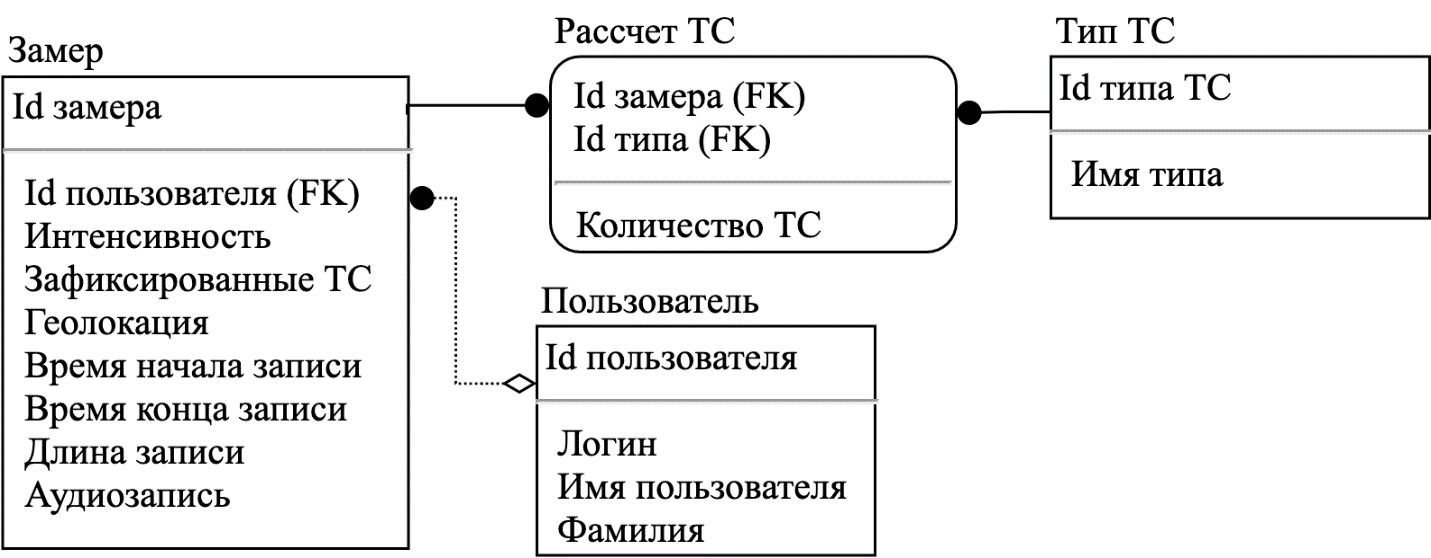


Рисунок 2.11 – Основные сущности логической модели данных

* + 1. Разработка физической модели данных

Под физической моделью базы данных понимают модель, определяющую способы размещения данных в среде хранения и способы доступа к этим данным. Физическая модель позволяет описывать все детали, которые необходимы для создания базы данных в конкретной СУБД: названия таблиц и столбцов, определения первичных и внешних ключей типы данных, и т.д.

Физическая модель базы данных, представленная на рисунке 2.12, построена на основе логической модели из п. 2.3.6.

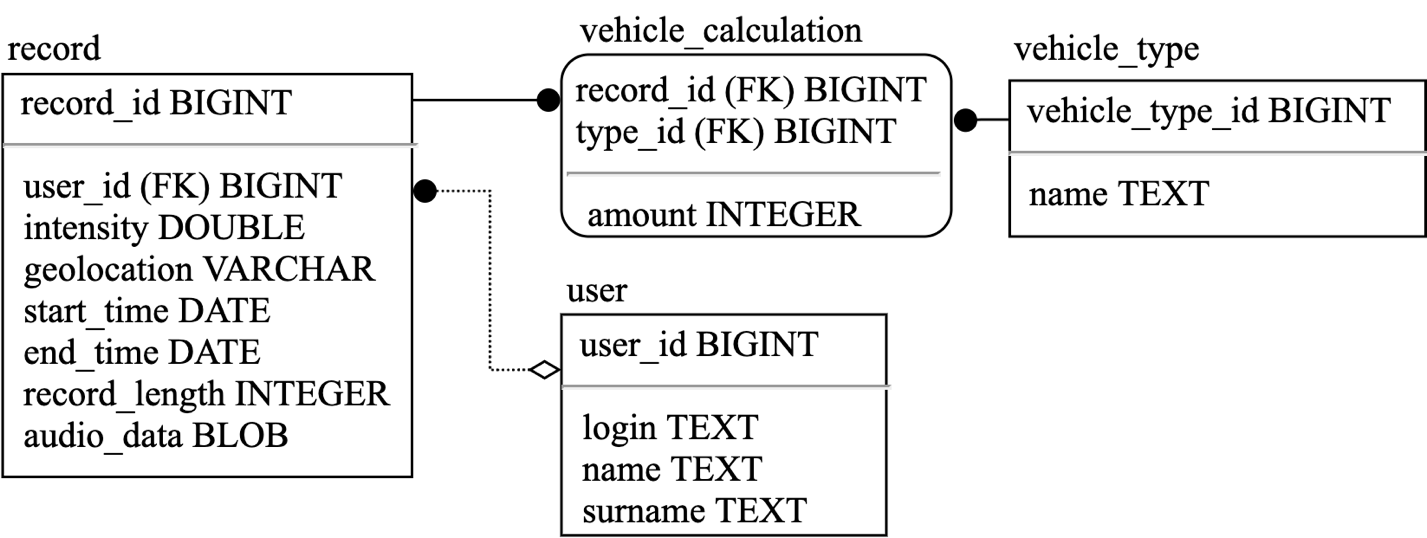


Рисунок 2.12 – Физическая модель данных

* 1. Выбор комплекса технических средств
     1. Расчет объема оперативной памяти

Объем оперативной памяти, необходимый для нормальной работы системы, рассчитывается в соответствии со следующей формулой:

,(19)

где – объем оперативной памяти, занимаемый операционной системой;

– объем оперативной памяти, необходимый для нормальной работы системы, определяемый объемом памяти запущенной системы, равный 40Мб;

– объем данных БД, который может быть одновременно загружен в оперативную память.

Расчет объема оперативной памяти проведен, исходя из предположения, что в качестве операционной системы сервера (ОС) используются наиболее распространенные в настоящее время ОС Windows 8 и Windows 10, таким образом: = 750 Мб. Общий объем ОЗУ составляет = 750 + 25 + 100 = 875 Мб.

* + 1. Расчет объема дискового пространства

Для расчета объема дискового пространства, необходимого для нормальной работы системы, использована следующая формула:

,(20)

где – объем оперативной памяти, необходимый для нормальной работы операционной системы;

– объем оперативной памяти, необходимый для нормальной работы системы (включает объем памяти для работы с базой данных);

–­ объем памяти, занимаемый всем необходимым сопутствующим программным обеспечением;

– объем памяти, занимаемый базой данных (всеми таблицами) при ее максимальном заполнении;

– объём памяти необходимый для хранения файла справки.

Расчет необходимого дискового пространства проведен, исходя из того, что в качестве ОС используются наиболее распространенные в настоящее время ОС Windows 8 и Windows 10, т.е. = 8.5 Гб.

Согласно формуле, объем памяти, необходимый для хранения программ определяется объемом памяти, который занимает запущенное приложение в памяти, а также объём необходимый для хранений файла справки. Для хранения системы необходимо 25 Мб. Для хранения БД необходимо 1024 Мб.

Таким образом, получаем, что = 8500 + 25 + 0 + 1024 + 0 = 9549 Мб.

* + 1. Рекомендованные характеристики технических средств

Для работы серверной части разработанной системы детектирования признаков акустического излучения ТС на аудиозаписях необходимы следующие технические средства:

* IBM-совместимый компьютер;
* не менее 875 Мбайт оперативной памяти;
* не менее 9549 Мбайт на жестком диске;
* поддерживаемая Windows 8 графическая карта;
* поддерживаемая Windows 8 мышь и клавиатура;
* операционная система Windows 8 и выше.

Для работы клиентской части разработанной системы необходимо одно из следующих технических средств:

* мобильное устройство под управлением операционной системы iOS не ниже 12 версии;
* мобильное устройство под управлением операционной системы Android не ниже 6 версии.
  1. Разработка алгоритмов функционирования системы

На рисунке 2.13 представлена схема алгоритма, соответствующего предложенному в 2.1 методу детектирования признаков (паттернов). Алгоритм учитывает особенности архитектуры системы, а именно использование мобильных устройств для сбора информации.

* 1. Разработка и описание интерфейса пользователя

Пользовательский интерфейс представляет собой совокупность правил и средств, которая регламентирует и обеспечивает взаимодействие пользователя с системой [62].

Перед началом работы пользователю нужно будет ввести логин и пароль в форме аутентификации. Экран формы входа в систему представлен на рисунке 2.14.

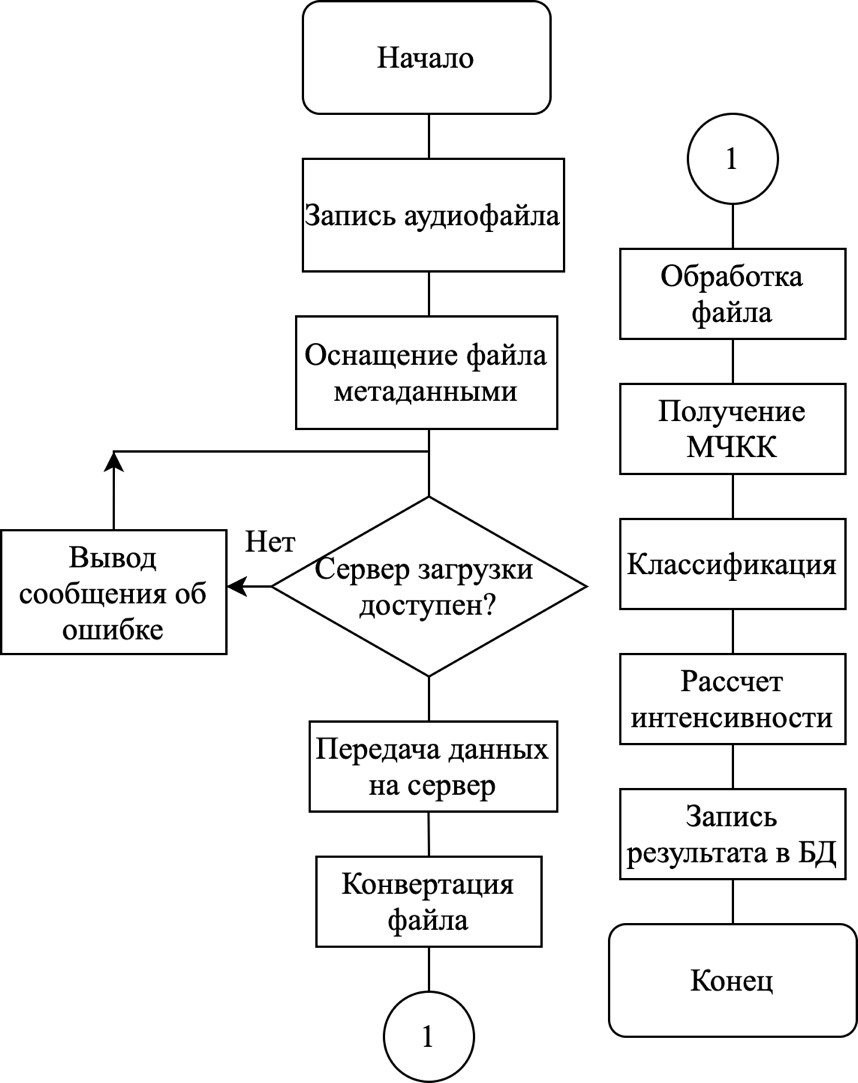


Рисунок 2.13 – Алгоритм классификации ТС на аудиозаписи

Если у пользователя нет учетной записи, то он может её создать, перейдя в форму создания учетной записи. Для создания учетной записи пользователю необходимо указать свое имя, логин и пароль. Экран создания учетной записи представлен на рисунке 2.15.

Первая форма после аутентификации – это карта, на которой расположены пункты, где был произведена запись и подсчитано количество ТС. Данная форма представлена на рисунке 2.16.

