Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Уфимский университет науки и технологий»

На правах рукописи

ЛУШНИКОВ НИКИТА ДМИТРИЕВИЧ

# МОДЕЛИ И АЛГОРИТМЫ МУЛЬТИМОДАЛЬНОЙ БИОМЕТРИЧЕСКОЙ АУТЕНТИФИКАЦИИ НА ОСНОВЕ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

Специальность 2.3.6. Методы и системы защиты информации, информационная безопасность

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор физико–математических наук, доцент
Исмагилова Альбина Сабирьяновна

Уфа – 2025

ОГЛАВЛЕНИЕ

# ВВЕДЕНИЕ

**Актуальность темы исследования.** Процедуры идентификации и аутентификации пользователей являются одними из важнейших механизмов защиты современных информационных систем, реализуются на первичных этапах их обороны, в связи с чем данные механизмы наиболее часто подвергаются различным атакам со стороны злоумышленников. Стойкость ко взлому процедур идентификации и аутентификации субъектов доступа во многом определяет общий уровень защищенности всей информационной системы, поэтому качеству реализации данных процедур всегда уделяется особое внимание. В настоящее время биометрические системы становятся одними из наиболее перспективных средств аутентификации пользователей. Данные системы обеспечивают удобство аутентификации человека с одной стороны и высокий уровень безопасности с другой. В отличие от паролей и носителей информации, которые могут быть утеряны, украдены или скомпрометированы, биометрические системы основаны на человеческих уникальных характеристиках, которые являются неотъемлемой частью пользователя информационной системы. Это делает подделку или хищение аутентифицирующей информации практически невозможной.

Однако, к основным недостаткам биометрических систем аутентификации следует отнести наличие для них ошибочных отказов и ошибочных подтверждений, а также возможность формирования поддельных биометрических образцов, в том числе реализуемых с использованием дипфейк-технологий на основе нейронных сетей, которые олицетворяют собой методику синтеза изображений лица и/или голоса. По данным Министерства внутренних дел, в 2024 году в России зарегистрированы 765,4 тыс. преступлений, совершенных с использованием дипфейк-технологий, что на 13,1% больше, чем за предыдущий год. При этом убытки от них в 2024 году достигли 300 млрд. рублей.

Один из подходов к повышению качества работы биометрических систем, уменьшению количества ошибочных отказов и подтверждений, устойчивости к подделкам, заключается в применении мультимодальности - использование для принятия решений нескольких биометрических характеристик, например, изображения лица и голоса человека. Однако, применение данного подхода на практике характеризуется рядом сложностей, требующих решения. В частности, необходимо разработать подходы к комбинированию различных биометрических признаков, к формированию итогового решения об аутентификации пользователя на основании нескольких биометрических характеристик. Также необходимо исследовать этапы предобработки биометрических характеристик для устранения возможных шумов.

К числу наиболее актуальных проблем в настоящее время следует отнести атаки и угрозы с использованием дипфейк-технологий на основе нейронных сетей, который олицетворяют собой методику синтеза изображений лица и/или голоса. Технологии дипфейк становятся доступнее, а их использование – проще. Изначально злоумышленники применяли дипфейки для сложных адресных атак с потенциально большой выгодой. Зачастую злоумышленники создают дипфейки работодателей, сотрудников государственных органов, известных личностей. Постепенно круг потенциальных жертв расширяется, а атаки становятся более массовыми. По данным Министерства внутренних дел, в 2024 году в России зарегистрированы 765,4 тыс. преступлений, совершенных с использованием информационно-телекоммуникационных технологий, что на 13,1% больше, чем в предыдущем году. Убытки россиян достигли 300 млрд. рублей [35].

Атаки с использованием дипфейк-технологий на основе нейронных сетей приводят к фальсификации, компрометации и утечке данных (в т.ч. биометрических персональных данных). Реализация данных угроз может спровоцировать несанкционированный доступ как к информационной системе

в целом, так и к мультимодальной системе биометрической аутентификации пользователей данной информационной системы в частности.

Таким образом, на данный момент в области биометрических технологий наиболее актуальной проблемой является проблема обеспечения защищенности мультимодальных биометрических систем от несанкционированного доступа. К числу нерешенных задач в данной области исследований следует отнести повышение надежности распознавания пользователей информационной системы на основе нейронных сетей.

Основная задача исследований в рассматриваемой предметной области состоит в высоконадежном распознавании личности (высоконадежной биометрической аутентификацией). На данный момент в области биометрических технологий наиболее актуальной проблемой является проблема обеспечения защищенности мультимодальных биометрических систем от несанкционированного доступа. К числу нерешенных задач следует отнести повышение надежности распознавания пользователей информационной системы на основе нейронных сетей, поддержание высокого уровня точности процедуры биометрической аутентификации и показателей достоверности.

**Степень разработанности темы исследований.** Ранее процессы биометрической аутентификации были исследованы в трудах Ахметова Б.С., Бабенко Л.К., Безяева А.В., Брюхомицкого Ю.А., Васильева В. И., Волчихина В.И., Епифанцева Б.Н., Дельдаго Х., Еременко А. В., Иванова А.И., Катасева А.С., Ложникова П.С., Маршалко Г.Б., Машкиной И. В., Ниргуде М., Пирале Д., Рабинера Л. Р., Сабанова А. Г., Сулавко А. Е., Тодиско М., Цзюана Б. Х., Частиковой В. А., Шелупанова А. А., Эванса Н. и др.

Так, Рабинер Л.Р. в своих трудах проводил исследования по распознаванию речи с применением статистических данных и метода скрытых марковских моделей. Васильевым В.И. были предложены новые методы и алгоритмы машинного обучения в процессах идентификации биометрических систем на базе статических признаков. Шелупановым А.А. и Сабановым А.Г.

приведен анализ цифровой идентификации и аутентификации субъектов доступа применительно к задаче управления доступом к информационным ресурсам, а также предложены критерии доверия к результатам идентификации и аутентификации. В работах Ложникова П.С. представлены методы машинного обучения и новые подходы к применению искусственного интеллекта в процессах биометрической аутентификации. Тодиско М., Дельгадо Х. и Еванс Н. в своих исследованиях вывели новый извлеченный биометрический признак – Q-константный кепстральный коэффициент.

Применение биометрических систем регламентируются рядом российских и зарубежных нормативно–правовых документов, которые описывают терминологию, общие понятия, а также процессы аутентификации. Так, ГОСТ Р 54411–2018 «Информационные технологии. Биометрия. Мультимодальные и другие мультибиометрические технологии» устанавливает требования к описанию общих понятий методов мультибиометрических объединений, подходы к сравнению биометрических характеристик и принятию решений.

Однако, несмотря на значительное количество работ в данной области, ранее вышеперечисленные сложности, свойственные мультимодальным биометрическим системам, остаются в настоящее время не до конца исследованными, что делает тему диссертационной работы актуальной.

**Объектом исследования** является мультимодальная система биометрической аутентификации пользователей информационной системы.

**Предметом исследования** являются модели и алгоритмы мультимодальной системы биометрической аутентификации пользователей по изображению лица и по голосу.

**Целью диссертационной работы** является повышение надежности биометрической аутентификации пользователей информационной системы за счет конкатенации их биометрических признаков лица и голоса, а также совершенствования алгоритмов предварительной обработки биометрических данных.

Для выполнения обозначенной цели в диссертационной работе поставлены и решены следующие **задачи исследования:**

1. Разработать модели биометрической аутентификации пользователей информационной системы по изображению лица и по голосу в составе мультимодальной биометрической системы с применением предварительной шумоочистки данных.
2. Разработать алгоритмы обучения сверточной нейронной сети в составе мультимодальной системы биометрической аутентификации.
3. Разработать программный комплекс мультимодальной системы биометрической аутентификации пользователей, реализующий разработанные модели и алгоритмы.
4. Исследовать эффективность разработанных решений.

**Научная новизна**

1. Разработана модель биометрической аутентификации пользователей информационной системы по изображению лица, отличающаяся предварительной обработкой данных с применением фильтра Калмана, позволяющая в составе мультимодальной биометрической системы повысить помехоустойчивость алгоритма распознавания пользователей информационной системы.
2. Разработана модель биометрической аутентификации пользователей информационной системы по голосу, отличающаяся применением метода шумоочистки и составом выделяемых признаков, позволяющая в составе мультимодальной биометрической системы уменьшить показатели ошибок первого и второго рода при распознавании пользователей информационной системы.
3. Предложен алгоритм обучения сверточной нейронной сети модели биометрической аутентификации пользователей информационной системы по изображению лица в составе мультимодальной биометрической системы, отличающийся извлечением дополнительных дескрипторов

изображения, что позволяет минимизировать возможность синтеза изображений (дипфейк).

4. Предложен алгоритм обучения сверточной нейронной сети модели биометрической аутентификации пользователей информационной системы по голосу в составе мультимодальной системы биометрической аутентификации, отличающийся извлечением дополнительных речевых признаков, что позволяет минимизировать возможность синтеза голоса (дипфейк).

**Практическая значимость результатов работы.** Разработан программный комплекс мультимодальной системы биометрической аутентификации пользователей информационной системы на основе сверточной нейронной сети с применением конкатенации векторов признаков (изображение лица и голос), достигающий точности распознавания 99,31 %.

Модели и алгоритмы мультимодальной системы биометрической аутентификации пользователей информационной системы внедрены в автоматизированные рабочие места ООО «Информзащита» Уфа и ООО «ИТ Энигма Уфа», применены в процессах Координационного центра ФГБОУ ВО «Уфимский университет науки и технологий».

**Положения, выносимые на защиту:**

1. Модель биометрической аутентификации пользователей информационной системы по изображению лица в составе мультимодальной биометрической системы с применением предварительной обработки изображения лица.

2. Модель биометрической аутентификации пользователей информационной системы по голосу в составе мультимодальной биометрической системы с применением предварительной шумоочистки данных.

3. Алгоритмы обучения сверточной нейронной сети в составе мультимодальной системы биометрической аутентификации.

4. Мультимодальная система биометрической аутентификации пользователей информационной системы, реализующая разработанные модели и алгоритмы.

**Достоверность полученных результатов** в диссертационной работе подтверждается теоретической обоснованностью, опубликованными научными публикациями в 30 издании, в том числе 7 научных статей в рецензируемых научных журналах, рекомендованных Высшей аттестационной комиссией Российской Федерации (из них: 6 научных статей К2, 1 научная статья К3), 1 научная статья в научных изданиях, индексируемых базой данных RSCI, 3 статьи в журналах, индексируемых в базах данных Web of Science и Scopus; 3 свидетельства о государственной регистрации программ для ЭВМ, 16 работ в трудах международных и всероссийских конференций.