Чтобы сделать новую запись, необходимо нажать на кнопку «+», таким образом, пользователь перейдет в форму записи аудиофайла. Для того, чтобы отфильтровать точки замеров по каким-либо признаком, необходимо нажать на кнопку «Фильтры» и установить необходимую конфигурацию для последующего отображения.

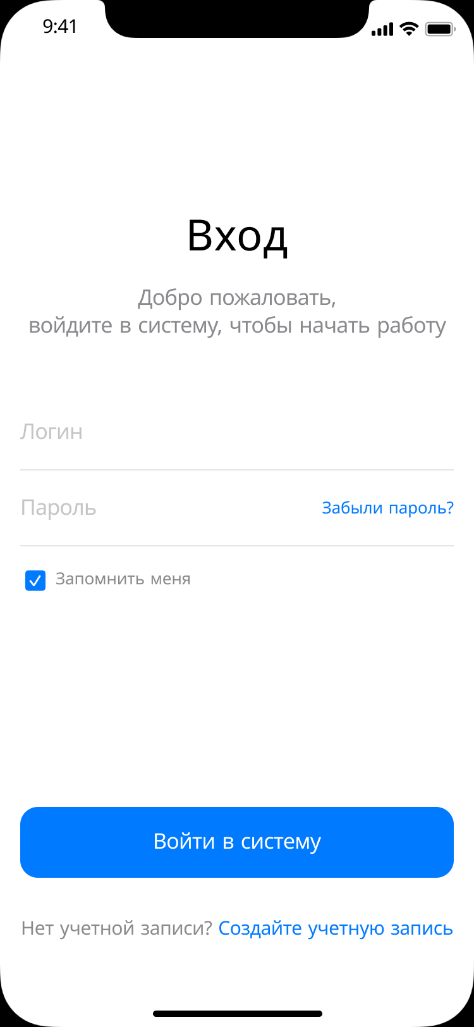


Рисунок 2.14 – Вход в систему

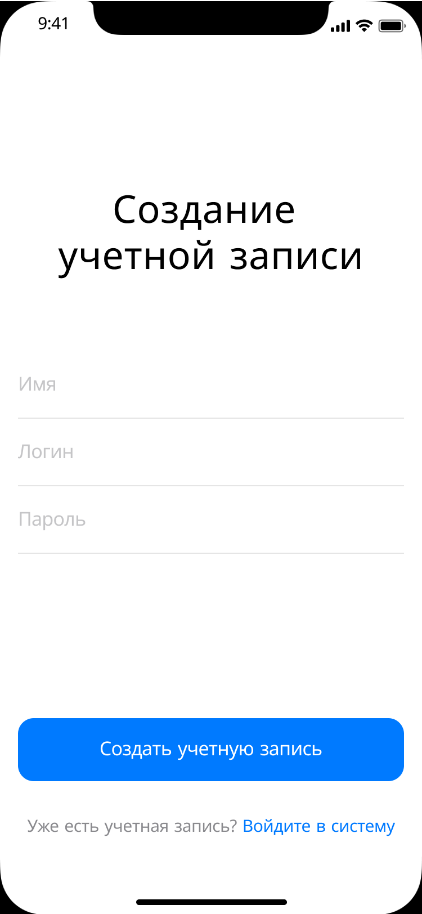


Рисунок 2.15 – Создание учетной записи

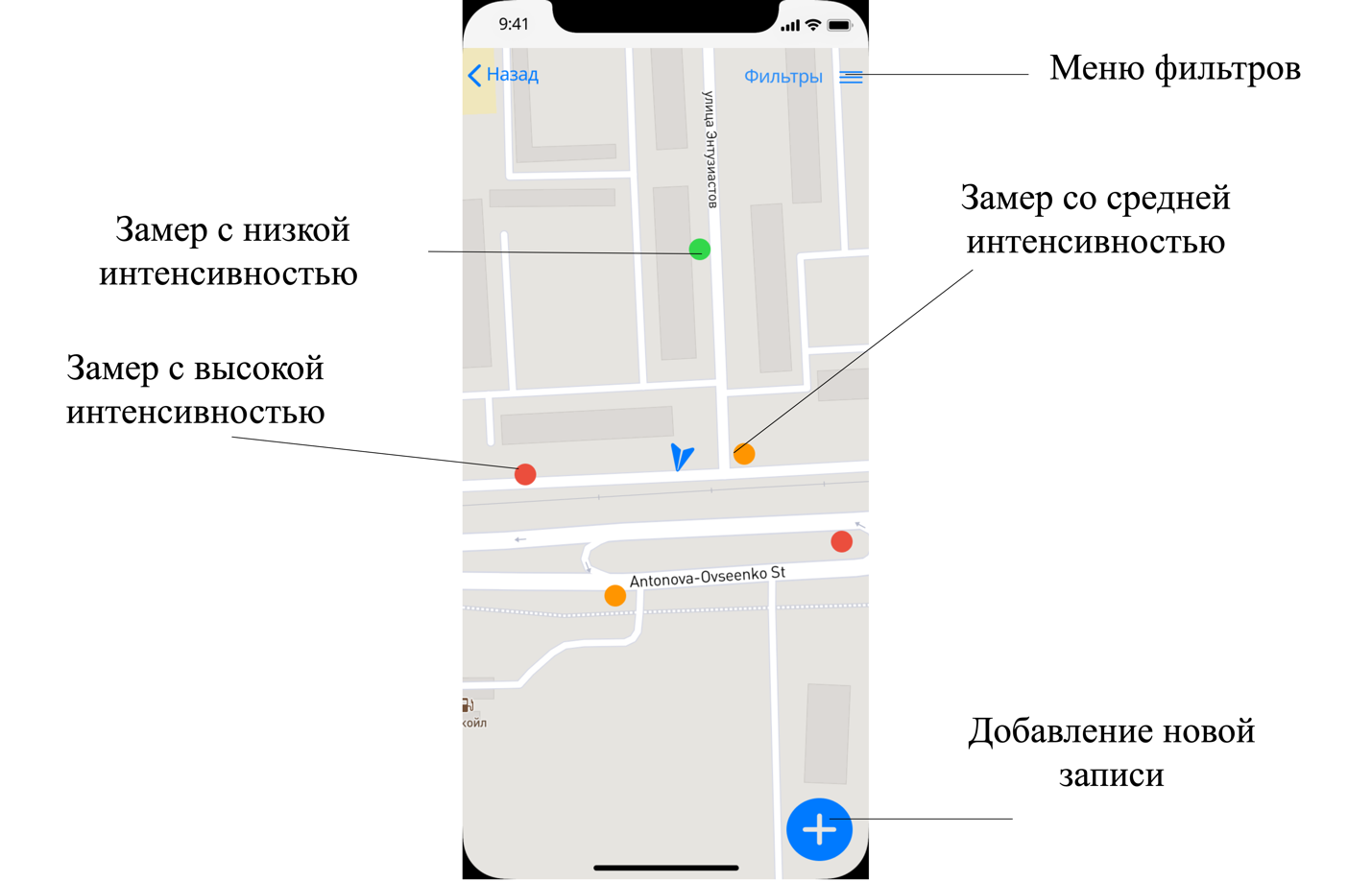


Рисунок 2.16 – Просмотр карты со списком замеров

На форме записи пользователю представлена возможность сделать запись. После того, как аудиозапись сделана, пользователю будет представлена спектрограмма, при нажатии на которую можно будет прослушать сделанную аудиозапись. При нажатии на кнопку «Отправить запись» происходит отправка сделанного аудиофайла на сервер. Также на сервер отправляются данные о местоположении, где была совершена запись, а также логин пользователя.

На рисунке 2.17 представлена форма, на которой можно сделать аудиозапись ТП, а также отправить данную запись на сервер.

При выборе замера на карте или на обзорном списке замеров, пользователю будет представлена форма обзора записи, пример которой представлен на рисунке 2.18. На данной форме содержится информация о дате, когда была совершена запись, точное время начала и конца записи, спектрограмма аудиосигнала и перечень ТС, который был определен. Также имеется возможность экспорта данной записи в PDF формат при выборе данной функции из контекстного меню.

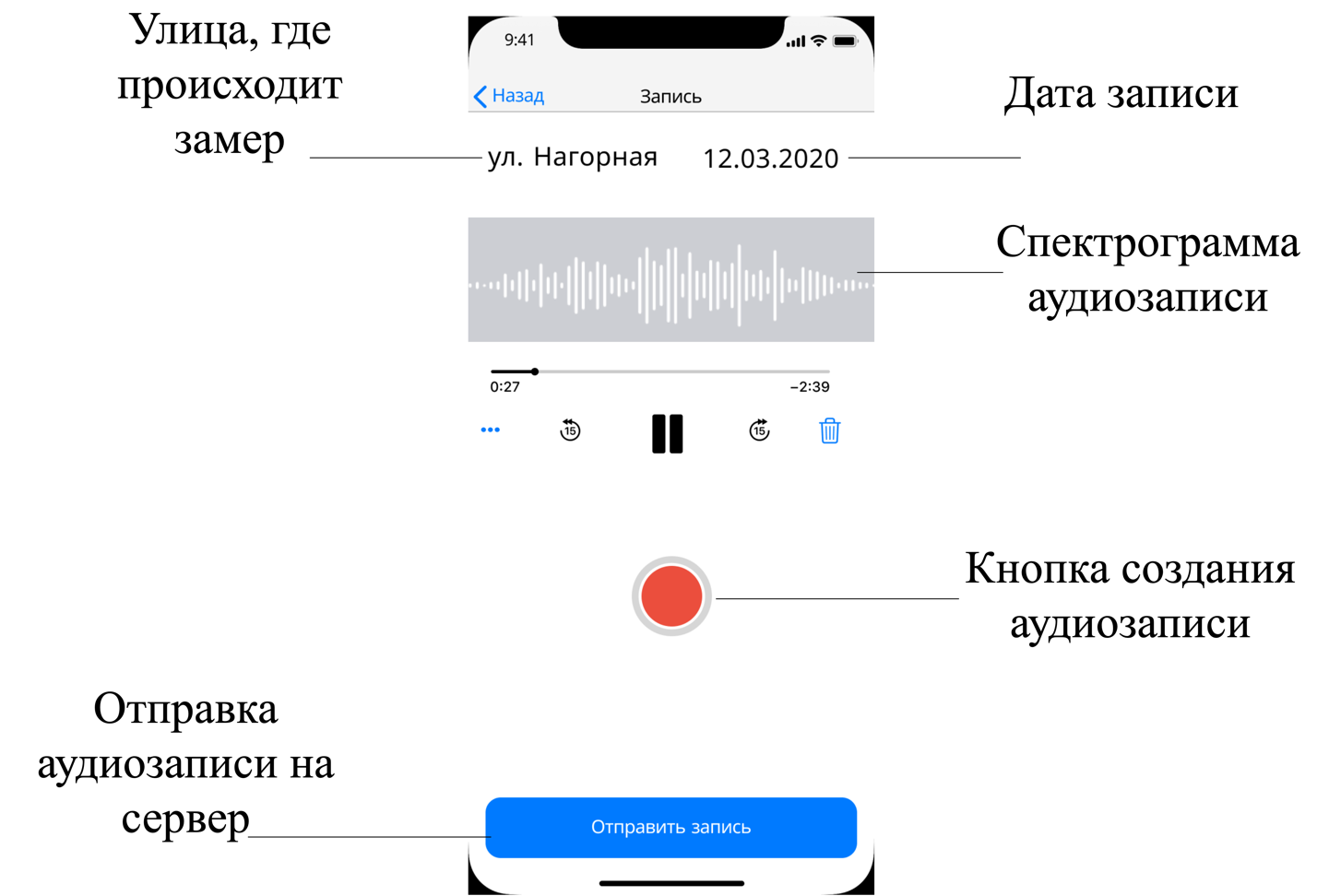


Рисунок 2.17 – Запись аудиофайла



Рисунок 2.18 – Обзор записи

На рисунке 2.19 представлен список замеров для более удобной навигации. Возможна фильтрация, для ограничения количества списков замеров.

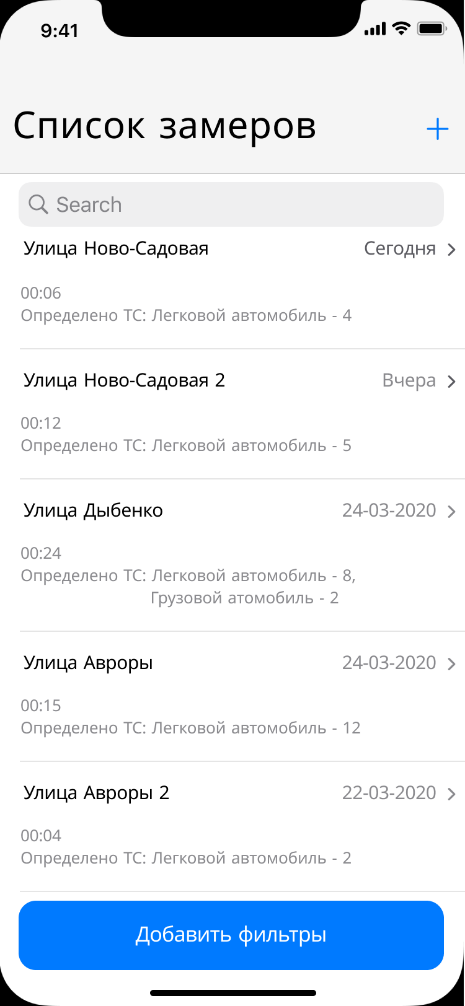


Рисунок 2.19 – Список замеров

Основными фильтрами являются: дата записи, время, в которое это запись была совершена, удаленность замера от текущей геолокации пользователя. Пример данной формы представлен на рисунке 2.20.

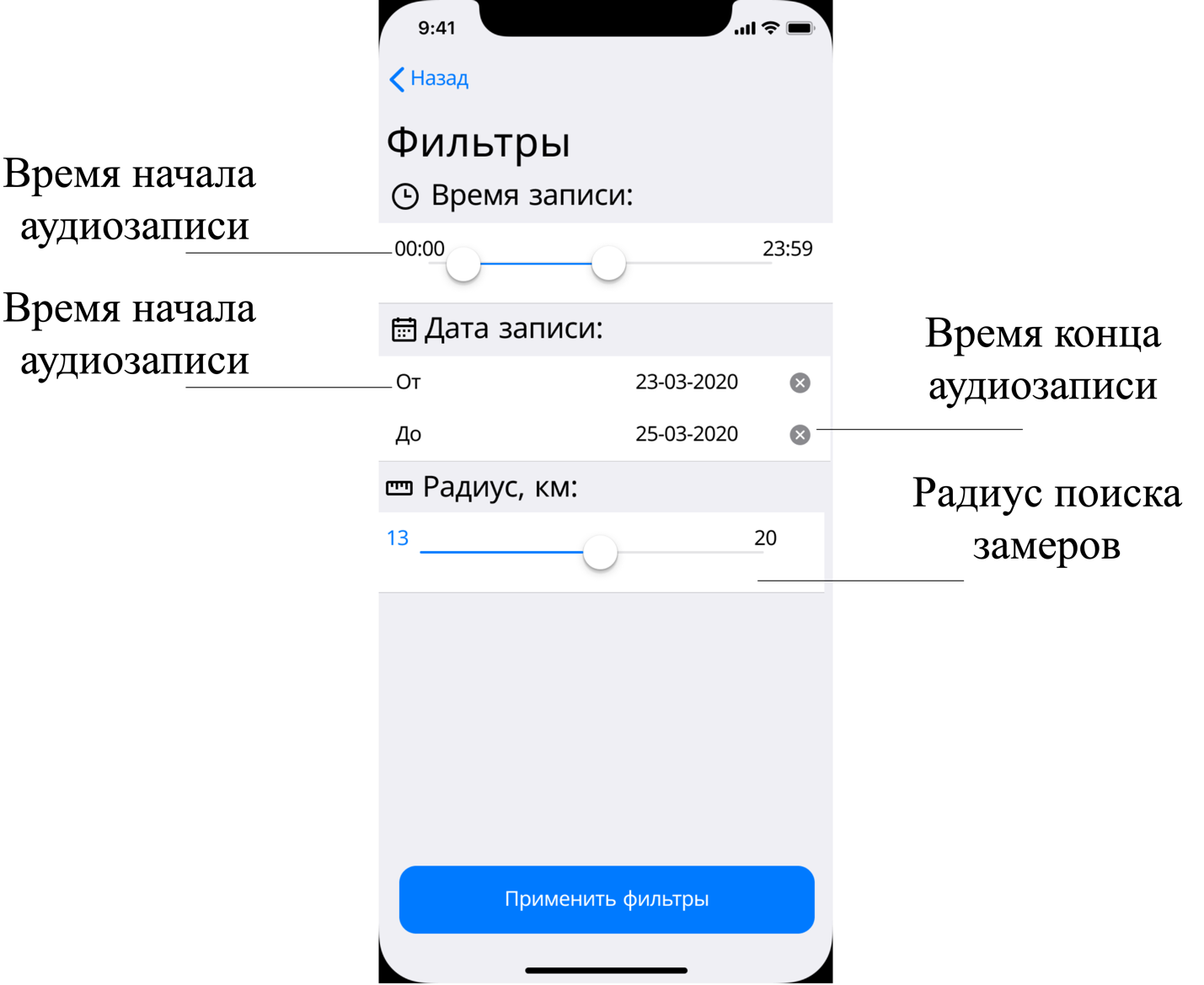


Рисунок 2.20 – Форма добавления фильтров

1. Экспериментальные исследования автоматизированной системы
   1. Исследование и визуализация данных
      1. Набор данных

Для разработки данной системы используется набор данных Urbansound8K [63]. Набор данных UrbanSound8K состоит из 8732 коротких (менее 4х секунд) фрагментов звуков города, которые разделены в 10 классов, которые состоят из:

* звук сигнала автомобиля;
* звук играющих детей;
* собачий лай;
* звук сверления;
* звук холостого хода двигателя;
* звук выстрела;
* звук пневматического бурильного молотка;
* звук сирены;
* звуки проезжающей машины;
* уличные шумы/ уличная музыка.

Обучающая и тестируемая выборки состоят из .wav файлов и описывающими их метаданными, хранящиеся в .csv таблице. Распределение классов представлено в таблице 3.1.

Как можно видеть, метки классов не сбалансированы. Хотя 7 из 10 классов имеют ровно 1000 образцов, звук сирены имеет примерно такое же значение (929), оставшиеся две метки (звук сигнала автомобиля и звук выстрела) имеют 43% и 37% относительно количества других образцов соответственно. Звуковые волны оцифровываются путем их дискретизации с дискретными интервалами, более известными как частота дискретизации (как правило, 44,1 кГц для звука с качеством для CD диска означает, что выборки берутся 44 100 раз в секунду).

Таблица 3.1 – Распределение классов звуков в наборе данных

|  |  |
| --- | --- |
| Имя класса | Количество записей в наборе |
| Звук играющих детей | 1000 |
| Собачий лай | 1000 |
| Уличные шумы/ уличная музыка | 1000 |
| Звук отбойного молотка | 1000 |
| Звук сверления | 1000 |
| Звук холостого хода двигателя | 1000 |
| Звук проезжающего автомобиля | 1000 |
| Звук сирены | 929 |
| Звук сигнала автомобиля | 429 |
| Звук выстрела | 374 |

Каждая выборка представляет собой амплитуду волны в конкретном интервале времени, где глубина в битах определяет, насколько детализированная выборка будет также известна как динамический диапазон сигнала (обычно 16 бит, что означает, что выборка может варьироваться от 65 536 значений амплитуды).

Пример дискретизации 16 битного сигнала представлен на рисунке 3.1.

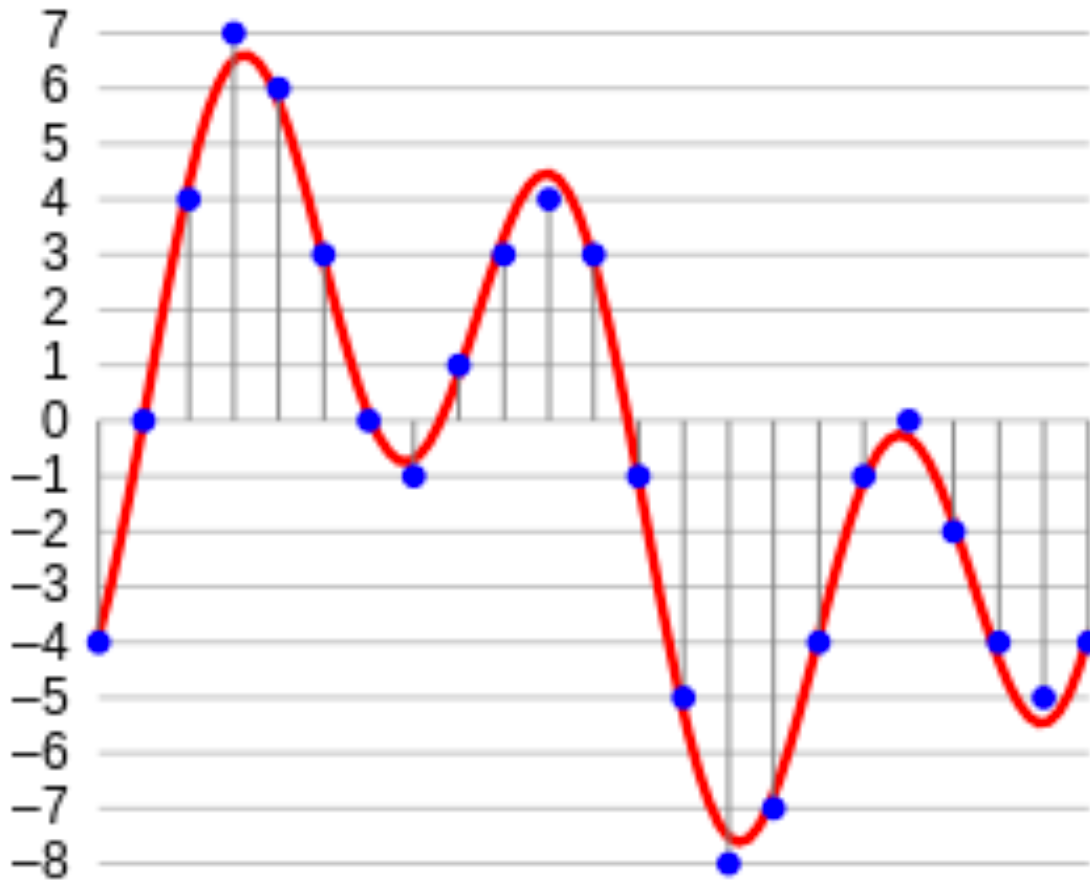


Рисунок 3.1 – Пример дискретизации сигнала

Поэтому данные, которые мы будем анализировать для каждого звукового фрагмента, по сути, представляют собой одномерный массив или вектор значений амплитуды.

* + 1. Преобразование данных

Сперва, происходит буферизация с перекрытием в исходном файле. Далее следует предобработка сигнала. Чтобы преобразовать данные в представления спектрограммы, была использована библиотека LibROSA, который представляет собой пакет с открытым исходным кодом, реализованный на Python. Данная библиотека предоставляет возможности для анализа звуковых и музыкальных данных, а также для визуализации спектрограммы.

На рисунке 3.2 представлен пример формы звукового сигнала, проезжающего мимо одного легкового автомобиля, а на рисунке 3.3 представлена спектрограмма выстрела из пистолета.