**Апробация результатов диссертации.** Полученные результаты апробированы на международных и Всероссийских конференциях: Международная научная конференция, посвящённая памяти доктора технических наук, профессора А.А. Тарасова и доктора технических наук, старшего научного сотрудника О.В. Казарина (РГГУ, г. Москва, 2023); Международная конференция и молодежная школа «Информационные технологии и нанотехнологии» ИТНТ (г. Самара, 2023); Международная научная конференция студентов, аспирантов и молодых ученых «Ломоносов» (г. Москва, 2023); Всероссийская научно–практическая конференция «Информационная безопасность цифровой экономики» (г. Улан–Удэ, 2023); Международная научно–практическая конференция «Информационные системы и технологии в моделировании и управлении» (г. Ялта, 2023); Международные конференции по системам управления, математическому моделированию, автоматизации и энергоэффективности «SUMMA» (г. Липецк, 2022, 2023); Всероссийские научные школы-семинары «Современные тенденции развития методов и средств защиты информации» (МТУСИ, г. Москва, 2022, 2023); Международная научная молодежная школа–семинар

«Математическое моделирование, численные методы и комплексы программ» имени Е.В. Воскресенского (г. Саранск, 2022); Российско–Китайский молодежный форум «Волга–Янцзы» (г. Уфа, 2022); Международная научно–техническая конференция «Актуальные проблемы прикладной математики, информатики и механики» (г. Воронеж, 2021); Всероссийская научно–теоретическая конференция «Теория и практика обеспечения информационной безопасности» (МТУСИ, г. Москва, 2021); Сибирские форумы «Информационная безопасность» (СибГУТИ, г. Новосибирск, 2021, 2022); Всероссийские молодежные научно–практические конференции «Информационные технологии обеспечения комплексной безопасности в цифровом обществе» (г. Уфа, 2021, 2022, 2023); Международная молодежная научно–практическая конференция «Математическое моделирование процессов и систем» (г. Стерлитамак, 2021); Международный военно–технический Форум «Армия» (ФГАУ КВЦ «Патриот», г. Кубинка, Московская обл., 2021) и других.

Связь с научными программами:

1. Гранты ИБ МТУСИ при финансовой поддержке ФГБОУ ВО «Московский технический университет связи и информатики» под руководством Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации (Соглашение № 40469-20/2022-к от 30.06.2022 г.).

2. Студенческий стартап (очередь II) при финансовой поддержке Фонда содействия инновациям (Договор № 696ГССС15-L/80924 от 01.12.2022 г.).

3. Грант Республики Башкортостан (государственная поддержка молодых ученых, Приказ Министерства образования и науки Республики Башкортостан № 2987 от 29.11.2022 г.).

Пройдена научно-образовательная стажировка на базе Института системной интеграции и безопасности Томского государственного университета систем управления и радиоэлектроники (ТУСУР). Тема

стажировки: «Угрозы безопасности и принципы формирования наборов данных для систем искусственного интеллекта и машинного обучения» (11.09.2023 – 25.09.2023 гг.).

Полученные результаты диссертации представлены на научных семинарах Института информатики, математики и робототехники ФГБОУ ВО «Уфимский университет науки и технологий».

**Соответствие паспорту специальности.** Содержание диссертационного исследования соответствует следующим пунктам паспорта научной специальности 2.3.6 «Методы и системы защиты информации, информационная безопасность»:

п. 12. Технологии идентификации и аутентификации пользователей и субъектов информационных процессов. Системы разграничения доступа.

п. 15. Принципы и решения (технические, математические, организационные и др.) по созданию новых и совершенствованию существующих средств защиты информации и обеспечения информационной безопасности.

**Личный вклад автора.** Все изложенные результаты в диссертации, в том числе программные реализации предложенных в работе алгоритмов, получены автором самостоятельно. Проработка цели и задачи, способов их решения и вариантов представления результатов реализован автором совместно с научным руководителем. Ключевые публикации подготовлены в соавторстве с научным руководителем д. ф. – м. н., профессором Исмагиловой А. С.

**Объем и структура диссертации.** Диссертационная работа состоит из введения, 4 глав, заключения, списка литературы и 3 приложений. Текст диссертационного исследования представлен на 188 страницах. Список литературы состоит из 128 источников.

# ГЛАВА 1. АНАЛИЗ СОВРЕМЕННОГО СОСТОЯНИЯ ИССЛЕДОВАНИЙ В ОБЛАСТИ БИОМЕТРИЧЕСКОЙ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ

Задачи управления доступом пользователей актуальны для любой информационной системы. С каждым зарегистрированным в сети пользователем связана определенная информация, которая идентифицирует его. Внедрение криптографических и биометрических систем положительно влияет на разработку инновационных решений для обеспечения информационной безопасности. Особенно перспективными являются мультимодальная и мультибиометрическая системы аутентификации, объединившие в себе биометрические признаки и методы их преобразования в основные последовательности [8].

В трудах российских и зарубежных ученых проведены исследования по биометрической системе аутентификации на основе статических и динамических методов с применением аппаратных средств [16, 70, 98]. Предложены методы и модели по совершенствованию способов и средств защиты информации применительно к задаче идентификации и аутентификации пользователей информационной системы [72]. Для повышения точности процедуры аутентификации на основе биометрических признаков получены результаты применения нейронных сетей с обучающими выборками, которые являются основополагающими и фундаментальными [22-23]. Разработаны новые архитектуры нейронных сетей, составлены новые обучающие выборки, проведено сравнение с международными датасетами [57].

Процессы идентификации и аутентификации нуждаются в применении инновационных алгоритмов машинного обучения на основе биометрических признаков. Для формирования обучающей выборки и получения высоких

показателей обучения необходимо обработать и преобразовать входные данные.

В главе проанализированы основные стандарты и нормативно–правовые акты по рассматриваемой теме исследования. Приведены основные характеристики мультимодальной биометрической системы аутентификации. Проведена оценка вероятностных характеристик, проанализирован уровень принятия решения биометрической системы аутентификации, представлен метод нормализации степеней схожести биометрических образов.

## 1.1. Нормативно–правовое регулирование биометрической системы аутентификации

Биометрические системы регламентируются рядом российских и зарубежных нормативно–правовых документов, которые описывают терминологию, общие понятия, а также процессы идентификации и аутентификации.

Согласно Федеральному закону от 29.12.2022 № 572–ФЗ «Об осуществлении идентификации и (или) аутентификации физических лиц с использованием биометрических персональных данных, о внесении изменений в отдельные законодательные акты Российской Федерации и признании утратившими силу отдельных положений законодательных актов Российской Федерации», вектор единой биометрической системы – персональные данные, полученные в результате математического преобразования биометрических персональных данных физического лица, содержащихся в единой биометрической системе, которое произведено с использованием информационных технологий и технических средств.

Используемая для аутентификации биометрических персональных данных обработка в информационных системах организаций, осуществляющая аутентификацию на основе биометрических персональных данных физических лиц, допускается с использованием векторов единой биометрической системы, в том числе с использованием векторов,

являющихся результатом обработки биометрических персональных данных, размещенных в единой биометрической системе [2].

При осуществлении процедур идентификации и аутентификации необходимо проанализировать информационную систему, состав мер защиты информации и их базовые наборы.

В международном методическом ресурсе NIST Special Publications 800 Series стоит обратить внимание на ряд документов [12-14].

Данная серия документов содержит технические рекомендации при осуществлении процессов идентификации и аутентификации. Рекомендации описывают процессы удаленной идентификации и аутентификации пользователей, взаимодействующих с государственными информационными системами через открытые сети. Определен перечень технических требований для каждого из четырех уровней проверки личности, регистрации, токенов, процессов управления, протоколов аутентификации [5].

Мультимодальные и мультибиометрические технологии регламентированы ГОСТ Р 54411–2018 «Информационные технологии. Биометрия. Мультимодальные и другие мультибиометрические технологии». Согласно приведенному стандарту, биометрическая характеристика – биологическая или поведенческая характеристика субъекта, позволяющая выявить уникальные, стабильные и собираемые биометрические признаки для автоматического распознавания субъекта [8].

Биометрическая модальность – биометрическая характеристика, применяемая в биометрическом процессе [8].

Мультимодальная биометрическая система аутентификации – система, использующая несколько различных биометрических модальностей [8].

Мультибиометрический процесс – процесс, включающий в себя применение биометрического объединения [8].

Мультибиометрия – автоматическое распознавание личности субъекта, основанное на его биологических и поведенческих характеристиках и включающее в себя применение биометрического объединения [8].

Применение мультибиометрической технологии заключается в возможности получения наибольшего количества информации об идентифицируемом субъекте. Обобщенно процесс обработки информации в биометрической системе может быть представлен в виде схемы (Рисунок 1.1) [34].


Рисунок 1.1 – Обобщенная схема обработки информации в биометрической системе

При рассмотрении объединений на уровне регистрации и уровне извлечения признаков удается сформировать объединенный шаблон, который содержит большее количество идентифицирующих признаков, что может минимизировать ошибки первого рода и ошибки второго рода. В случае с объединением на уровне степеней схожести и принятия решений повышения надежности распознавания можно достичь с использованием конфигурации весовых коэффициентов для более надежных и достоверных результатов [6]:

$$\Delta w_i = -2 * \alpha * \Delta y * f'(z) * x_i^* \quad (1)$$

где $w_i$ – $i$-ый вес, $\Delta y$ – ошибка предсказания, $\Delta y = y - y^*$ – величина ошибки предсказания, $\alpha$ – коэффициент скорости обучения, $f'(z)$ – значение производной функции $f$ в точке $z = \sum_{i=1}^{n+1} x_i^* w_i$.

Поскольку мультибиометрия направлена на увеличение надежности системы за счет извлечения и обработки большего количества информации об объекте распознавания, то в достижении эффективных показателей следует воспользоваться объединением отдельных параметров. В зависимости от уровня объединения могут возникать различные типы взаимодействия данных [34]:

– взаимосвязь между модальностями. Имеет отношение к биометрическим образам, которые физически связаны.

– взаимосвязь, возникающая вследствие идентичности биометрических образов. В случае, когда один и тот же биометрический образ или подмножества биометрического образа применяются разными алгоритмами извлечения признаков и алгоритмами сравнения.

– взаимосвязь значений признаков. Подмножество значений признаков, представляющих собой векторы признаков разных модальностей, могут быть взаимосвязаны.

– взаимосвязь экземпляров, возникающая при общей технике эксплуатации. Использование одного и того же устройства регистрации, один и тот же уровень подготовки оператора.

– взаимосвязь экземпляров, возникающая вследствие особенностей субъекта.

Для оценки взаимосвязи используют корреляционный коэффициент, который определяется по формуле [34]:

$$p_{nC} = \frac{n \times N_C^f}{N - N_C^t - N_C^f - n \times N_C^f} \quad (2)$$

где $n$ – общее число тестируемых классификаторов, $N$ – общее число входных данных, $N_C^f$ – число входных данных, ошибочно классифицируемых всеми классификаторами при использовании порога $C$, $N_C^t$ – число входных данных, правильно классифицируемых всеми классификаторами при использовании порога $C$.

Для получения наибольшего эффекта от объединения при формировании мультибиометрической технологии целесообразно применять параметры, коэффициент корреляции между которыми минимальный.

В настоящее время наибольшее практическое применение получили мультибиометрические системы, объединяющие две биометрические системы на уровне принятия решений [34].

Объединение на уровне принятия решения также позволяет использовать биометрические системы различных производителей, а в случае

применения логического правила принятия окончательного решения – оценивать конечные вероятностные характеристики мультибиометрической системы [1].

## 1.2. Анализ научных публикаций в области биометрической аутентификации пользователей на основе нейронных сетей

Большое количество научных исследований и публикаций в области биометрии свидетельствует о повышенном интересе к данному направлению. Научными сотрудниками проводятся исследования по созданию мультимодальных системы биометрической аутентификации пользователей информационной системы и автоматизированных комплексов обучения.

Значительный вклад в становление теоретических основ данного направления внес Спиридонов И. Н., который являлся основоположником научной биометрии, основателем Научно-исследовательского и испытательного центра биометрической техники МГТУ им. Н. Э. Баумана, организатором и председателем некоммерческого партнерства «Русское общество содействия развитию биометрических технологий, систем и коммуникаций».

Рассматривая историю биометрии, нельзя не отметить тот огромный вклад в развитие теории вероятностей и математической статистики, который внесли такие ученые нашей страны, как Бернштейн С. Н., Хинчин А. Я., Слуцкий Е. Е., Хотимский А. И., Ястремский Б. С., Романовский В. И., Немчинов В. С. и многие другие, особенно Колмогоров А.Н. и его школа, получившие мировое признание.