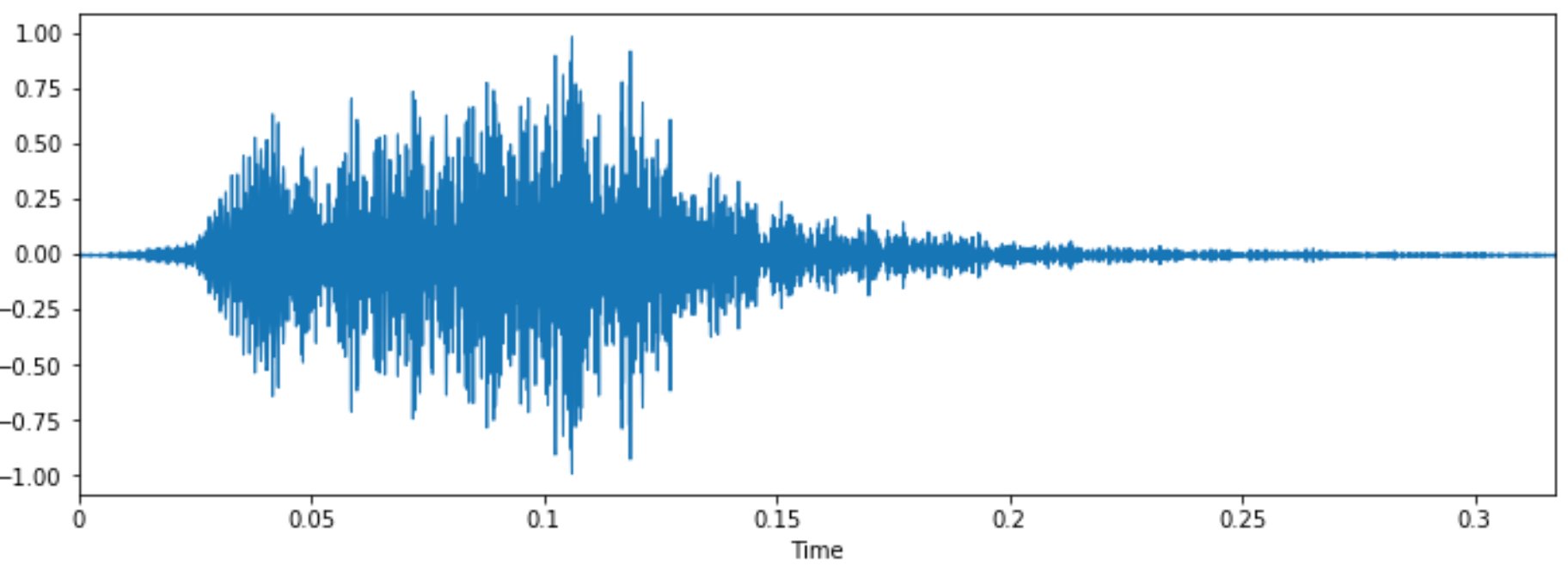


Рисунок 3.2 – Спектрограмма сигнала проезжающего легкового автомобиля

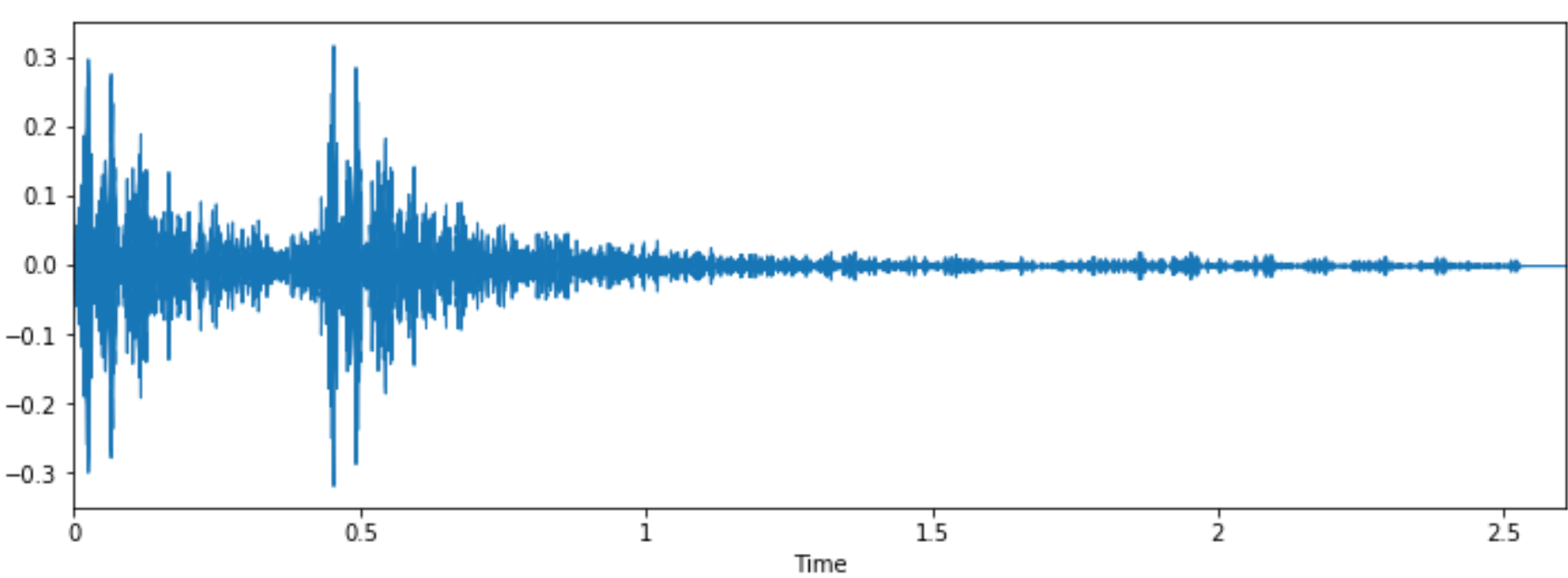


Рисунок 3.3 – Спектрограмма сигнала при выстреле из пистолета

Из визуального анализа видно, что тяжело идентифицировать разницу между некоторыми классами. Примером являются схожие формы сигналов повторяющихся звуков сверления, холостого хода двигателя и отбойного молотка. Аналогично, пики в образце проезжающего автомобиля похож по форме на образец выстрела из пистолета (хотя образцы отличаются тем, что для двух выстрелов имеется два пика по сравнению с одним пиком для проезжающего автомобиля). Также у формы автомобиля есть схожести с уличной музыкой, и звуком играющих детей.

Человеческое ухо может естественным образом обнаружить разницу между гармониками, будет интересно посмотреть, насколько хорошо модель глубокого обучения сможет извлечь необходимые функции для различения этих классов.

Тем не менее, легко отличить от формы волны, разницу между определенными классами, такими как лай собаки и отбойный молоток.

Также стоит отметить, что большинство экземпляров выборки, имеют два аудиоканала (стереозвук), хотя у некоторых всего лишь один аудиоканал. Самый простой способ избежать данной проблемы это объединить два канала в один, путем усреднения двух каналов.

* 1. Проведение исследования
     1. Методика исследования

Методика исследования заключается в подборе наиболее оптимальной и точной архитектуры нейронной сети. Для сравнения выбраны две архитектуры: МЛП и СНС. МЛП был выбран потому, что он является классической архитектурой и является первым выбором в качестве архитектуры сети, когда рассматривается новая область задач для решения которых может примениться нейронная сеть.

Архитектура СНС предлагается в качестве альтернативы МЛП, потому что архитектура СНС широко применяется в задача классификации образов, а данная область смежная с задачей, которая рассматривается в данной работе.

Для сравнения эффективностей данных архитектур, было произведено обучение сетей с данными архитектурами на одинаковом наборе данных. После обучения был произведен замер точности классификации и выбиралась сеть с наибольшей точностью.

Большую точность классификации показала архитектура СНС.

* + 1. Инструменты получения признаков

Для решения проблемы детектирования паттернов а аудиосигналах, было предложено применить методы глубоко обучения, которые оказались крайне успешными в решении задач классификации изображений.

В качестве инструмента извлечения признаков используется МЧКК. Сперва извлекаем МЧКК из экземпляров для каждого кадра с размером окна в несколько миллисекунд. МЧКК суммируют распределение частот по размеру окна, поэтому можно анализировать как частотные, так и временные характеристики звука. Подобное представление аудио позволит нам определить признаки для классификации. Далее следует постобработка полученных признаков.

На рисунке 3.4 показан график вычленения МЧКК для аудиофайла длинной в 60 секунд.

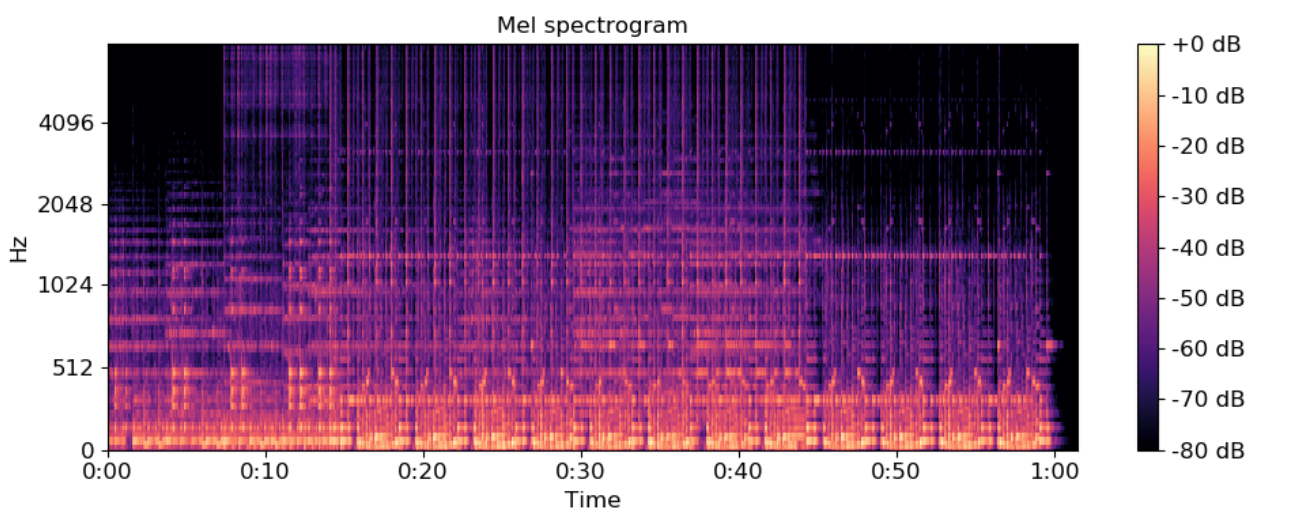


Рисунок 3.4 – Спектрограмма проезжающего потока автомобилей

* + 1. Обучение многослойного персептрона

Для решения задачи классификации мы будет использовать нейронную сеть. Для начала был рассмотрен простая структура нейронной сети – многослойный персептрон (МЛП). Построение МЛП выполнено с помощью Keras и TensorFlow.

Модель сети является последовательной, что позволяет нам размещать слои последовательно. Архитектура разработанного многослойного персептрона состоит из трех слоев: входной, скрытый и выходной слои.

Первый слой является входным. Каждый образец содержит 40 МЧКК, поэтому слой имеет форму 1х40. Первые два слоя имеют 256 нейронов, а функция активации – ReLU, потому что это является оптимальным выбором для нейронных сетей с подобной архитектурой. Уровень исключения равен 50%, так как это позволит нам регуляризовать нейронную сеть во время обучения, что позволяет получить сеть с более качественным предсказаниями.

Выходной слой будет иметь 10 нейронов, которые соотносятся с количеством классов объектов наборе данных. Функцией активации для этого класса выбрана softmax. Softmax делает выходную сумму близкой к 1, поэтому выходные значения могут быть интерпретированы как вероятности. Затем модель сделает свой прогноз на основе того, какой вариант имеет наибольшую вероятность.

Для тестирования модели существуют различные алгоритмы. Эффективность алгоритмов тестирования представлена в таблице 3.2 [64].

Таблица 3.2 – Точность классификации алгоритмов тестирования

|  |  |
| --- | --- |
| Имя алгоритма | Точность классификации, % |
| RBF SVM | 68 |
| Random Forest 500 | 66 |
| IBk5 | 55 |
| J48 | 48 |
| ZeroR | 10 |

Результаты тестирования точности распознавания модели на тестовой и обучающих выборках получись равными 87,6% и 92,5% соответственно. Значения получились довольно высокие, а также небольшая разница между ними (около 5%). Это нам говорит о том, что при обучении модель не была подвержена переобучению. График зависимости точности классификации от количества эпох представлен на рисунке 3.5.

Рисунок 3.5 – График зависимости точности классификации  
 от количества эпох обучения МЛП

* + 1. Обучение сверточной нейронной сети

Обучение МЛП показало достаточно хороший результат. Однако стоит выяснить, может ли другая архитектура сети получить еще более высокую точность. СНC показывает хороший результат в задачах классификации изображений, поэтому данная архитектура была выбрана для проверки предположения.

Так как СНC требует, чтобы количество входов было равно, мы обнуляем векторы, чтобы они все стали одинакового размера.