Первый учебник по теории вероятностей был издан в России в 1846 г. Буняковским В. Я. Первая полная сводка биометрических методов была составлена в 1909 г. Леонтовичем А. В.

Группировка первичных биометрических данных, основные характеристики варьирующих объектов, дисперсионный и корреляционно-регрессионный анализ представлен в трудах Лакина Г. Ф. [36].

Для повышения точности распознавания личности при проводимой процедуре биометрической аутентификации используют методы машинного обучения. Значительный вклад в развитие систем искусственного интеллекта, а также интеллектуальных систем управления сложными динамическими объектами в классе многоуровневых иерархических систем внес Васильев В. И. [19].

## 1.3. Обзор (анализ) методов биометрической аутентификации, определение их преимуществ и недостатков

Метод Виолы-Джонса — это алгоритм обнаружения объектов на изображениях в режиме реального времени. Алгоритм может распознавать различные классы изображений, но его основной целью является обнаружение лиц [118].

Этапы алгоритма Виолы–Джонса [118]:

– инициализация параметров (например, окно с ячейками 25 *x* 25);

– окно, сканирующее изображение с выбранными признаками;

– сканирование изображения происходит путем перемещения анализирующего окна по его поверхности, каждый сдвиг равен размеру одной ячейки внутри окна;

– в процессе обработки каждого фрагмента изображения, соответствующего положению окна, производится расчет приблизительно 200 000 различных комбинаций характеристик. Это достигается за счет варьирования масштаба и пространственного расположения признаков относительно окна;

– сканирование повторяется для различных уровней масштабирования. При этом изменяется размер самого сканирующего окна, а не исходного изображения;

– итоговый результат определяется с помощью классификатор [19, 57, 72].

В стандартном методе Виолы–Джонса используются прямоугольные признаки Хаары (Рисунок 1.2).

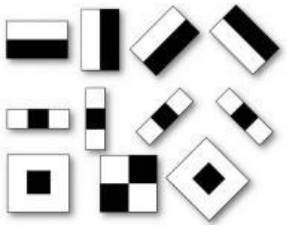

Рисунок 1.2 – Признаки Хаара

В качестве иллюстрации ценообразования деформации графа $G^I$, созданного на основе исходного изображения лица с графом $B$, приведена формула [100]:

$$S_B(G^I, B) = \frac{1}{N}\sum\nolimits_n max S_\Phi(J_N^I, J_N^B) - \frac{\lambda}{E}\sum\nolimits_e \frac{(\Delta x_e^I - \Delta x_e^B)^2}{(\Delta x_e^B)^2} \quad (3)$$

где $S_B$ – ценовая функция, $G^I$ – граф исходного изображения лица, $B$ – граф изображения из базы данных биометрических образов, $N$ – количество вершин, $S_\Phi$ – фазочувствительная функция, $J_N^I, J_N^B$ – джеты исходного изображения и изображения из датасета, состоящего из биометрических образов, $\lambda$ – коэффициент относительной важности топографической информации, $E$ – количество граней, $(\Delta x_e^I - \Delta x_e^B)$ – квадрат разности расстояний между соответствующими вершинами сравниваемых изображений.

К числу методов распознавания личности следует отнести метод главных компонент, основанный на преобразовании Карунена–Лоева [25].

Построение локальных бинарных шаблонов (Local binary patterns, LBP) основано на ассоциации каждого пикселя изображения с группой пикселей его окрестности [26]. Применение оператора LBP позволяет каждому пикселю полутонового изображения поставить в соответствие бинарный код, который описывает его текстурные характеристики.

Оператор работает с группой пикселей и вычисляет бинарный код для центрального пикселя группы. Применение оператора LBP зависит от

количества пикселей окрестности, которыми описывается центральный пиксель области (Рисунок 1.3).

В зависимости от конкретной задачи, качества изображения эмпирическим путем выбирается количество значимых пикселей.

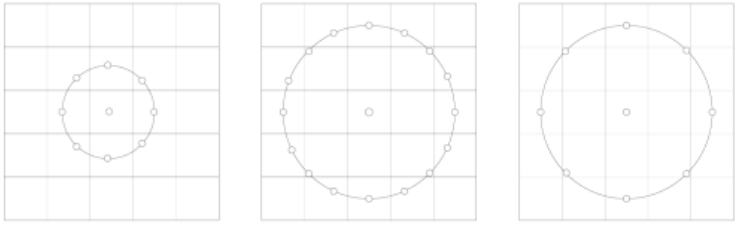

Рисунок 1.3 – Группы пикселей для применения LBP

Каждый пиксель изображения имеет определенное значение интенсивности. Применение оператора LBP позволяет вычислить бинарный код определенного пикселя, используя значения интенсивностей пикселей–соседей (Рисунок 1.4). Каждый квадрат условно описывает пиксель изображения.

Получить полутоновое изображение из полноцветного можно с помощью формулы [101]:

$$g = 0.3R + 0.59G + 0.11B \quad (4)$$

где $R$, $G$, $B$ – значения красного, зеленого и синего цветов соответственно [0, 255], $g$ – значение интенсивности оттенков какого–либо цвета пикселя.

Значение 0 соответствует черному цвету (отсутствие интенсивности), значение 255 – белому (максимальная интенсивность).

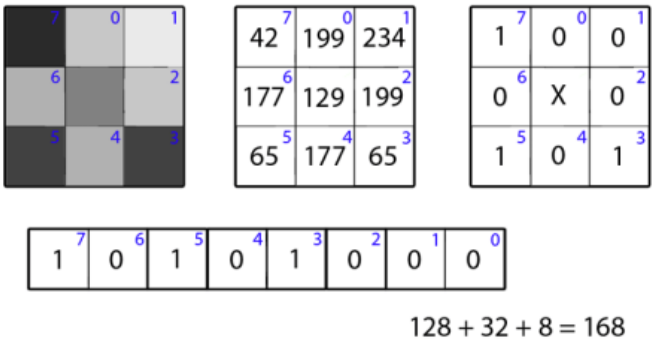


Рисунок 1.4 – Обработка изображений с использованием LBP

Пиксели, значения интенсивности которых больше центрального пикселя (или равное ему), принимают значения «1»; значения интенсивности

пикселей, которые меньше центрального пикселя, равны «0». Получается бинарный код, представляющий окрестность пикселя.

Вычисление LBP с радиусом R и количеством пикселей окрестности P производится следующим образом [16]:

$$LBP_{p,c} = \sum_{p=0}^{p-1} s(g_p - g_c)2^p \quad (5)$$

где $c$ – точка, для которой вычисляется локальный бинарный шаблон, $p = \{0, \dots, P-1\}$ – окрестность точки $c$, $g_p$ – значение интенсивности $p$–ого пикселя, $g_c$ – значение интенсивности центрального пикселя, $s$ – функция, которая возвращает 1 если значение в скобках больше нуля.

Пороговая функция $s(x)$, в которой $x = (g_p - g_c)$, имеет вид:

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, \ x < 0 \end{cases} \quad (6)$$

Применение данного оператора позволяет отслеживать изменения не только каждого пикселя, но и его окрестности для анализа статических свойств человеческого лица.

Следует обратить внимание, что значение каждого бита в коде, полученном с помощью оператора локальных бинарных шаблонов, несет одинаковую значимость. Последовательность из двух бит может иметь совершенно разный смысл (Рисунок 1.5).

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 10011100 = 156 11001001 = 201 |
| 1 | | 1 | 01001110 = 78 11100100 = 228 |
| | | | 00100111 = 39 01110010 = 114 |
| 0 | 0 | 1 | 10010011 = 147 00111001 = 57 |

Рисунок 1.5 – Порядок нахождения коэффициента в зависимости от выбора начального пикселя

Гистограмма — это графическое представление распределения яркости элементов цифрового изображения. На горизонтальной оси отображается уровень яркости, а вертикальная ось показывает количество пикселей, соответствующих каждому уровню яркости. Гистограмма изображения

позволяет оценить количество и разнообразие тонов на изображении, а также общий уровень яркости изображения.

Суть гистограммы ориентированных градиентов заключается в том, что внешний вид и форма локального объекта внутри изображения можно описать распределением градиентов интенсивности или направлениями краев. Изображение делится на небольшие соединенные области, называемые ячейками, и для пикселей в каждой ячейке составляется гистограмма направлений градиента.

Вычисления HOG осуществляется на основе оператора Собеля [26]. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма.

В каждой точке изображения приближённое значение величины градиента можно вычислить путём использования полученных приближенных значений [105]:

$$G = \sqrt{G_x^2 + G_y^2} \tag{7}$$

где $G$ – величина градиента, $G_x$ и $G_y$ – координаты изображения.

$$\theta = \operatorname{arctg}\left(\frac{G_y}{G_x}\right) \tag{8}$$

где $\theta$ – направление градиента.

Лапласиан изображения выделяет области быстрых изменений интенсивности и, таким образом, может использоваться для обнаружения краев изображения [111]. Дискретное приближение лапласиана в конкретном пикселе можно определить, взяв средневзвешенное значение интенсивности пикселей в небольшой окрестности пикселя. Поскольку фильтр Лапласа обнаруживает края изображения, его можно использовать вместе с фильтром Гаусса для снижения чувствительности к шуму, а затем выделить края изображения. Данный метод называется лапласианом гауссианом (LOG) [111]:

$$\nabla^2 f(x,y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \times e^{-\frac{x^2+y^2}{\sigma^2}} \tag{9}$$

где $x, y$ – координаты, $\sigma$ – ширина распространения функции Гаусса, $\nabla$ - градиент функции.

Оператор LOG применяется для определения границ на изображениях. Однако существуют исследования, которые рассматривают использование этого метода для цветных изображений, выделяя каналы в соответствии с выбранной цветовой моделью.

Разнообразие методов реализации маски оператора LOG обусловлено множеством параметров дискретизации. В частности, на маску могут оказывать влияние такие факторы, как значение сигма, размер сетки при применении метода конечных разностей и коэффициент масштабирования [111].

Поскольку лапласиан является дифференциальным оператором, его применение может улучшить области внезапных изменений оттенков серого в изображении и ослабить медленно меняющиеся области оттенков серого.

Таким образом, процесс повышения резкости может выбрать оператор Лапласа для обработки исходного изображения, чтобы создать изображение, описывающее мутацию в градациях серого, а затем наложить лапласовское изображение на исходное изображение для создания изображения с повышенной резкостью (Рисунок 1.6):

$$\nabla^2 f = \Delta f = \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right) \quad (10)$$

где $\nabla$ - градиент функции, $\Delta$ - дифференциальный оператор, $x, y$ – координаты, $f$ – функция дифференциального оператора.

Рисунок 1.6 – Фильтрация лапласиан с повышенной резкостью исходного изображения

ORB (Oriented FAST and Rotated BRIEF) является комбинацией из модифицированных алгоритмов нахождения особых точек с последующим определением их особенностей в виде бинарной строки. Изображение разбивается на ячейки P, которые имеют размеры SxS пикселей. Из ячейки выбираются пары пикселей $\{(X,Y), \forall\, X,Y \text{ в окрестности}\}$, для которых строится набор бинарных тестов [115]:

$$\tau(P,X,Y) = \begin{cases} 1, & I(X) < I(Y) \\ 0, & I(X) > I(Y) \end{cases} \quad (11)$$

где $I(X), I(Y)$ – интенсивность пикселей $X, Y$, $P$ – ячейки изображения, $\tau$ – преобразование в попарном сравнении пикселей.

Для каждой ячейки выбирается набор, содержащий $n_d$ пар точек, которые однозначно определяют набор бинарных тестов. Затем на основе этих тестов строится двоичная строка:

$$f_{n_d}(P) = \sum_{1 \le i \le n_d} 2^{i-1}\, \tau(P, X_i Y_i) \quad (12)$$

где $n_d$– пары точек, определяющие набор бинарных тестов.

### 1.3.1. Мел–частотные кепстральные коэффициенты

Мел–частотные кепстральные коэффициенты – это кепстральные коэффициенты, полученные на основе искаженной частотной шкалы, основанной на слуховом восприятии человека.

Создание MFCC начинается с разделения входящего звукового сигнала на короткие отрезки времени – фреймы, с заданным интервалом. Каждый из этих фреймов подвергается ряду преобразований [69], включающему:

1. Первоначальную обработку сигнала с использованием КИХ-фильтра (фильтра с конечной импульсной характеристикой) [70]:

$$y_t = x_t - b \times x_{t-1} \quad (13)$$

где $y_t$ – звуковой сигнал после фильтрации, $x_t$– входной звуковой сигнал, $t$ – количество кадров, $b$ – коэффициент фильтрации.

2. Дискретное преобразование Фурье (ДПФ) [76]:

$$F_k = \sum_{t=0}^{T-1} w_t * y_t e^{\frac{-2\pi i}{T} kt}, k = 0, \dots, \frac{T}{2} \quad (14)$$

где $T$ – отсчеты в кадре (четное значение); $T > 1$, $w_t$ – весовая оконная функция, $k$ – индекс частоты.

Использование весовой функции (оконная функция) необходимо для минимизации артефактов, возникающих при разделении сигнала на отдельные фреймы. Самые популярные типы оконных функций:

a) Окно Хэмминга [71]:

$$w_t^{hamm} = 0.54 - 0.46 \cos\left(\frac{2\pi t}{T-1}\right) \quad (15)$$

b) Окно Хана [71]:

$$w_t^{hann} = 0.5 - \left(1 - \cos\left(\frac{2\pi t}{T-1}\right)\right) \quad (16)$$

3. Логарифм спектральной энергии для группы треугольных фильтров, расположенных в диапазоне частот мел-шкалы, рассчитывается по формуле (Рисунок 1.7) [68]:

$$E_s^{Mel} = log\ (\sum_{k=0}^{\frac{T}{2}} |F_k| H_s^{Mel}(f_k))\ s = 0, \dots, M-1 \quad (17)$$

где $F_k$ – дискретное преобразование Фурье, $M$ – количество треугольных фильтров, $H_s^{Mel}$ – s–й треугольный фильтр в мел–частотном диапазоне, который определяется по формуле [80]:

$$H_s^{Mel}(f) = \begin{cases} 0\ ,\ m(f) < m_{begin}^s \\ \dfrac{m(f) - m_{begin}^s}{m_{center}^s - m_{begin}^s}, m_{begin}^s \le m(f) \le m_{center}^s \\ \dfrac{m_{end}^s - m(f)}{m_{end}^s - m_{center}^s}, m_{center}^s \le m(f) \le m_{end}^s \\ 0,\ m(f) \ge m_{end}^s \end{cases} \quad (18)$$

где $m_{begin}^s, m_{center}^s, m_{end}^s$ – начало, середина и конец треугольного окна s–го мел–частотного фильтра, $m(f)$ – показатели частоты в масштабе мел–шкалы.

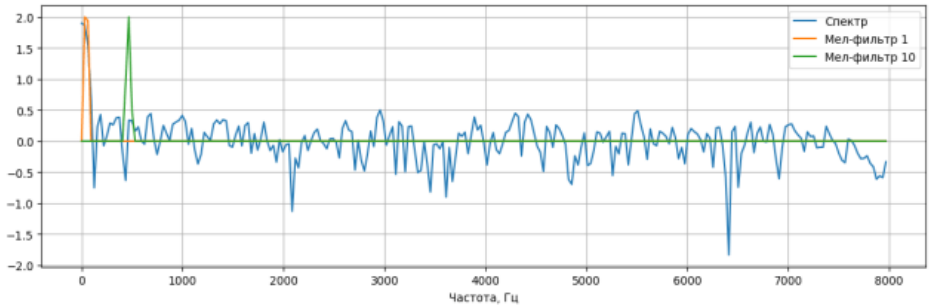


Рисунок 1.7 – Мел–фильтры с речевым спектром в частотной шкале

Данные значения вычисляются по формуле [81]:

$$f_k = \hat{f}\frac{k}{T} \tag{19}$$

$$m(f) = 1127\, ln\left(1 + \frac{f}{700}\right) \tag{20}$$

$$m^s_{begin} = m(f_{low}) + s \times \frac{m(f_{high}) - m(f_{low})}{M+1} \tag{21}$$

$$m^s_{end} = m(f_{low}) + (s+2) \times \frac{m(f_{high}) - m(f_{low})}{M+1} \tag{22}$$

$$m^s_{center} = \frac{1}{2}\left(m^s_{begin} + m^s_{end}\right) \tag{23}$$

где $\hat{f}$ – показатели частоты дискретизации входного звукового сигнала, $f_k, f_{high}, f_{low}$ – анализируемый частотный диапазон.

При применении линейной трансформации вместо мел-шкалы получаются линейно-частотные кепстральные коэффициенты (LFCC) [99].

4. Для полученных значений энергий фильтров $E_s$ применяется дискретное косинусное преобразование [94]:

$$C_l = \sum_{s=0}^{M-1} E_s\, cos\left(l\left(s + \frac{1}{2}\right)\frac{\pi}{M}\right), \quad l = 0, \ldots, M-1 \tag{24}$$

где $C_l$ – мел–кепстральный коэффициент под номером $l$, $M$ – количество треугольных фильтров, $s$ – номер порядка треугольного фильтра.

### 1.3.2. Коэффициенты линейного предсказания

Линейное предсказание P-го порядка предполагает определение текущего значения звукового сигнала на основе P предыдущих значений, что описывается следующей формулой [83]:

$$x_t = \sum_{p=1}^{P} \alpha_p x_{t-p} \tag{25}$$

где P – порядок, $\alpha_p$ – коэффициент линейного предсказания.

Разницей между фактическим значением сигнала и его прогнозируемым значением является ошибка прогнозирования [84]:

$$e_t = x_{t-l} - \hat{x}_{t-l} = x_{t-l} - \sum_{p=1}^{P} \alpha_p x_{t-p} \quad (26)$$

Для решения задачи линейного прогнозирования необходимо вычислить частные производные заданного уравнения по целевым переменным и приравнять их к нулю [91]:

$$\frac{\partial E}{\partial \alpha_p} = 0 \quad p = 1, \dots, P \quad (27)$$

По результатам выполнения ряда преобразований вычисление коэффициентов $\alpha_p$ сводится к решению системы линейных уравнений порядка P [93]:

$$\sum_{p=1}^{P} \alpha_p \varphi_{ip} = -\varphi_{i0} \quad i = 1, \dots, P \quad (28)$$

$$\varphi_{ij} = \sum_t x_{t-i} x_{t-j} \quad (29)$$

Автокорреляционный метод линейного прогнозирования направлен на минимизацию ошибки сигнала во всем временном интервале от $-\infty$ до $+\infty$ [97]:

$$E^{auto} = \sum_{t=0}^{T+P-1} {e_t}^2 = \sum_{t=0}^{T+P-1} \left( x_t - \sum_{p=1}^{P} \alpha_p x_{t-p} \right)^2 \quad (30)$$

И искомые уравнения примут вид [100]:

$$\sum_{p=1}^{P} \alpha_p \varphi_{ip}^{auto} = -\varphi_{i0}^{auto} \quad i = 1, \dots, P \quad (31)$$

$$\varphi_{ij}^{auto} = \sum_{t=0}^{T+P-1} x_{t-i} x_{t-j} = \sum_{t=0}^{T-1+(i-j)} x_t x_{t+(i-j)} \quad (32)$$

### 1.3.3. Перцепционные коэффициенты линейного предсказания

Перцепционные коэффициенты линейного предсказания – это коэффициенты, предназначенные для точного приближения звукового сигнала к параметрам восприятия человека.

В ходе обработки сигнал включает в себя следующую последовательность преобразований:

1. Спектральная энергия группы частотных фильтров по барк-шкале [105]:

$$E_s^{Bark} = \sum_{k=0}^{\frac{T}{2}} |F_k| H_s^{Bark}(f_k) \quad s = 0, \dots, M-1 \tag{33}$$

где $M$ – количество фильтров, $H_s^{Bark}$– s–й частотный фильтр в масштабе барк–шкалы, определяемый по формуле [106]:

$$H_s^{Bark}(f) = \begin{cases} 0\,, \ B(f) - B_{center}^s < -1.3 \\ 10^{2.5(B(f) - B_{center}^s + 0.5)}, \ -1.3 \le B(f) - B_{center}^s < -0.5 \\ 1, \ -0.5 \le B(f) - B_{center}^s < 0.5 \\ 0\,, \ B(f) - B_{center}^s \ge 2.5 \end{cases} \tag{34}$$

где $B_{center}^s$ – центр окна s–го фильтра, $B(f)$ – значение частоты в масштабе барк–шкалы. Эти значения можно рассчитать по формулам [110]:

$$B(f) = 6 \operatorname{arsh}\left(\frac{f}{600}\right) \tag{35}$$

$$B_{begin}^s = B(f_{low}) + s \times \frac{B(f_{high}) - B(f_{low})}{M+1} \tag{36}$$

$$B_{end}^s = B(f_{low}) + (s+2) \times \frac{B(f_{high}) - B(f_{low})}{M+1} \tag{37}$$

$$B_{center}^s = \frac{1}{2}\left(B_{begin}^s + B_{end}^s\right) \tag{38}$$

где $f_{high}$, $f_{low}$ – анализируемый частотный диапазон.

2. Применение степенного закона для определения интенсивности восприятия [111]:

$$\Phi_s^{Bark} = (E_s^{Bark})^{0.33} \tag{39}$$

3. Подсчет коэффициентов линейного предсказания.

### 1.3.4. Признаки частоты спектрального центроида

Частота спектрального центроида (Spectral centroid frequency, SCF) представляет собой средневзвешенную частоту для данного поддиапазона, где весовые коэффициенты представляют собой нормированную энергию каждого частотного компонента в этом поддиапазоне [52], [115]:

$$F_k = \frac{\sum_{f=l_k}^{u_k} f|S[f]\omega_k[f]|}{\sum_{f=l_k}^{u_k} |S[f]\omega_k[f]|}, k = 0, \dots, \frac{T}{2} \tag{40}$$

где $u, l$ – граничные частоты поддиапазона, $S[f]$–спектр фрейма, состоящего из $k$ поддиапазонов, $\omega_k$–отклик фильтра частоты.

Амплитуда спектрального центроида является средневзвешенным значением амплитуды для того поддиапазона, где весовые коэффициенты – это частоты каждого компонента амплитуды в этом поддиапазоне, вычисленные с помощью формулы [84]:

$$M_k = \frac{\sum_{f=l_k}^{u_k} f|S[f]\omega_k[f]|}{\sum_{f=l_k}^{u_k} f} \tag{41}$$

### 1.3.5. Q–константные кепстральные коэффициенты

Уникальной особенностью данного метода является его альтернатива преобразованию Фурье, которое преобразует сигнал из временной области в частотную, улучшая спектральное разрешение [107]:

$$X^{CQ}(k,n) = \sum_{j=n-\frac{N_k}{2}}^{n+\frac{N_k}{2}} x(j) a_k^* \left(j - N + \frac{N_k}{2}\right) \tag{42}$$

где $k = 1,2,\dots,K$ – индекс частотного интервала, $N_k$– вариативная длина окна, $n$ – длина окна временного сигнала, $a_k^*$– комплексно–сопряженное базисных функций $a_k(n)$.