Модель сети также остается последовательной, с 4 сверточными слоями и плотным выходным слоем. Количество фильтров для сверточных определяется как 16, 32, 64 и 128. Размер ядра равен матрице 2х2, потому что размер окна в данном случае равен 2.

Первый слой получает форму 40, 174 на 1, где 40 – количество МЧКК, 174 – количество кадров с учетом заполнения и 1 – количество каналов. Функция активации для сверточных слоев будет, как и в прошлой модели – ReLU, а уровень исключения – 20%.

Каждый сверточный слой имеет субдискретизирующий слой размерностью 2 на 2. Все субдискретизирующие слои соединены с одни с усредняющим слоем, который подает усредненные данные на выходной слой. Субдискретизирующие слои позволяют снизить размерность модели (путем уменьшения количества параметров и последовательных вычислений), что ведет к уменьшению времени обучения и снижает переобучаемость.

Выходной слой идентичен выходному слою персептрона.

На рисунке 3.6 представлен график зависимости точности классификации от количества эпох обучения.

Рисунок 3.6 – График зависимости точности классификации  
 от количества эпох обучения СНС

Результаты тестирования точности распознавания модели на тестовой и обучающих выборках получись равными 98,2% и 92,0% соответственно. Точность обучения повысилось на 6% для обучающей и около 4% для тестирующей. Хотя разница между ними увеличилась (до 6%), данное значение не столь велико, что является индикатором того, при обучении модель не была подвержена переобучению.

* 1. Результаты исследования

На этапе разработки модели данные валидации использовались для оценки модели. СНС была выбрана в качестве архитектуры потому, что она показала более высокую точность классификации. Точность классификации у СНС составила 92,0% против 87,6% у МЛП.

Результирующая модель достигла точности классификации 92,0%, что превышает ожидания по точности классификации тестирующей модели RBF SVM.

Валидация модели проводилась c использованием аудиофайлов, которые были взяты из свободного аудио банка, а также с данными, собранными самостоятельно.

В качестве контрольной проверки работоспособности системы рассмотрены аудиофайлы, которые были собраны.

* 1. Описание контрольного примера работы системы
     1. Назначение

Контрольный пример предназначен для проверки правильности работы следующих функций автоматизированной системы детектирования акустического излучения ТС:

* выгрузка аудиофайла на сервер приложения;
* обработка и классификация переданного файла;
* классификация;
* сохранение результатов классификации в базу данных;
* отображение результатов классификации на клиентском устройстве.
  + 1. Исходные данные

Для выполнения контрольного примера 12.03.2020 была сделана аудиозапись ТП на пересечении улиц Нагорная и Майская. Запись сделана с помощью разработанного приложения, которое установили на мобильный телефон iPhone XS под управление iOS 13.

На данной аудиозаписи длиной 160 секунд должно быть зафиксировано 23 автомобиля. На рисунке 3.5 представлена аудиозапись перед отправлением на сервер.



Рисунок 3.5 – Спектрограмма проезжающего потока автомобилей

* + 1. Результаты расчета

В ходе обработки файла, отправленного на сервер, было обнаружено 21 ТС. Пример отображения результата после обработки показан на рисунке 3.6.

Система верно определила большую часть ТС, хотя не смогла 2 ТС. Это связано с тем, что они проехали в один и тот же момент времени с другим ТС, что вызывало перекрытие их акустического излучения. Однако этот факт не повлиял на общую точность классификации, которая составила 92.0% и 87,6%.

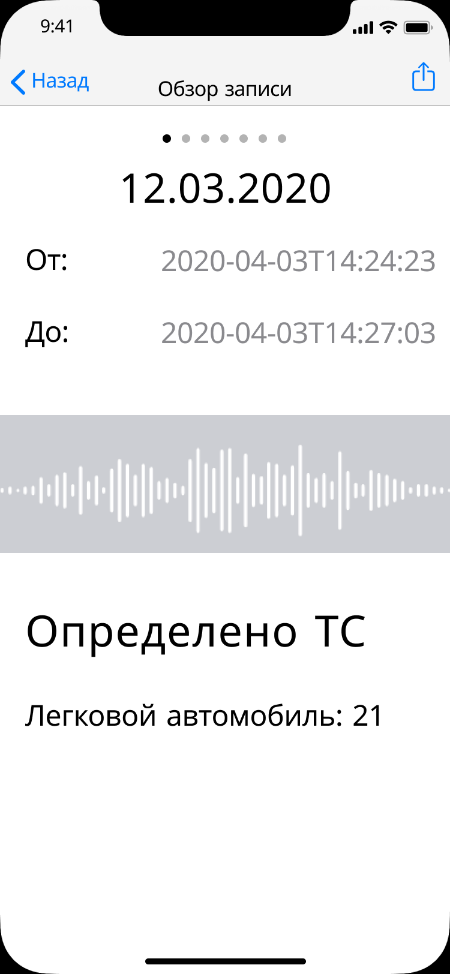


Рисунок 3.6 – Спектрограмма проезжающего потока автомобилей

В таблице 3.3 представлена точность классификации контрольного набора данных, который был собран вручную и взят с открытых звуковых банков.

Таблица 3.3 – Точность классификации алгоритмов тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование аудиозаписи | Количество подсчитанных вручную ТС | Количество определенных системой ТС | Точность классификации, % |
| Запись 1 | 68 | 65 | 92,5 |
| Запись 2 | 43 | 43 | 100,0 |
| Запись 3 | 10 | 10 | 100,0 |
| Запись 4 | 14 | 13 | 92,9 |
| Запись 5 | 33 | 30 | 90,9 |
| Запись 6 | 68 | 65 | 92,5 |
| Запись 7 | 22 | 18 | 81,8 |
| Запись 8 | 3 | 103 | 100,0 |
| Запись 9 | 55 | 50 | 90,9 |
| Запись 10 | 46 | 53 | 86,8 |
| Личная запись 1 | 15 | 13 | 86,7 |
| Личная запись 2 | 7 | 7 | 100,0 |
| Личная запись 3 | 33 | 30 | 90,9 |
| Личная запись 4 | 46 | 35 | 76,1 |
| Личная запись 5 | 18 | 9 | 50,0 |
| Личная запись 6 | 13 | 7 | 53,8 |
| Личная запись 7 | 18 | 15 | 83,3 |
| Личная запись 8 | 24 | 20 | 83,3 |
| Личная запись 9 | 68 | 55 | 80,9 |
| Личная запись 10 | 34 | 17 | 50,0 |

ЗАКЛЮЧЕНИЕ

В соответствии с заданием на выпускную квалификационную работу магистра проведен анализ предметной области, рассмотрены задачи детектирования признаков и особенности акустического излучения, проанализированы нейросетевые технологии и алгоритмы дискретного преобразования сигналов, применяемые в решении аналогичных задач. Выполнен аналитический обзор существующих систем-аналогов. Разработан метод классификации объектов по их акустическому излучению. Построены диаграммы по методологии UML, создана модель данных, разработаны алгоритмы функционирования и описана архитектура системы. Разработана автоматизированная система детектирования признаков акустического излучения транспортных средств на аудиозаписях с использованием сверточных нейронных сетей. Проведены исследования эффективности разработанной системы для решения поставленной задачи. Программное обеспечение системы разработано на языках Python и JavaScript в среде Visual Studio Code.

По теме выпускной квалификационной работы опубликовано 8 научных работ [65-72]. Результаты работы докладывались на различных конференциях международного, областного и регионального уровней, отмечены дипломами о лучших докладах.

Таким образом, основные результаты работы:

* разработан новый метод детектирования признаков (паттернов) акустического излучения транспортных средств в аудиофайлах путем получения мел-частотных кепстральных коэффициентов в качестве признаков, классифицируемых с помощью СНС;
* разработаны алгоритмы функционирования, а также разработано программное обеспечение автоматизированной системы, реализующие предложенный метод детектирования признаков. Проведенное с использованием разработанной системы исследование на наборе данных UrbanSound8k, состоящем из 8732 помеченных аудиозаписей, показало результирующую точность классификации с использованием СНС в 92%. Так, в сравнении с МЛП, СНС обладает более высокой точностью классификации: 92,0% у СНС против 87,6% у МЛП. Кроме этого, проведено исследование на авторском наборе данных, включающем 200 аудиофайлов, записанных на улично-дорожной сети г.о. Самара с использованием разработанного мобильного приложения, и дополненного аудиофайлами, взятыми из открытых банков данных. Результирующая точность на авторском наборе данных с применением СНС составила 84,5%. Некоторое сокращение точности на авторском наборе относительно эталонного UrbanSound8k связано с тем, что в авторских аудиозаписях присутствует наслоение звукового излучения от нескольких ТС в один и тот же момент времени.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Головнин, О.К. Управляемый данными анализ транспортных потоков в различных дорожных условиях [Текст] / О.К. Головнин // V Междунар. конф. и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2019) : сборник трудов (Самара, 21–24 мая 2019 г.). – Самара : Новая техника, 2019. – Т. 4. Науки о данных. – С. 533-542.
2. Гасников, А.В. Введение в математическое моделирование транспортных потоков [Текст] / А.В. Гасников, С.Л. Кленов, Е.А. Нурминский, Я.А. Холодов [и др.]; под ред. А.В. Гасникова. – М.: МФТИ, 2010. – 360 с.
3. Кузмин, С.А. Алгоритмы обработки видеоинформации в системе мониторинга транспортных потоков [Текст] / С.А. Кузмин. – Электротехнические и информационные комплексы и системы. – № 3. – 2015. – 138 с. – C. 90-96.
4. DeepWiTraffic: Low cost WiFi-based traffic monitoring system using deep learning [Электронный ресурс] // URL: https://arxiv.org/abs/1812.08208 (дата обращения: 23.05.2020).
5. Lee, H. Using LIDAR to validate the performance of vehicle classification stations [Текст] / H. Lee, B. Coifman. – Journal of Intelligent Transportation Systems. – Vol. 19. – 2015. – P. 355-369.
6. Rajab, S.A. Vehicle classification and accurate speed calculation using multi-element piezoelectric sensor [Текст] / S.A. Rajab, A. Mayeli, H.H. Refai. – IEEE Intelligent Vehicles Symposium Proceedings. IEEE, 2014. – P. 894–899.
7. Xu, C. Vehicle classification using an imbalanced dataset based on a single magnetic sensor [Текст] / C. Xu, Y. Wang, X. Bao, F. Li. – Sensors, Vol. 18, №. 6, 2018. – 1690 p.
8. Stocker, M. Situational knowledge representation for traffic observed by a pavement vibration sensor network [Текст] / M. Stocker, M. Ronkko, and M. Kolehmainenm. – IEEE Transactions on Intelligent Transportation Systems, Vol. 15, № 4, 2014. – P. 1441–1450.
9. Meta, S. Vehicle-classification algorithm based on component analysis for single-loop inductive detector [Текст] / S. Meta, M. G. Cinsdikici. – IEEE Trans- actions on Vehicular Technology, Vol. 59, №. 6, 2010. – P. 2795–2805.
10. Yang, B. Vehicle detection and classification for low-speed congested traffic with anisotropic magnetoresistive sensor [Текст] / B. Yang, Y. Lei. – IEEE Sensors Journal, Vol. 15, № 2, 2015. – P. 1132–1138.
11. Ma, W. wireless accelerometer-based automatic vehicle classification prototype system [Текст] / W. Ma, D. Xing, A. McKee, R. Bajwa, C. Flores, B. Fuller, P. Varaiya A. – IEEE Transactions on Intelligent Transportation Systems, Vol. 15, № 1, 2014. – P. 104–111.
12. Georg, J. Vehicle detection and classification from acoustic signal using ANN and KNN in Control Communication and Computing (ICCC) [Текст] / J. George, L. Mary, K. Riyas. – International Conference on. IEEE, 2013. – P. 436–439.
13. Lee, H. Side-fire LIDAR-based vehicle classification [Текст] / H. Lee, B. Coifman. – Transportation Research Record: Journal of the Transportation Research Board, № 2308, 2012. – P. 173–183
14. Odat, E. Vehicle classification and speed estimation using combined passive infrared/ultrasonic sensors [Текст] / E. Odat, J. S. Shamma, C. Claudel. – IEEE Transactions on Intelligent Transportation Systems, 2017. 1241 p.
15. Won, M. WiTraffic: Low-cost and non-intrusive traffic monitoring system using WiFi [Текст] / M. Won, S. Zhang, S. H. Son // 26th International Conference on Computer Communication and Networks (ICCCN). – IEEE, 2017. – P. 1–9.
16. Tang, T. Arbitrary-oriented vehicle detection in aerial imagery with single convolutional neural networks [Текст] / T. Tang, S. Zhou, Z. Deng, L. Lei, H. Zou. – Remote Sensing, Vol. 9, № 11, 2017. – 1170 p.
17. Chen, Z. Vehicle detection, tracking and classification in urban traffic [Текст] / Z. Chen, T. Ellis, S. A. Velastin // 15th International IEEE Conference on Intelligent Transportation Systems, IEEE, 2012. – P. 951–956.
18. Chidlovskii, B. Vehicle type classification from laser scans with global alignment kernels [Текст] / B. Chidlovskii, G. Csurka, J. Rodriguez-Serrano // 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). – IEEE, 2014. – P. 2840–2845.
19. What is a feature extraction? [Электронный ресурс] // URL: https://deepai.org/machine-learning-glossary-and-terms/feature-extraction (дата обращения: 23.05.2020).
20. Alpaydin, E. Introduction to Machine Learning [Текст] / E. Alpaydin. – London: The MIT Press, 2010. – 110 p.
21. An introduction to feature selection [Электронный ресурс]. – https://machinelearningmastery.com/an-introduction-to-feature-selection/
22. World of Physics. Wolfram [Электронный ресурс] // URL: http://scienceworld.wolfram.com/physics/Radiation.html (дата обращения: 23.05.2020).
23. Прохоров, М.А. Физическая Энциклопедия. Звук [Текст] / М.А. Прохоров, А.М. Бонч-Бруевич, А. М. Балдин. – Физическая энциклопедия: [в 5 т.] / Гл. ред. А. М. Прохоров. — М.: Советская энциклопедия (т. 1 – 2); Большая Российская энциклопедия (т. 3—5), 1988—1999. – 1463 c.
24. Васильев, А.В. Исследование уровня акустического излучения системы «шины автотранспортного средства – дорожное покрытие» [Текст] / А. В. Васильев, Е.А. Комлик // Noise Theory And Practise, 2015. – С. 38-34.
25. Васильев, А.В. Акустическая экология города: учеб. пособие для студентов вузов [Текст] / А.В. Васильев. – Федеральное агентство по образованию, Тольяттинский гос. ун-т Тольятти, 2007. – 166 с.
26. Прохоров, С.А. Структурно-спектральный анализ случайных процессов [Текст] / С. А. Прохоров, В. В. Графкин. – Самарский национальный исследовательский университет им. акад. С.П. Королева, 2010. – 128 c.
27. Прохоров, С.А. Прикладной анализ случайных процессов [Текст] / С.А. Прохоров, В.В. Графкин, О.А. Дягтерева. – Cамарский государственный аэрокосмический университет им. акад. С.П. Королева, 2007. – 586 c.
28. Chen, Y.Y. Design and Implementation of Cloud Analytics-Assisted Smart Power Meters Considering Advanced Artificial Intelligence as Edge Analytics in Demand-Side Management for Smart Homes [Текст] / Y. Y. Chen, Y. H. Lin, C. C. Kung, M.H. Chung, I.H. Yen // Sensors. 19 (9), 2019. – 2047 p.
29. Davis, S. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences [Текст] / S. Davis, P. Mermelstein // IEEE Transactions on Acoustics, Speech, and Signal Processing. – 1980. – Vol. 28. – P. 357-366.
30. Миркес, Е.М. Нейрокомпьютер [Текст] / Е. М. Миркес. – Новосибирск: Наука, 1999. – 337 с.
31. Из истории искусственного интеллекта: история искусственного интеллекта до середины 80-х годов [Электронный ресурс] // URL: https://refdb.ru/look/1538888.html (дата обращения: 23.05.2020).
32. Hoppensteadt, F.C. Weakly connected neural networks [Текст] / F. C. Hoppensteadt, E. M. Izhikevich. – Springer, 1997. – 402 p.
33. Maan, A.K. Survey of Memristive Threshold Logic Circuits [Текст] / A.K. Maan, D. A. Jayadevi, A. P James // IEEE Transactions on Neural Networks and Learning Systems, 2016. – P. 1734–1746.
34. Солдатова, О.П. Основы нейроинформатики [Текст] / О.П. Солдатова. – Самара:СГАУ, 2006. – 131 с.
35. LeCun, Y. Backpropagation Applied to Handwritten Zip Code Recognition [Текст] / Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel // Neural Computation, Vol. 1(4), 1989. – P. 541-551.
36. DeepLearning 0.1. LISA Lab [Электронный ресурс] // URL: http://deeplearning.net/tutorial/lenet.html (дата обращения: 23.05.2020).
37. Matusugu, M. Subject independent facial expression recognition with robust face detection using a convolutional neural network [Текст] / M. Matusugu, M. Katsuhiko, M. Yusuke, K. Yuji // Neural Networks: journal. Vol. 16, № 5, 2003. – P. 555—559.
38. Федоренко, С.В. Модификация алгоритма Грецеля-Блейхута [Текст] / С.В. Федоренко. – Статья. - Журнал Приборостроение, 2013. – С 17–20.
39. Гаусс, К.Ф. Арифметические исследования [Текст] / К.Ф. Гаусс // Труды по теории чисел. – М.: Изд-во АН СССР. – 1959. – 703 с.
40. Котельников, В.А. Собрание трудов в пяти томах [Текст] / В.А. Котельников. – Т. 5, Часть 2. Основы радиотехники. – 486 с.
41. Min, X. HMM-based audio keyword generation [Текст] / X. Min, A. Kiyoharu, N. Yuichi, S. Shin'ichi // Advances in Multimedia Information Processing. – PCM 2004: 5th Pacific Rim Conference on Multimedia. Springer, 2004. – 544 p.
42. Stevens, S. S. Scale for the Measurement of the Psychological Magnitude Pitch [Текст] / S. S. Stevens, J. Volkmann, E. B. Newman. – 1937. – 188 p.
43. Dixon Ward, W. Musical Perception [Текст] / W. Dixon Ward. – Foundations of Modern Auditory Theory / Jerry V. Tobias. — Academic Press, 1970. – 412 p.
44. O'Shaughnessy, D. Speech communication: human and machine [Текст] / D. O'Shaughnessy. – Addison-Wesley, 1987. – 150 p.
45. Оппенгейм, А.В. Цифровая обработка сигналов [Текст] / А. В. Оппенгейм, Р. В. Шафер. – М.: Связь, 1979. – 361 c.
46. Рабинер, Л.Р. Цифровая обработка речевых сигналов [Текст] / Л.Р. Рабинер, Р.В. Шафер. – Рипол Классик, 1981.
47. Sahidullah, M. Design, analysis and experimental evaluation of block-based transformation in MFCC computation for speaker recognition [Текст] / M. Sahidullah, G. Saha // Speech Communication. – Vol. 54 (4), 2012. – P. 543–565.
48. ShotSpotter: The review of the system [Электронный ресурс] // URL: https://www.shotspotter.com/technology/ (дата обращения: 23.05.2020).
49. An Industrial-Strength Audio Search Algorithm [Электронный ресурс] // URL: https://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf (дата обращения: 23.05.2020).
50. Audio Analytics – enabling intelligent products through sound recognition [Электронный ресурс] // URL: https://www.audioanalytic.com/ (дата обращения: 23.05.2020).
51. Иващенко, А.В. Теоретические основы проектирования автоматизированных систем обработки информации и управления [Текст] / А.В. Иващенко, И.А. Лёзин, И.В. Лёзина. – Самара: СНЦ РАН, 2007. – 94 с.
52. Knuth, D.E. Mathematical writing [Электронный ресурс] // URL: http://jmlr.csail.mit.edu/reviewing-papers/knuth\_mathematical\_writing.pdf (дата обращения: 23.05.2020).
53. Visual Studio Code source code [Электронный ресурс] // URL: https://github.com/microsoft/vscode (дата обращения: 23.05.2020).
54. Microsoft’s new Code editor is built on Google’s Chromium [Электронный ресурс] // URL: https://arstechnica.com/information-technology/2015/04/microsofts-new-code-editor-is-built-on-googles-chromium/ (дата обращения: 23.05.2020).
55. What Is MongoDB? [Электронный ресурс] // URL: https://www.mongodb.com/what-is-mongodb (дата обращения: 23.05.2020).
56. Hassan, S. Microservices and their design trade-offs: A self-adaptive roadmap [Текст] / S. Hassan, R Bahsoon // IEEE International Conference on Services Computing (SCC), IEEE, 2016. – P. 813–818.
57. Lewis, J.M. Microservices: a definition of this new architectural term [Электронный ресурс] // URL: martinfowler.com/articles/microservices.html (дата обращения: 23.05.2020).
58. Fowler, M. Monolith first [Электронный ресурс] // URL: http://martinfowler.com/bli- ki/MonolithFirst.html (дата обращения: 23.05.2020).
59. Теория и практика UML. Диаграмма состояний [Электронный ресурс] // URL: http://www.it-gost.ru/articles/view\_articles/97 (дата обращения: 23.05.2020).
60. Дерябкин, В.П. Методические указания к лабораторному практикуму по UML [Текст] / В.П. Дерябкин, О.К. Либерзон. – Самара, 2008. – 120 c.
61. Буч, Г. Язык UML Руководство пользователя [Текст] / Г. Буч, Д. Рамбо, А. Джекобсон – 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2006. – 496 с.
62. Пользовательский интерфейс [Электронный ресурс] // URL: http://atworks.ru/polzovatelskij-interfejs-chto-eto.html (дата обращения: 23.05.2020).
63. Dataset UrbanSound8k [Электронный ресурс] // URL: https://urbansounddataset.weebly.com/urbansound8k.html (дата обращения: 23.05.2020).
64. A dataset and taxonomy for urban sound research [Электронный ресурс] // URL: www.justinsalamon.com/uploads/4/3/9/4/4394963/salamon\_urbansound\_ acmmm14.pdf (дата обращения: 23.05.2020).
65. Головнин, О.К. Анализ характеристик транспортных потоков в интеллектуальной транспортной системе на основе аудиофайлов [Текст] / О.К. Головнин, А.С. Привалов // Актуальные проблемы автотранспортного комплекса: межвузовский сборник научных статей (с международным участием). – Самара: СамГТУ, 2019. – С. 126-130.
66. Привалов, А.С. Детектирование транспортных средств на аудиозаписи с использованием сверточных нейронных сетей [Текст] / А.С. Привалов // Новые информационные технологии в научных исследованиях : материалы XXIV Всероссийской научно-технической конференции студентов, молодых ученых и специалистов (Рязань, 13–15 ноября 2019 г.). – Рязань : Book Jet, 2019. – С. 221.
67. Golovnin, O.K. A Web-Oriented Approach for Urban Road Traffic Simulation [Электронный ресурс] / O.K. Golovnin, K.V. Pupynin, A.S. Privalov // 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon) (1-4 Oct. 2019, Vladivostok, Russia), IEEE, 2019. – URL : https://ieeexplore.ieee.org/document/8934302 (дата обращения: 23.05.2020).
68. Golovnin, O.K. Vehicle Detection in Audio Recordings by Machine Learning [Электронный ресурс] / O.K. Golovnin, A.S. Privalov, K.V. Pupynin, // 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon) (1-4 Oct. 2019, Vladivostok, Russia), IEEE, 2019. – URL : https://ieeexplore.ieee.org/document/ 8933842 (дата обращения: 23.05.2020).
69. Привалов, А.С. Мобильное приложение для определения характеристик транспортных потоков [Электронный ресурс] / А.С. Привалов, О.К. Головнин // Математика. Компьютер. Образование : тезисы докладов двадцать шестой международной конференции (Пущино, 28 января – 2 февраля 2019 г.). – URL : http://mce.su/rus/archive/mce26/doc333051/ (дата обращения: 23.05.2020).
70. Привалов, А.С. Автоматизированная система извлечения информации о транспортном потоке из аудиозаписи [Текст] / А.С. Привалов, О.К. Головнин // XV Королёвские чтения, посвященные 100-летию со дня рождения Д.И. Козлова : сборник трудов междунар. молодежной научной конф.(Самара, 8–10 октября 2019 г.). – Самара : АНО «Издательство СНЦ», 2019. – Т. 1. – С. 540.
71. Головнин, О.К. Алгоритм детектирования элементарных признаков акустического излучения транспортных средств [Текст] / О.К. Головнин, А.С. Привалов // Перспективные информационные технологии (ПИТ 2019) : труды Международной научно-технической конференции (Самара, 24–26 июня 2019 г.) ; под ред. С.А. Прохорова. – Самара : Издательство Самарского научного центра РАН, 2019. – С. 430–432.
72. Privalov, A.S. System of audio mining for obtaining traffic flow characteristics [Текст] / A.S Privalov, O.K. Golovnin // Information Technologies for Intelligent Decision Making Support : Proceedings of the 7th All-Russian Scientific Conference, May 28–30, 2019, Ufa. – Ufa, 2019. – Vol. 1. – P. 7–10.