Базисные функции $a_k(n)$ рассчитываются по формуле [107]:

$$a_k(n) = \frac{1}{C}\left(\frac{n}{N_k}\right) \exp\left[i\left(2\pi n \frac{f_k}{f_s} + \Phi_k\right)\right] \tag{43}$$

где $f_k$– центральная частота частотного интервала $k$, $f_s$– частота дискретизации, $\Phi_k$– сдвиг фазы, $C$– коэффициент масштабирования, $w(t)$ – оконная функция, $N_k$ – окно.

Длина окна $N_k$ зависит от Q–фактора:

$$Q = \frac{f_k}{f_{k-1} - f_k} \tag{44}$$

$$N_k = \frac{f_k}{f_s} Q \tag{45}$$

## 1.4. Цель и задачи исследования

На данный момент актуальна задача разработки биометрических систем аутентификации пользователей, которая необходима для повышения точности распознавания пользователей и снижения показателей ошибки первого рода, ошибки второго рода. При этом в биометрической системе аутентификации рекомендуется применять современные математические методы и методы машинного обучения.

Таким образом, целью диссертационной работы является повышение эффективности процедуры обучения нейронных сетей на основе биометрической аутентификации пользователей информационной системы.

Для достижения данной цели необходимо поставить и решить следующие задачи исследования:

1. Разработать модели биометрической аутентификации пользователей информационной системы по изображению лица с применением фильтра Калмана, по голосу с применением метода шумоочистки и представления спектров речевых сигналов в составе мультимодальной биометрической системы.
2. Разработать принципы, решения и алгоритмы предварительной обработки изображения лица и голоса, обучения сверточной нейронной сети в составе мультимодальной системы биометрической аутентификации.
3. Разработать мультимодальную систему биометрической аутентификации пользователей информационной системы на основе нейронных сетей, программный комплекс данной системы.
4. Осуществить оценку эффективности проведенных экспериментальных работ и полученных результатов.

## Выводы по первой главе

Проведен обзор актуальных нормативно-правовых актов по биометрической аутентификации пользователей, литературный обзор библиографических источников по теме диссертационного исследования.

Рассмотрен принцип работы современных мультимодальных и мультибиометрических систем аутентификации. Проанализированы уровни принятия решения биометрических систем аутентификации, метод нормализации степеней схожести каждого биометрического процесса в общей области. Представлена структура объединения в биометрической системе аутентификации признаков на уровне степеней схожести. Приведено описание используемого математического аппарата в виде теории вероятностей, теории принятия решений и математической статистики для интерпретации выполняемых процессов в рамках биометрической системы аутентификации.

# ГЛАВА 2. РАЗРАБОТКА И ИССЛЕДОВАНИЕ МОДЕЛИ РАСПОЗНАВАНИЯ ПОЛЬЗОВАТЕЛЕЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО ИЗОБРАЖЕНИЮ ЛИЦА

В данном разделе приведены основные алгоритмы распознавания пользователей информационных систем по изображению лица, проведено сравнение рассматриваемых алгоритмов. Осуществлена предварительная обработка и морфологическое преобразование изображений лица. Описаны процессы извлечения признаков направленных градиентов HOG и локальных бинарных шаблонов LBP. Представлены гистограммы направлений и величин градиентов, значимых (доминантных) локальных бинарных шаблонов изображений лиц пользователей информационной системы, а также иллюстрации с применением морфологического преобразования. Рассмотрены основные аспекты видеоаналитики, алгоритм обработки видеосигнала. Разработана архитектура искусственной нейронной сети распознавания пользователей информационных систем по изображению лица. На основе сформированного Dataset изображений пользователей информационной системы проведено обучение искусственной нейронной сети. На иллюстрациях изображены основные результаты работы модели распознавания пользователей информационных систем на видеозаписи в режиме реального времени. Приведены основные показатели разработанной модели распознавания пользователей информационных систем по изображению лица на основе построенного многослойного персептрона.

В данном разделе приведены основные методы фильтрации изображений, включая метод Калмана. Проведено исследование на основе адаптивной фильтрации Калмана в процессе обработки изображений лица пользователей информационной системы.  В рамках поставленной задачи разработан программный модуль адаптивной фильтрации изображений лица, взятых с видеопотока в режиме реального времени. Приведены основные показатели разработанного модули, которые отражены в виде ошибки первого

рода и ошибки второго рода при распознавании пользователей информационной системы по изображению лица.

## 2.1. Предобработка и морфологическое преобразование изображений с целью формирования биометрического образа пользователя по лицу

Для проведения анализа цифровых изображений часто требуется обработка с целью повышения информативности и качества. Существуют специальные операции для работы с изображениями: получение негатива, бинаризация (преобразование снимка в черно–белые цвета), конвертирование в серый.

Для редактирования цифровых изображений существуют различные алгоритмы обработки изображений, реализованные в современных программах. Их применение позволяет получить высокое качество изображения.

Существуют такие методы обработки изображений, как:

– геометрические (поворот, масштаб, обрезание);

– морфологические (дилатация, эрозия);

– преобразования цветных изображений (негатив, гамма, сглаживание);

– изображений в градациях серого (преобразование Лапласа, пороговое, нахождение границ);

– операции по работе с измерениями (поиск контуров) [37].

Морфология в контексте анализа и обработки изображений описывает свойства формы и структуры его областей. То есть морфологические преобразования неразрывно связаны с анализом изображений на основе формы.

Морфологические преобразования осуществляются с помощью фильтров:

– эрозия;

– дилатация;

– размыкание;

– замыкание;

– градиент;

– преобразование Top Hat;

– преобразование Black Hat [37].

Данные преобразования производятся с помощью структурообразующего элемента (примитива), наложение которого на изображение решает определенные задачи, и основаны на операциях над множествами. Множествами в морфологии обозначены объекты на изображении (точки в двумерном и трехмерном пространствах). Например, все черные пиксели черно–белого изображения являются одним из способов его морфологического описания.

Размыкание применяется для подсчёта участков на бинарном (черно–белом) изображении. После порогового преобразования какого–либо изображения с помощью данного фильтра можно подсчитать количество частиц, расположенных ближе друг к другу перед подсчётом участков.

Для вычисления индивидуальных характеристик пользователей информационной системы произведена фильтрация всех изображений (Рисунок 2.1) [51].

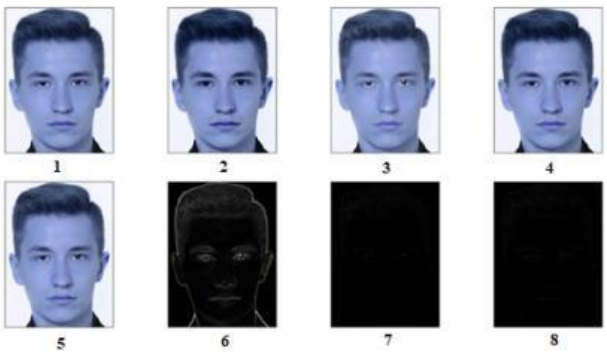


Рисунок 2.1 – Применение морфологических преобразований с помощью фильтров на примере исходного изображения (1 – исходное изображение, 2 – эрозия, 3 – дилатация, 4 – размыкание, 5 – замыкание, 6 – градиент, 7 – преобразование Top Hat, 8 - преобразование Black Hat)

## 2.2. Обоснование преимущества использования нейронных сетей для решения задачи распознавания пользователей информационной системы

При наличии входных данных можно вычислить значения выходных данных, подставив входное значение в функцию активации. Немаловажно подобрать конфигурацию весов нейронной сети. Корректная конфигурация весов нейронной сети поможет достичь высоких результатов в процессе обучения (Рисунок 2.2).

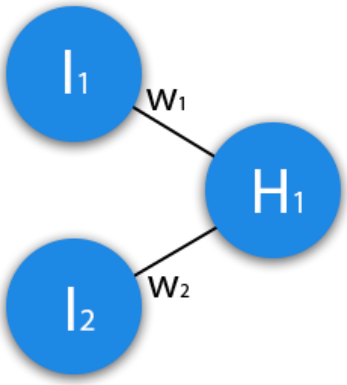


Рисунок 2.2 – Фрагмент нейронной сети с конфигурацией весовых коэффициентов

$$H_{1_{\text{вход}}} = (l_1 \times w_1) + (l_2 \times w_2) \quad (46)$$

где $H_{1_{\text{вход}}}$ – входное значение нейрона, $l_1$ и $l_2$ – входные нейроны, $w_1$ и $w_2$ – весовые коэффициенты.

$$H_{1_{\text{выход}}} = f_{\text{активации}}\left(H_{1_{\text{вход}}}\right) \quad (47)$$

где $H_{1_{\text{вход}}}$ – значение выходного нейрона, $f_{\text{активации}}$ – функция активации.

Так как значения извлечения коэффициентов LBP и признаков HOG больше или равны 0, то для обучения используется логистическая функция активации $f(x) = \frac{1}{1+e^{-x}}$.

Ошибка нейронной сети – это процентная величина, отражающая расхождение между ожидаемым и полученным результатами. Ошибку можно вычислить следующими способами:

– Среднеквадратическая ошибка (Mean Squared Error, MSE) [39]:

$$\frac{(i_1 - a_1)^2 + (i_2 - a_2)^2 + \cdots + (i_n - a_n)^2}{n} \quad (48)$$

где $i$ – фактическое значение, $a$ – значение ожидаемого результата, $n$ – номер нейрона.

– Среднеквадратичная ошибка (Root Mean Squared Error, RMSE) [39]:

$$\sqrt{\frac{(i_1 - a_1)^2 + (i_2 - a_2)^2 + \cdots + (i_n - a_n)^2}{n}} \quad (49)$$

– Arctan [39]:

$$\frac{arctan^2(i_1 - a_1) + arctan^2(i_2 - a_2) + \cdots + arctan^2(i_n - a_n)}{n} \quad (50)$$

## 2.3. Выбор архитектуры нейронной сети

В рамках данного исследования была создана архитектура сверточной нейронной сети (СНС) для репрезентативного набора обучающей выборки, которая позволяет достичь высоких показателей обучения нейронной сети в процессе распознавания личности по изображению лица.

После извлечения LBP и признаков HOG составляется структура СНС, которая состоит из 4 сверточных слоев, одного полносвязного слоя с применением функции активации (Рисунок 2.3) [19].

В ходе исследования составлена блок-схема алгоритма обучения указанной сверточной нейронной сети, отличающаяся извлечением дополнительных дескрипторов изображения, что позволяет минимизировать возможность синтеза изображений (дипфейк) (Рисунок 2.4).


Рисунок 2.3 – Архитектура сверточной нейронной сети распознавания пользователей информационных систем по изображению лица

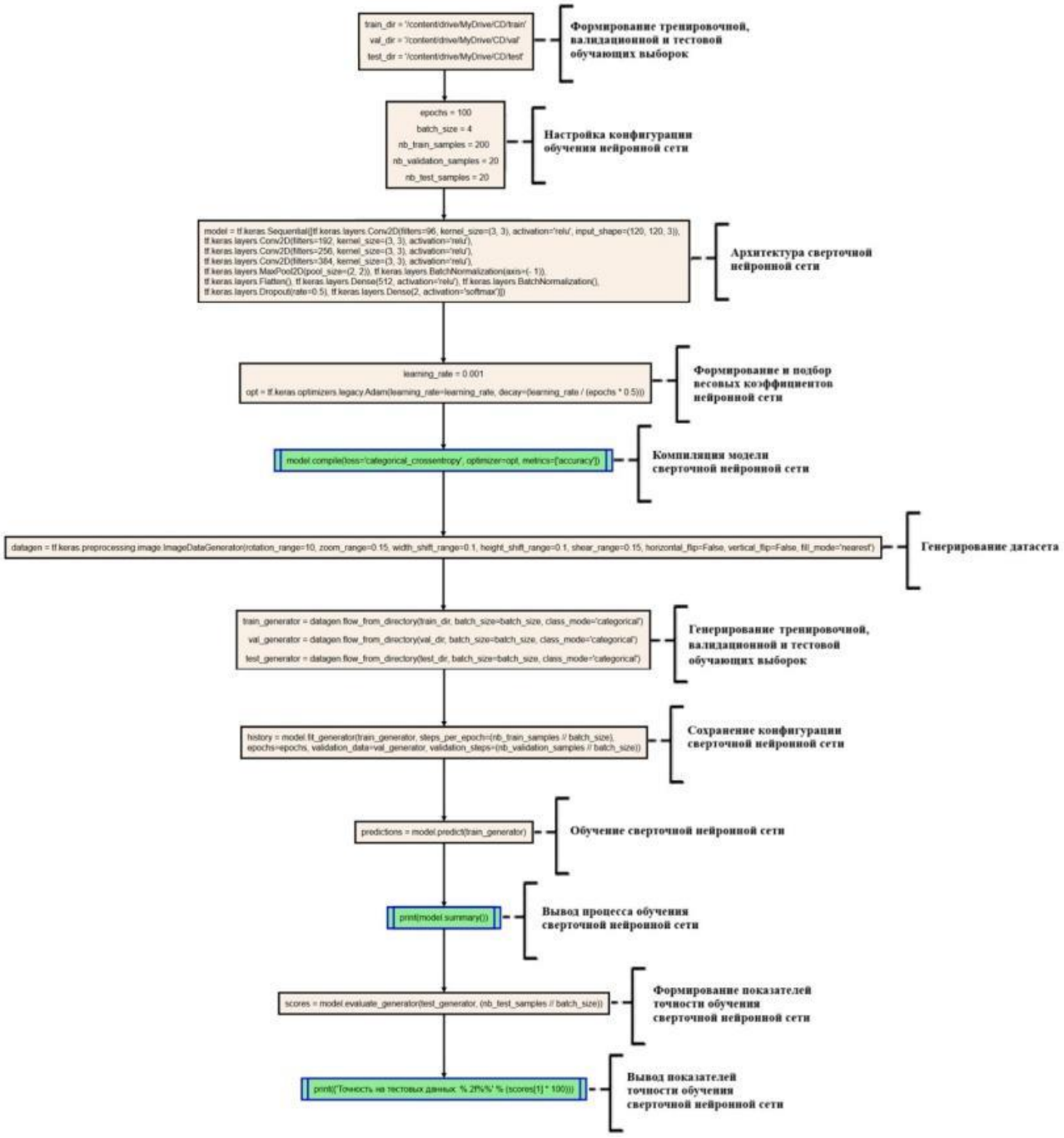


Рисунок 2.4 – Блок-схема программной реализации алгоритма сверточной нейронной сети по изображению лица (на основе pyflowchart)

В рамках проведенного исследования были проведены экспериментальные работы с различными архитектурами нейронных сетей и различными наборами обучающих выборок (Таблица 2.1).

Таблица 2.1 – Используемые обучающие выборки при распознавании пользователей информационной системы по изображению лица

| Наименование обучающей выборки | Количество пользователей информационной системы | Объем обучающей выборки |
|---|---|---|
| CelebFaces | 100 | 10 000 |
| Yalefaces | 15 | 8 775 |
| Собственный датасет (DataSet 600) | 2 | 600 |

Проведен сравнительный анализ полученных результатов после обучения нейронной сети с разными наборами обучающих выборок (Таблица 2.2, Таблица 2.3, Таблица 2.4).

Таблица 2.2 – Сравнительные результаты проведенных экспериментов с альтернативными архитектурами нейронных сетей на выбранном наборе обучающей выборки при аутентификации пользователей по изображению лица

| **Параметры** | **Сверточная нейронная сеть (DataSet 600)** | **Сверточная нейронная сеть ResNet (DataSet 600)** | **Сверточная нейронная сеть AlexNet (DataSet 600)** | **Сверточная нейронная сеть FeatherNet (DataSet 600)** | **Сверточная нейронная сеть YOLO11 (DataSet 600)** |
|---|---|---|---|---|---|
| Показатели функции потерь | 0,043 | 0,04 | 0,126 | 0,043 | 0,04 |
| Точность (accuracy), % | 98,14 % | 97,82 % | 83,45 % | 97,78 % | 96,67 % |
| Количество нейронов по слоям | 96, 192, 256, 384, 512 | 256, 1382, 512 | 256, 1382, 512 | 32, 32, 16, 32, 64, 96, 40, 11, 5, 1, 1 | 64, 64, 192, 128, 256, 256, 512, 256, 512, 512, 1024, 512, 1024, 1024, 1024, 1024 |
| Функции активации нейронов по слоям | relu, softmax | linear, relu, softmax | relu, softmax | hardswish, sigmoid | relu, leaky relu |
| Dropout (Регуляризация) | Да | Да | Да | Да | Да |

Таблица 2.3 – Сравнительные результаты проведенных экспериментов с альтернативными архитектурами нейронных сетей на наборе обучающей выборки CelebFaces при аутентификации пользователей по изображению лица

| **Параметры** | **Сверточная нейронная сеть (CelebFaces)** | **Сверточная нейронная сеть ResNet (CelebFaces)** | **Сверточная нейронная сеть AlexNet (CelebFaces)** | **Сверточная нейронная сеть FeatherNet (CelebFaces)** | **Сверточная нейронная сеть YOLO11 (CelebFaces)** |
|---|---|---|---|---|---|
| Показатели функции потерь | 0,04 | 0,041 | 0,098 | 0,038 | 0,037 |
| Точность (accuracy), % | 97,91 % | 97,56 % | 85,55 % | 97,2 % | 96,51 % |
| Количество нейронов по слоям | 96, 192, 256, 384, 512 | 256, 1382, 512 | 256, 1382, 512 | 32, 32, 16, 32, 64, 96, 40, 11, 5, 1, 1 | 64, 64, 192, 128, 256, 256, 512, 256, 512, 512, 1024, 512, 1024, 1024, 1024, 1024 |
| Функции активации нейронов по слоям | relu, softmax | linear, relu, softmax | relu, softmax | hardswish, sigmoid | relu, leaky relu |
| Dropout (Регуляризация) | Да | Да | Да | Да | Да |

Таблица 2.4 – Сравнительные результаты проведенных экспериментов с альтернативными архитектурами нейронных сетей на наборе обучающей выборки Yalefaces при аутентификации пользователей по изображению лица

| **Параметры** | **Сверточная нейронная сеть (Yalefaces)** | **Сверточная нейронная сеть ResNet (Yalefaces)** | **Сверточная нейронная сеть AlexNet (Yalefaces)** | **Сверточная нейронная сеть FeatherNet (Yalefaces)** | **Сверточная нейронная сеть YOLO11 (Yalefaces)** |
|---|---|---|---|---|---|
| Показатели функции потерь | 0,044 | 0,048 | 0,072 | 0,045 | 0,039 |
| Точность (accuracy), % | 96,33 % | 95,49 % | 87,78 % | 95,93 % | 96,21 % |
| Количество нейронов по слоям | 96, 192, 256, 384, 512 | 256, 1382, 512 | 256, 1382, 512 | 32, 32, 16, 32, 64, 96, 40, 11, 5, 1, 1 | 64, 64, 192, 128, 256, 256, 512, 256, 512, 512, 1024, 512, 1024, 1024, 1024, 1024 |
| Функции активации | relu, softmax | linear, relu, softmax | relu, softmax | hardswish, sigmoid | relu, leaky relu |

| | | | | | |
|---|---|---|---|---|---|
| нейронов по слоям | | | | | |
| Dropout (Регуляризация) | Да | Да | Да | Да | Да |

## 2.4. Обучение нейронной сети: формирование обучающей выборки для поиска и распознавания пользователей информационной системы на видео в режиме реального времени

Помимо поставленной задачи следует изучить не менее актуальную проблему распознавания искомого пользователя информационной системы. Для осуществления несанкционированного доступа используется комплекс нелигитимных программных решений [67]. В том числе данные решения применяются в процессах аутентификации пользователей и олицетворяют собой методику синтеза изображения (дипфейк). Данная методика используется для соединения и наложения существующих изображений и видео на исходные изображения или видеоролики (например, FakeApp) [26]. Несмотря на то, что данные инструменты используются в других сферах деятельности (в том числе в качестве развлекательного контента), количество киберинцидентов продолжает расти. Не каждое современное программное решение, предназначенное для защиты биометрических персональных данных на устройстве, может справиться с действиями злоумышленников. Был разработан модуль, предназначенный для противодействия методике синтеза изображения и для противодействия распознавания пользователей информационной системы по изображениям, распечатанным на бумажном носителе и сохраненных на других устройствах [20].

Для корректного распознавания пользователей информационной системы была сформирована база данных изображений, состоящая из следующих папок:

- Анфас.
- Профиль слева.
- Профиль справа.

– Глаза открыты.

– Глаза закрыты.

Dataset обучающей выборки разделен на три папки: test (тестовый набор), train (тренировочный набор) и val (валидационный набор). Каждая из этих папок разбивается на два класса. Под классом подразумевается пользователь информационной системы. Папка каждого класса (пользователя информационной системы) содержит 100 изображений. Так как два класса содержатся в каждой папке обучающей выборки, общий объем базы данных состоит из 600 биометрических образов (изображений) [48].

Согласно приведенным результатам исследований ученых [16, 22-23, 75, 102], любую непрерывную функцию нескольких переменных можно с любой степенью точности реализовать с помощью трехслойного (считая входной слой) персептрона, имеющего достаточное количество нейронов в скрытом слое. Таким образом, общее количество коэффициентов, представляющих собой веса синаптических связей, соответствует минимальному числу нейронов в скрытом слое в соответствии с нижней границей сложности нейронных сетей [75]:

$$N = \left[\frac{Q * n}{m + n}\right] \quad (51)$$

Выходной слой состоит из результатов, который определяет авторизованных пользователей информационной системы (АП) и неавторизованных пользователей (НАП).

В дальнейшем при распознавании авторизованных и неавторизованных пользователей информационной системы сохраняется снимок изображения, созданный в режиме реального времени [27]. После данной процедуры актуальное изображение пройдет предварительную обработку и морфологическое преобразование. В качестве меры различия гистограмм и, соответственно, в качестве погрешности использовано расстояние Кульбака–Лейблера, предназначенное для распознавания пользователей информационной системы [57]:

$$D_{KL}(f,g) = \sum_{m=1}^{P\,(P-1)+3} f_m \, ln \frac{f_m}{g_m} \quad (52)$$

где $f$ и $g$ – гистограммы изображений, $P$ – число точек в окрестности LBP, $m$ – номер столбца.

Для корректности оценки эффективности используемой методики извлечения характеристик изображений лица пользователей информационной системы приведены график зависимостей ошибки первого рода от ошибки второго рода в мультимодальной системе биометрической аутентификации пользователей информационной системы (Рисунок 2.5), ROC-кривая полученных показателей обучения нейронных сетей мультимодальной системы биометрической аутентификации пользователей информационной системы (Рисунок 2.6) и матрица ошибок мультимодальной системы биометрической аутентификации пользователей информационной системы (Рисунок 2.7).