ПРИЛОЖЕНИЕ А   
Функции активации нейронов

В таблице А.1 представлен перечень наиболее известных функций активации.

Таблица А.1 – Примеры функций активации

|  |  |  |
| --- | --- | --- |
| Название | Формула | Область значений |
| Линейная |  | (-∞, ∞) |
| Полулинейная |  | (0, ∞) |
| Линейная с насыщением |  | (-1, 1) |
| Логистическая |  | (0, 1) |
| Гиперболический тангенс |  | (-1, 1) |
| Рациональная |  | (-1, 1) |
| Синусоидальная |  | (-1, 1) |
| Экспоненциальная |  | (0, ∞) |
| Гаусса |  | (-∞, ∞) |
| Пороговая |  | (0, ∞) |

ПРИЛОЖЕНИЕ Б   
Руководство пользователя

Б.1 Назначение системы

Автоматизированная система детектирования признаков акустического излучения ТС на аудиозаписях с использование СНС. Система позволяет вести учет количества проезжающих ТС на совершенной аудиозаписи, а также сохранять данные результаты в базу данных.

Б.2 Требования к обеспечению

Минимальные требования к техническому и программному обеспечению серверной части системы:

* тактовая частота процессора – не менее 1,5 ГГц;
* число ядер процессора — не менее 2;
* объем оперативной памяти – не менее 2 Гб;
* дисковая подсистема – не менее 5 Гб свободного места на диске;
* операционная система – любая;
* скорость сетевого адаптера – не менее 100 Мбит/сек.

Для клиентской части понадобятся следующее аппаратное и программное обеспечение:

* мобильный телефон под управление операционной системы Android не ниже версии 8.0;
* мобильный телефон под управление операционной системы iOS не ниже версии 13.0

Б.3 Описание функционала пользователя

На рисунке Б.1 представлена форма аутентификации пользователя. Для входа в систему необходимо ввести логин и пароль и нажать на кнопку «Войти в систему». Учетная запись нужна для того, чтобы можно было идентифицировать отправителя аудиозаписи.

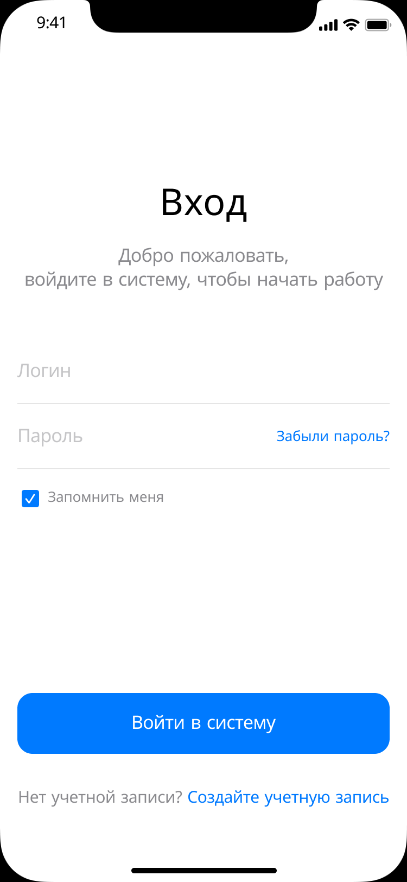


Рисунок Б.1 – Форма аутентификации

Если пользователя еще не существовало в системе, он может создать учетную запись при нажатии на кнопку «Создайте учетную запись» (рисунок Б.2).



Рисунок Б.2 – Форма создании учетной записи

После прохождения аутентификации пользователь будет перенаправлен на карту, где расположена карта с произведенными замерами (рисунок Б.3).

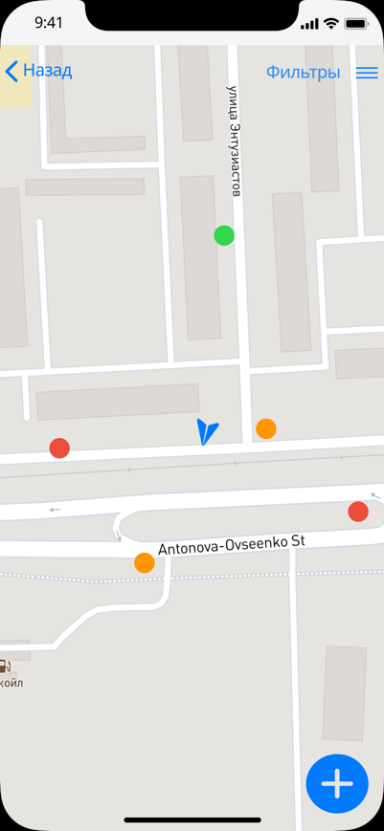


Рисунок Б.3 – Карта с произведенными замерами

При нажатии на «+» пользователь будет перенаправлен на форму записи аудиофайла. Данная форма представлена на рисунке Б.4.



Рисунок Б.4 – Форма записи аудиофайла

После отправки файла и получения ответа с сервера пользователь перейдет на форму, которая содержит информацию о ТП, аудиозапись которого он сделал. Пример данной формы представлен на рисунке Б.5.



Рисунок Б.5 – Форма обзора аудиозаписи

При нажатии на копку «Назад» на форме с картой, пользователь перейдет на списковый обзор замеров, совершенных с помощью системы. Пример данной формы представлен на рисунке Б.6.

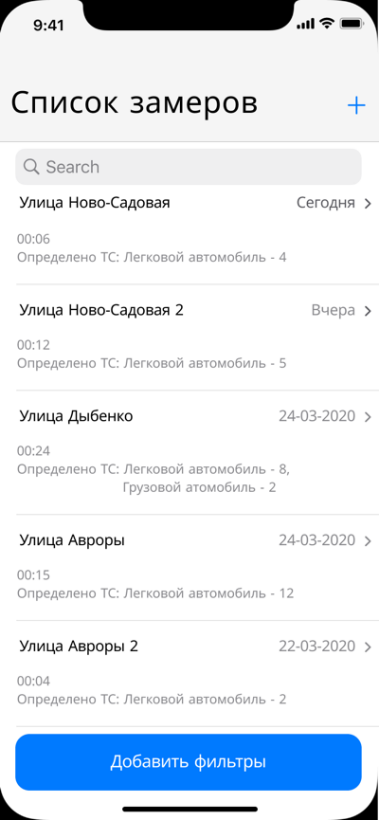


Рисунок Б.6 – Список замеров

В системе предусмотрена фильтрация данных. При нажатии на кнопку «Добавить фильтры» пользователь перейдет на форму выбора фильтров, которые применяются для фильтрации данных. Пример данной формы показан на рисунке Б.7.

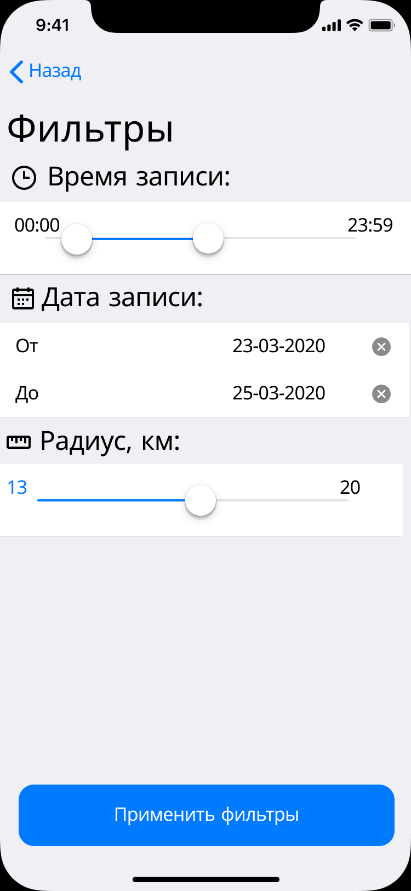


Рисунок Б.7 – Фильтры

ПРИЛОЖЕНИЕ В   
Листинг программы

import pandas as pd

metadata = pd.read\_csv('../UrbanSound Dataset sample/metadata/UrbanSound8K.csv')

metadata.head()

print(metadata.class\_name.value\_counts())

import pandas as pd

import os

import librosa

import librosa.display

from helpers.wavfilehelper import WavFileHelper

wavfilehelper = WavFileHelper()

audiodata = []

for index, row in metadata.iterrows():

file\_name = os.path.join(os.path.abspath('/Volumes/Untitled/ML\_Data/Urban Sound/UrbanSound8K/audio/'),'fold'+str(row["fold"])+'/',str(row["slice\_file\_name"]))

data = wavfilehelper.read\_file\_properties(file\_name)

audiodata.append(data)

audiodf = pd.DataFrame(audiodata, columns=['num\_channels','sample\_rate','bit\_depth'])

print(audiodf.num\_channels.value\_counts(normalize=True))

import librosa

from scipy.io import wavfile as wav

import numpy as np

filename = '../UrbanSound Dataset sample/audio/100852-0-0-0.wav'

librosa\_audio, librosa\_sample\_rate = librosa.load(filename)

scipy\_sample\_rate, scipy\_audio = wav.read(filename)

print('Original sample rate:', scipy\_sample\_rate)

print('Librosa sample rate:', librosa\_sample\_rate)

print('Original audio file min~max range:', np.min(scipy\_audio), 'to', np.max(scipy\_audio))

print('Librosa audio file min~max range:', np.min(librosa\_audio), 'to', np.max(librosa\_audio))

import matplotlib.pyplot as plt

# Original audio with 2 channels

plt.figure(figsize=(12, 4))

plt.plot(scipy\_audio)

plt.figure(figsize=(12, 4))

plt.plot(librosa\_audio)

mfccs = librosa.feature.mfcc(y=librosa\_audio, sr=librosa\_sample\_rate, n\_mfcc=40)

print(mfccs.shape)

import librosa.display

librosa.display.specshow(mfccs, sr=librosa\_sample\_rate, x\_axis='time')

def extract\_features(file\_name):

try:

audio, sample\_rate = librosa.load(file\_name, res\_type='kaiser\_fast')

mfccs = librosa.feature.mfcc(y=audio, sr=sample\_rate, n\_mfcc=40)

mfccsscaled = np.mean(mfccs.T,axis=0)

except Exception as e:

print("Error encountered while parsing file: ", file)

return None

return mfccsscaled

import pandas as pd

import os

import librosa

# Set the path to the full UrbanSound dataset

fulldatasetpath = '/Volumes/Untitled/ML\_Data/Urban Sound/UrbanSound8K/audio/'

metadata = pd.read\_csv('../UrbanSound Dataset sample/metadata/UrbanSound8K.csv')

features = []

# Iterate through each sound file and extract the features

for index, row in metadata.iterrows():

file\_name = os.path.join(os.path.abspath(fulldatasetpath),'fold'+str(row["fold"])+'/',str(row["slice\_file\_name"]))

class\_label = row["class\_name"]

data = extract\_features(file\_name)

features.append([data, class\_label])

# Convert into a Panda dataframe

featuresdf = pd.DataFrame(features, columns=['feature','class\_label'])

print('Finished feature extraction from ', len(featuresdf), ' files')

from sklearn.preprocessing import LabelEncoder

from keras.utils import to\_categorical

X = np.array(featuresdf.feature.tolist())

y = np.array(featuresdf.class\_label.tolist())

le = LabelEncoder()

yy = to\_categorical(le.fit\_transform(y))

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, yy, test\_size=0.2, random\_state = 42)