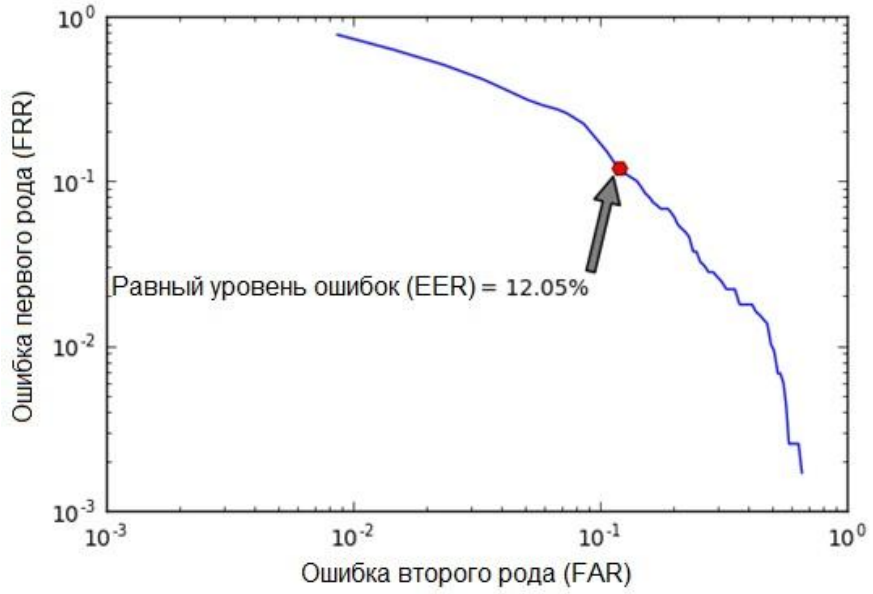


Рисунок 2.5 – График зависимостей ошибки первого рода от ошибки второго рода в мультимодальной системе биометрической аутентификации пользователей информационной системы

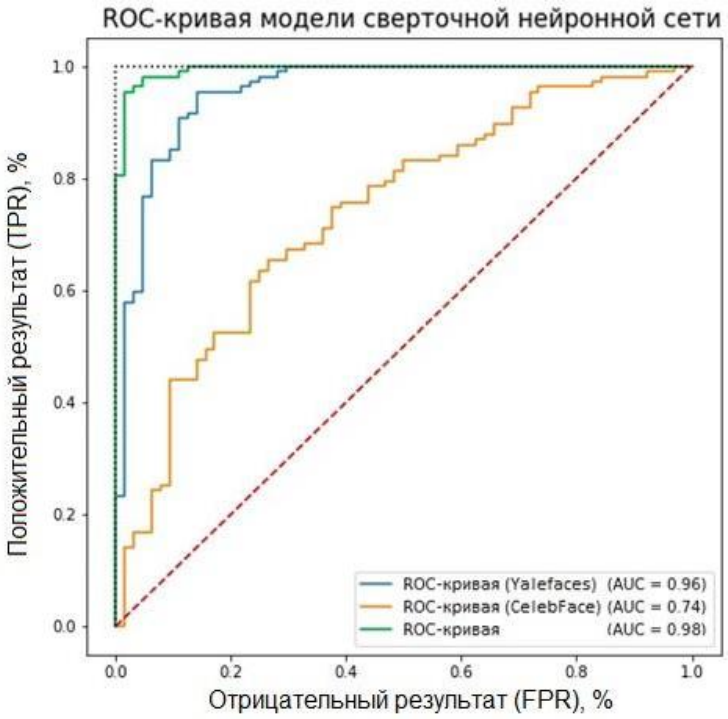


Рисунок 2.6 – ROC-кривая полученных показателей обучения нейронных сетей мультимодальной системы биометрической аутентификации пользователей информационной системы

| n= 100 | Принадлежит классу: Пользователь | Не принадлежит классу: Не авторизован | |
|---|---|---|---|
| Предсказана принадлежность классу: Пользователь | TN = 50 | FP = 5 | 55 |
| Предсказано отсутствие принадлежности к классу: Не авторизован | FN = 4 | TP = 41 | 45 |
| | 54 | 46 | |

Рисунок 2.7 – Матрица ошибок мультимодальной системы биометрической аутентификации пользователей информационной системы

## 2.5. Преимущества использования фильтра Калмана для повышения точности распознавания пользователей с помощью нейронной сети

Для разграничения доступа в информационной системе применяются процессы идентификации и аутентификации на основе биометрических признаков. Основными показателями распознавания пользователей информационной системы являются качество и точность проведенной процедуры. В процессе распознавания пользователей по изображению лица зачастую обнаруживаются цифровые шумы. Для коррекции изображений в системе биометрической аутентификации следует применять морфологические преобразования и фильтрацию. Алгоритмы фильтрации изображения на основе фильтра Калмана ранее были представлены в трудах А. А. Сироты, А. Ю. Иванкова [73], М. В. Куликовой, Ю. В. Цыгановой, А. А. Афанасьева с описанием обработки сигналов на основе обратного адаптивного моделирования и оценкой градиента.

В рамках проводимого исследования был применен алгоритм адаптивной фильтрации Калмана при распознавании пользователей информационной системы по изображению лица. Изображения лица пользователей информационной системы являются частью формируемой обучающей выборки, которая сохранена в целевой папке. Общий объем обучающей выборки составляет 900 изображений.

Оценка изображения $X$ (аддитивная смесь информативного сигнала $X = \{x_{ij}\}$) проводится по наблюдениям $Z$ (наблюдаемое изображение $Z = \{z_{ij}\}$), где оценки $X$ находятся построчно. Первая строка $x_1 = \{x_{1j} : j = \overline{1,N}\}$ оценивается по первой строке наблюдений $z_1 = \{z_{1,j} : j = \overline{1,N}\}$ с помощью уравнения фильтра Калмана [77]:

$$\tilde{x}_{1j} = a\tilde{x}_{1,j-1} + b\left(z_{1j} - a\tilde{x}_{1,j-1}\right) = c\tilde{x}_{1,j-1} + bz_{1j} \quad (53)$$

$$c = a(1-b) \quad (54)$$

где $a, b, c$ – коэффициенты фильтра Калмана, $\tilde{x}_{1j}$ – оценка изображений по текущим наблюдениям.

Далее производится сглаживание полученных показателей $\tilde{x}_{1j}$ обратным ходом [62]:

$$\hat{x}_{1j} = \tilde{x}_{1j} + b\left(\hat{x}_{1,j+1} - a\tilde{x}_{1j}\right) \quad (55)$$

После оценки $\hat{x}_{i-1}$ строки с номером $i-1$ следующая $i-$ая строка представляется в виде [62]:

$$x_i = (x_i - rx_{i-1}) + rx_{i-1} = y_i + rx_{i-1} \quad (56)$$

где $r = r_{ij}$ – постоянный параметр модели $z_{ij} = x_{ij} + \theta_{ij}$ – коэффициент корреляции между соседними строками, $\theta_{ij}$ – параметр модели наблюдения, $y_i$ – параметр оценки на основе наблюдений.

Оценка

$$\hat{x}_i = \hat{y}_i + r\hat{x}_{i-1} \quad (57)$$

находится по наблюдениям

$$z_i' = z_i - r\hat{x}_{i-1} \quad (58)$$

которые были получены вычитанием прогноза $r\hat{x}_{i-1}$строки $x_i$ из наблюдений $z_i$ этой строки.

Для минимизации остатков применяется алгоритм адаптации. Адаптивный вариант фильтра Калмана состоит из процессов с переменными коэффициентами $a, b, c, r$ и подстройки этих параметров в процессе обработки.

В адаптивном фильтре Калмана формирование прогнозов $x^*{}_{ij} = r\hat{x}_{i-1,j}$ элементов $x_{ij}$ осуществляется по уже полученным оценкам $\hat{x}_{i-1,j}$ предыдущей строки. Прогнозы, определяемые рекуррентным алгоритмом коррекции (РАК), предназначены для минимизации дисперсии ошибок прогноза $s_{ij} = x_{ij} - x^*{}_{ij}$. Наблюдения $z_{ij} = x_{ij} + \theta_{ij}$ отличаются от $x_{ij}$ некоррелированным с $x_{ij}$ цифровым шумом $\theta_{ij}$, поэтому оптимальный прогноз $rx_{i-1,j}$ минимизирует не только дисперсию остатков $s_{ij}$, но и дисперсию остатков прогноза наблюдений $z_{ij}$. Также применяется метод градиентного спуска при подстройке коэффициента прогноза $r$. Реализация адаптивного фильтра Калмана основана на том, что если вектор параметров $\bar{\alpha} = (a, b)^T$

минимизирует средние квадраты ошибок оценок $x_{ij}$, то он также минимизирует средние квадраты ошибок прогнозов [77]:

$$\Delta_{ij} = z_{ij} - a\tilde{x}_{i,j-1} \quad (59)$$

и наоборот. Подстройка $\bar{\alpha}$ может быть осуществлена по наблюдаемым $\Delta_{ij}$. Остатки при минимизации средних квадратов ошибок прогнозов зависят от параметра $b$ через $\hat{x}_{i,j-1}$[77]:

$$\Delta_{ij+1} = z_{i,j+1} - a^2\tilde{x}_{i,j-1} - ab\Delta_{ij} \quad (60)$$

Рассчитанные на очередном шаге коэффициенты $a_{i,j+1}$ и $b_{i,j+1}$ используются для вычисления очередного прогноза $\tilde{x}_{i,j+1}$ и уточнения значения $\Delta_{i,j+1}$.

В основе теории оптимальной фильтрации Калмана лежит понятие пространства состояний. Поэтому метод Калмана называют также методом пространства состояний [63]. Данный метод предназначен для шумоподавления цифрового шума исходных изображений пользователей информационной системы.

Пусть спектральная плотность задающего воздействия имеет вид:

$$S_g(\omega) = \frac{!S_0}{\alpha_n\omega^{2n} + \alpha_{n-1}\omega^{2n-2} + \cdots + \alpha_1\omega^2 + 1} \quad (61)$$

где $S_g(\omega)$ – спектральная плотность, $\omega$ – частота, определяющая ширину спектра, $\alpha$ – коэффициент.

Произведя факторизацию спектральной плотности, найдем передаточную функцию $\Psi_g(\omega)$ формирующего фильтра для задающего воздействия.

Пусть $S_g(\omega) = \Psi_g(j\omega)\Psi_g(-j\omega)$, тогда передаточная функция имеет вид [63]:

$$\Psi_g(j\omega) = \frac{1}{\gamma_n(j\omega)^n + \gamma_{n-1}(j\omega)^{n-1} + \cdots + \gamma_1 j\omega + 1} \quad (62)$$

где $\Psi_g(j\omega)$ - передаточная функция формирующего фильтра, $\gamma_k$, $k = \overline{1,\ n}$, $\alpha_i$, $i = \overline{1,\ n}$ - коэффициенты.

Следует обозначить через $u(t)$ порождающий белый шум, из которого фильтр с передаточной функцией формирует случайное задающее воздействие со спектральной плотностью. Тогда [63]:

$$g(t) = \Psi_g(p)u(t) = \frac{b_0}{p^n + a_{n-1}p^{n-1} + \cdots + a_1 p^1 + a_0}u(t) \qquad (63)$$

где $g(t)$ – случайное задающее воздействие, $u(t)$ – порождающий белый шум.

Далее стохастическое дифференциальное уравнение для случайного задающего воздействия со спектральной плотностью будет иметь вид [63]:

$$\frac{d^n g(t)}{dt^n} + a_{n-1}\frac{d^{n-1}g(t)}{dt^{n-1}} + \cdots a_1 \frac{dg(t)}{dt} + a_0 g(t) = b_0 u(t) \qquad (64)$$

где $b_0 u(t)$ - стохастическое дифференциальное уравнение для случайного задающего воздействия со спектральной плотностью.