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Dropout, Activation, Flatten

from keras.layers import Convolution2D, MaxPooling2D

from keras.optimizers import Adam

from keras.utils import np\_utils

from sklearn import metrics

num\_labels = yy.shape[1]

filter\_size = 2

# Construct model

model = Sequential()

model.add(Dense(256, input\_shape=(40,)))

model.add(Activation('relu'))

model.add(Dropout(0.5))

model.add(Dense(256))

model.add(Activation('relu'))

model.add(Dropout(0.5))

model.add(Dense(num\_labels))

model.add(Activation('softmax'))

model.compile(loss='categorical\_crossentropy', metrics=['accuracy'], optimizer='adam')

model.summary()

score = model.evaluate(x\_test, y\_test, verbose=0)

accuracy = 100\*score[1]

print("Pre-training accuracy: %.4f%%" % accuracy)

from keras.callbacks import ModelCheckpoint

from datetime import datetime

num\_epochs = 100

num\_batch\_size = 32

checkpointer = ModelCheckpoint(filepath='saved\_models/weights.best.basic\_mlp.hdf5',

verbose=1, save\_best\_only=True)

start = datetime.now()

model.fit(x\_train, y\_train, batch\_size=num\_batch\_size, epochs=num\_epochs, validation\_data=(x\_test, y\_test), callbacks=[checkpointer], verbose=1)

duration = datetime.now() - start

print("Training completed in time: ", duration)

score = model.evaluate(x\_train, y\_train, verbose=0)

print("Training Accuracy: ", score[1])

score = model.evaluate(x\_test, y\_test, verbose=0)

print("Testing Accuracy: ", score[1])

import librosa

import numpy as np

def extract\_feature(file\_name):

try:

audio\_data, sample\_rate = librosa.load(file\_name, res\_type='kaiser\_fast')

mfccs = librosa.feature.mfcc(y=audio\_data, sr=sample\_rate, n\_mfcc=40)

mfccsscaled = np.mean(mfccs.T,axis=0)

except Exception as e:

print("Error encountered while parsing file: ", file)

return None, None

return np.array([mfccsscaled])

def print\_prediction(file\_name):

prediction\_feature = extract\_feature(file\_name)

predicted\_vector = model.predict\_classes(prediction\_feature)

predicted\_class = le.inverse\_transform(predicted\_vector)

print("The predicted class is:", predicted\_class[0], '\n')

predicted\_proba\_vector = model.predict\_proba(prediction\_feature)

predicted\_proba = predicted\_proba\_vector[0]

for i in range(len(predicted\_proba)):

category = le.inverse\_transform(np.array([i]))

print(category[0], "\t\t : ", format(predicted\_proba[i], '.32f') )

filename = '../UrbanSound Dataset sample/audio/100852-0-0-0.wav'

print\_prediction(filename)

filename = '../UrbanSound Dataset sample/audio/103199-4-0-0.wav'

print\_prediction(filename)

filename = '../UrbanSound Dataset sample/audio/101848-9-0-0.wav'

print\_prediction(filename)

filename = '../UrbanSound Dataset sample/audio/100648-1-0-0.wav'

print\_prediction(filename)

filename = '../Evaluation audio/dog\_bark\_1.wav'

print\_prediction(filename)

filename = '../Evaluation audio/drilling\_1.wav'

print\_prediction(filename)

filename = '../Evaluation audio/gun\_shot\_1.wav'

print\_prediction(filename)

filename = '../Evaluation audio/siren\_1.wav'

print\_prediction(filename)

import numpy as np

max\_pad\_len = 174

def extract\_features(file\_name):

try:

audio, sample\_rate = librosa.load(file\_name, res\_type='kaiser\_fast')

mfccs = librosa.feature.mfcc(y=audio, sr=sample\_rate, n\_mfcc=40)

pad\_width = max\_pad\_len - mfccs.shape[1]

mfccs = np.pad(mfccs, pad\_width=((0, 0), (0, pad\_width)), mode='constant')

except Exception as e:

print("Error encountered while parsing file: ", file\_name)

return None

return mfccs

import pandas as pd

import os

import librosa

# Set the path to the full UrbanSound dataset

fulldatasetpath = '/Volumes/Untitled/ML\_Data/Urban Sound/UrbanSound8K/audio/'

metadata = pd.read\_csv('../UrbanSound Dataset sample/metadata/UrbanSound8K.csv')

features = []

# Iterate through each sound file and extract the features

for index, row in metadata.iterrows():

file\_name = os.path.join(os.path.abspath(fulldatasetpath),'fold'+str(row["fold"])+'/',str(row["slice\_file\_name"]))

class\_label = row["class\_name"]

data = extract\_features(file\_name)

features.append([data, class\_label])

# Convert into a Panda dataframe

featuresdf = pd.DataFrame(features, columns=['feature','class\_label'])

print('Finished feature extraction from ', len(featuresdf), ' files')

from sklearn.preprocessing import LabelEncoder

from keras.utils import to\_categorical

# Convert features and corresponding classification labels into numpy arrays

X = np.array(featuresdf.feature.tolist())

y = np.array(featuresdf.class\_label.tolist())

# Encode the classification labels

le = LabelEncoder()

yy = to\_categorical(le.fit\_transform(y))

# split the dataset

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, yy, test\_size=0.2, random\_state = 42)

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Dropout, Activation, Flatten

from keras.layers import Convolution2D, Conv2D, MaxPooling2D, GlobalAveragePooling2D

from keras.optimizers import Adam

from keras.utils import np\_utils

from sklearn import metrics

num\_rows = 40

num\_columns = 174

num\_channels = 1

x\_train = x\_train.reshape(x\_train.shape[0], num\_rows, num\_columns, num\_channels)

x\_test = x\_test.reshape(x\_test.shape[0], num\_rows, num\_columns, num\_channels)

num\_labels = yy.shape[1]

filter\_size = 2

# Construct model

model = Sequential()

model.add(Conv2D(filters=16, kernel\_size=2, input\_shape=(num\_rows, num\_columns, num\_channels), activation='relu'))

model.add(MaxPooling2D(pool\_size=2))

model.add(Dropout(0.2))

model.add(Conv2D(filters=32, kernel\_size=2, activation='relu'))

model.add(MaxPooling2D(pool\_size=2))

model.add(Dropout(0.2))

model.add(Conv2D(filters=64, kernel\_size=2, activation='relu'))

model.add(MaxPooling2D(pool\_size=2))

model.add(Dropout(0.2))

model.add(Conv2D(filters=128, kernel\_size=2, activation='relu'))

model.add(MaxPooling2D(pool\_size=2))

model.add(Dropout(0.2))

model.add(GlobalAveragePooling2D())

model.add(Dense(num\_labels, activation='softmax'))

model.compile(loss='categorical\_crossentropy', metrics=['accuracy'], optimizer='adam')

model.summary()

# Calculate pre-training accuracy

score = model.evaluate(x\_test, y\_test, verbose=1)

accuracy = 100\*score[1]

print("Pre-training accuracy: %.4f%%" % accuracy)

from keras.callbacks import ModelCheckpoint

from datetime import datetime

#num\_epochs = 12

#num\_batch\_size = 128

num\_epochs = 72

num\_batch\_size = 256

checkpointer = ModelCheckpoint(filepath='saved\_models/weights.best.basic\_cnn.hdf5',

verbose=1, save\_best\_only=True)

start = datetime.now()

model.fit(x\_train, y\_train, batch\_size=num\_batch\_size, epochs=num\_epochs, validation\_data=(x\_test, y\_test), callbacks=[checkpointer], verbose=1)

duration = datetime.now() - start

print("Training completed in time: ", duration)

score = model.evaluate(x\_train, y\_train, verbose=0)

print("Training Accuracy: ", score[1])

score = model.evaluate(x\_test, y\_test, verbose=0)

print("Testing Accuracy: ", score[1])

def print\_prediction(file\_name):

prediction\_feature = extract\_features(file\_name)

prediction\_feature = prediction\_feature.reshape(1, num\_rows, num\_columns, num\_channels)

predicted\_vector = model.predict\_classes(prediction\_feature)

predicted\_class = le.inverse\_transform(predicted\_vector)

print("The predicted class is:", predicted\_class[0], '\n')

predicted\_proba\_vector = model.predict\_proba(prediction\_feature)

predicted\_proba = predicted\_proba\_vector[0]

for i in range(len(predicted\_proba)):

category = le.inverse\_transform(np.array([i]))

print(category[0], "\t\t : ", format(predicted\_proba[i], '.32f') )

filename = '../UrbanSound Dataset sample/audio/100852-0-0-0.wav'

print\_prediction(filename)

filename = '../UrbanSound Dataset sample/audio/103199-4-0-0.wav'

print\_prediction(filename)

filename = '../UrbanSound Dataset sample/audio/101848-9-0-0.wav'

print\_prediction(filename)

filename = '../UrbanSound Dataset sample/audio/100648-1-0-0.wav'

print\_prediction(filename)

filename = '../Evaluation audio/dog\_bark\_1.wav'

print\_prediction(filename)

filename = '../Evaluation audio/drilling\_1.wav'

print\_prediction(filename)

filename = '../Evaluation audio/gun\_shot\_1.wav'

print\_prediction(filename)

from helpers.wavfilehelper import WavFileHelper

wavfilehelper = WavFileHelper()

data = wavfilehelper.read\_file\_properties('../Urban Data sample set/audio/30204.wav')

print(data)

export default class App extends Component {

render() {

return (

<View style={styles.container}>

<Geolocation>

<Context.Consumer>

{({loading}) =>

loading ? <Loading visible={true} /> : <AppWithRoute />

}

</Context.Consumer>

</Geolocation>

</View>

);

}

}

const styles = StyleSheet.create({

container: {

width: '100%',

height: '100%',

},

});

class ViewRoad extends Component {

static navigationOptions = ({navigation}) => ({

title: 'View Road',

headerRight: () => (

<Button title="Edit" onPress={() => navigation.navigate('EditRoad')} />

),

});

state = {

modalVisible: false,

damaged\_segments: [],

selected\_segment: {},

latitude: null,

longitude: null,

};

async updateCoordinate(current\_coordinate) {

const distance = getDistanceFrom(current\_coordinate, this.state);

if (this.state.latitude === null || distance > 50) {

await this.setCoordinate(current\_coordinate);

this.loadDamagedSegments();

}

}

setCoordinate(coordinate) {

this.setState({

...coordinate,

});

}

loadDamagedSegments() {

const {latitude, longitude} = this.state;

axios

.get(webservice + '/damaged\_road/' + latitude + '/' + longitude)

.then((response) => {

this.setState({damaged\_segments: response.data});

});

}

componentDidUpdate(prevProps) {

if (this.isScreenFocused(prevProps)) {

this.loadDamagedSegments();

}

}

isScreenFocused(prevProps) {

const isPreviouslyFocused = prevProps.isFocused;

const isCurrentFocused = this.props.isFocused;

return isCurrentFocused && !isPreviouslyFocused;

}

showModal(selected\_segment) {

this.setState({modalVisible: true, selected\_segment});

}

hideModal() {

this.setState({modalVisible: false});

}

render() {

return (

<Context.Consumer>

{({latitude, longitude}) => {

this.updateCoordinate({latitude, longitude});

return (

<Fragment>

<Map>

<Polylines

damaged\_segments={this.state.damaged\_segments}

onPress={(segment) => this.showModal(segment)}

/>

</Map>

<Modal

visible={this.state.modalVisible}

onRequestClose={() => this.hideModal()}

animationType="slide">

<SegmentDetail segment={this.state.selected\_segment} />

</Modal>

</Fragment>

);

}}

</Context.Consumer>

);

}

}

export default withNavigationFocus(ViewRoad);

class SegmentDetail extends Component {

renderItem(label, value, icon) {

return (

<View style={styles.item}>

<View style={styles.itemIcon}>

<Icon name={icon} size={30} />

</View>

<View style={styles.itemData}>

<Text style={styles.itemLabel}>{label}</Text>

<Text style={styles.itemValue}>{value}</Text>

</View>

</View>

);

}

render() {

const {segment} = this.props;

return (

<View style={styles.container}>

<Text style={styles.title}>Segment Detail</Text>

{this.renderItem(

'Damage Type',

segment.damage\_type.name,

'road-variant',

)}

{this.renderItem(

'Damage Level',

segment.damage\_level.name,

'signal-cellular-2',

)}

{this.renderItem('Information', segment.information, 'information')}

</View>

);

}

}