Также представлено уравнение состояния [63]:

$$x_k(t) = \frac{d^{k-1}g}{dt^{k-1}}, k = \overline{1,\ n} \qquad (65)$$

Тогда:

$$\dot{x_k}(t) = \frac{d^k g}{dt^k} = x_{k+1}(t), k = \overline{1,\ n-1} \qquad (66)$$

Используя векторно-матричные обозначения для совокупностей переменных и коэффициентов, уравнение состояния будет представлено в виде одного матричного уравнения первого порядка [63]:

$$\frac{dx}{dt} = Ax(t) + Bu(t) \qquad (67)$$

где $x(t)$ - n-мерное задающее воздействие.

Корреляционная матрица порождающего белого шума $u(t)$ имеет вид [63]:

$$K_{uu}(t, \xi) = u(t)u'(\xi) = Q(t)\delta(t - \xi) \qquad (68)$$

где $Q(t)$ — матрица, элементы которой имеют размерность и смысл спектральных плотностей $q_{ii}$, взаимных спектральных плотностей $q_{ij}$

составляющих $u_i(t)$ и $u_j(t)$ процесса $u(t)$, $\delta$ — дельта функция, $\xi$ – случайная величина.

На вход синтезируемого фильтра поступает m-мерная совокупность наблюдаемых величин, имеющая смысл многомерного входного воздействия [63]:

$$r(t) = C(t)x(t) + v(t) \tag{69}$$

где $C(t)$ — матрица наблюдений, $v(t)$ — случайный m-мерный процесс типа белого шума.

Выражение для критерия оптимума многомерного фильтра Калмана аналогично формуле и имеет вид:

$$I_k = min\tilde{e}_k^2 \tag{70}$$

где

$$\tilde{e}_k^2 = \overline{[x_k(t) - y_k(t)]^2} \tag{71}$$

Как доказывается в теории оптимальной фильтрации, оптимальная оценка $y(t)$ процесса $x(t)$ удовлетворяет матричному дифференциальному уравнению (уравнение оценки) [63]:

$$\frac{dy}{dt} = A(t)y(t) + K(t)[r(t) - C(t)y(t)] \tag{72}$$

где

$$K(t) = P(t)C'(t)R^{-1}(t) \tag{73}$$

матричный коэффициент передачи фильтра Калмана.

$$P(t) = \overline{[x(t) - y(t)][x(t) - x(t)]'} \tag{74}$$

дисперсионная матрица ошибок фильтра.

Элементами последней матрицы являются величины [63]:

$$D_{ij}(t) = \overline{[x_i(t) - y_i(t)]\left[x_j(t) - y_j(t)\right]} \tag{75}$$

дисперсия ошибок оптимального фильтра.

Дисперсионная матрица ошибок $P(t)$ определяется путем решения дисперсионного уравнения [63]:

$$\frac{dP}{dt} = AP + PA' - PC'R^{-1}CP + BQB' \tag{76}$$

Это матричное нелинейное дифференциальное уравнение Риккати. Для его решения необходимо задать начальное значение $P(t_0)$ дисперсионной матрицы. Если в начальный момент времени $t = t_0$ процесс $y(t_0)$ на выходе фильтра равен нулю.

В процессе распознавания пользователей по изображению лица зачастую обнаруживаются цифровые шумы. Для коррекции изображений в системе биометрической аутентификации следует применять морфологические преобразования и фильтрацию. В рамках проводимого исследования был выбран и применен алгоритм адаптивной фильтрации Калмана при распознавании пользователей информационной системы по изображению лица (Таблица 2.5).

Таблица 2.5 – Эффективность работы фильтров при обработке изображений

| **Фильтры** | **Критерии оценки эффективности фильтров** | | |
|---|---|---|---|
| **Тип** | **Критерий среднеквадратичной ошибки MSE** | **Пиковое отношение сигнал-шум PSNR, dB** | **Универсальный индекс качества UQI** |
| Фильтр Винера-Хопфа | 0,0052 | 22,8632 | 0,9610 |
| Медианный фильтр | 0,0033 | 24,7971 | 0,9774 |
| Фильтр Фроста | 0,0021 | 26,7966 | 0,9856 |
| Адаптивный фильтр Калмана | **0,0019** | **27,3862** | **0,9921** |

## 2.6. Модель распознавания пользователей информационной системы по изображению лица

Перед разработкой модели распознавания пользователей информационной системы по изображению лица необходимо рассмотреть алгоритм извлекаемого вектора признаков, на основании которого будет осуществлен процесс аутентификации (Рисунок 2.8). Разработанная модель распознавания пользователей информационной системы по изображению лица включает в себя такие этапы, как (Рисунок 2.9):

1) Обработка входных данных.

2) Применение морфологического преобразования и предварительной обработки изображений.

3) Извлечение признаков изображения лица и формирование вектора признаков.

4) Составление архитектуры сверточной нейронной сети.

5) Формирование обучающей выборки и применение сверточной нейронной сети.

6) Применение фильтра Калмана для повышения точности распознавания пользователей по изображению лица.

7) Результаты распознавания пользователей по изображению лица.

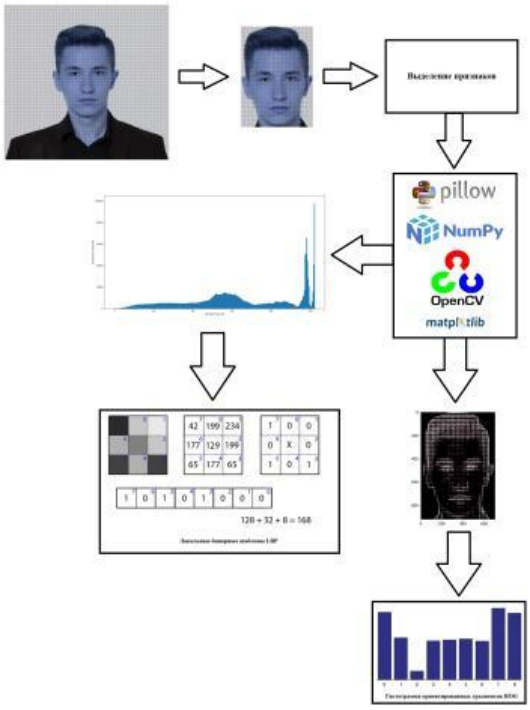


Рисунок 2.8 - Последовательность извлечения вектора признаков изображения лица


Рисунок 2.9 – Схема модели биометрической аутентификации при распознавании пользователей по изображению лица

Система распознавания представлена широким спектром задач: от классификации объекта до идентификации/верификации объекта по биометрическим признакам.

В настоящее время системы видеоаналитики работают с видеопотоком в режиме реального времени или с архивом. Во всех рассмотренных ситуациях это последовательность кадров, которые передаются с некоторой частотой в секунду (FPS, frames per second). Фрейм является статичной картинкой, которая представляет собой некую информацию. В зависимости от формата (количество бит на пиксель) и разрешения каждый фрейм имеет определенный объем.

Во время записи в режиме реального времени видеосигнал обладает следующими характеристиками:

– Частота кадров – 25 FPS.

– Разрешающая способность – 800 × 600 пикселей.

– Битрейт – 8–10 Мбит/сек.

Уровень погрешности распознавания соответствует граничному значению евклидова расстояния между характеристиками лица равен 0,6.

В результате прохождения этапов, разработанный модуль распознает авторизованных пользователей информационной системы (Рисунок 2.10) [44].

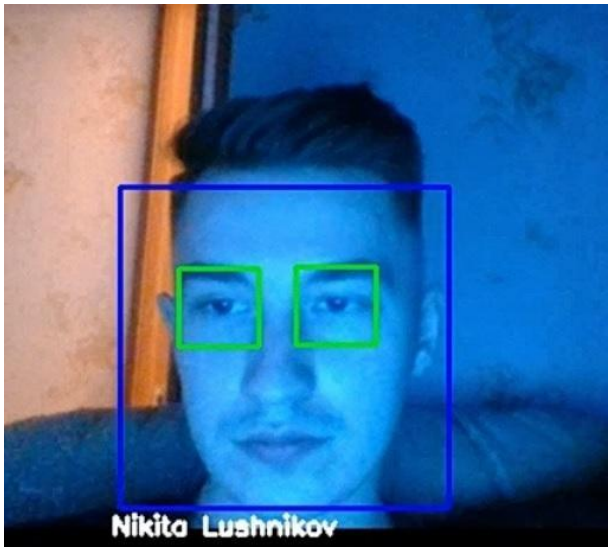


Рисунок 2.10 – Распознавание пользователя информационной системы

Помимо этого, выводится результат противодействия методике синтеза изображения и противодействия распознавания пользователей информационной системы по изображениям, распечатанным на бумажном носителе и сохраненных на других устройствах (Рисунок 2.11) [64].

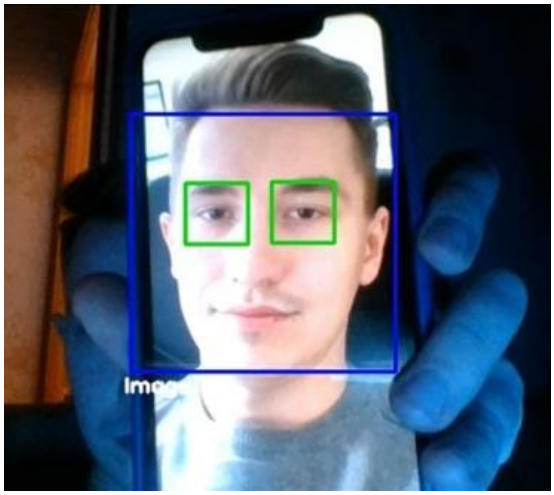

Рисунок 2.11 – Распознавание изображения, сохраненного на другом устройстве

Помимо извлечения LBP и признаков HOG при распознавании пользователей информационной системы на изображении выделяются основные точки лица, необходимые при процедуре аутентификации (Рисунок 2.12) [63].

Рисунок 2.12 – Пример описания точек и контуров лица пользователя информационной системы

Распознавание пользователей информационной системы по изображению лица в видеопотоке реализовано в следующем порядке [111]:

1. Представление исходного изображения лица с видеопотока в монохромном виде.
2. Осуществление поиска ключевых точек и коэффициентов.
3. Сравнение ключевых точек и сортировка результатов.
4. Представление конечного результата (Рисунок 2.13) [33].

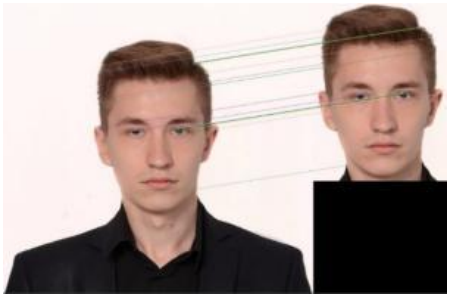

Рисунок 2.13 – Определение ключевых точек изображения лица

В рамках данного исследования были проведены эксперименты с различными архитектурами и наборами обучающих выборок для противодействия синтезу изображения лица. В ходе данного исследования был разработан модуль противодействия методике синтеза изображений

(дипфейк), который позволяет отличить настоящего пользователя информационной системы от объекта, который его олицетворяет (Рисунок 2.14). Проведены эксперименты с различными архитектурами и наборами обучающих выборок для противодействия синтезу изображения лица.

На всех кадрах видеозаписей производился поиск области лица с помощью нейронной сети. Каждое лицо приводилось к размеру 224 × 224 пикселя. На каждой из моделей проводилось тестирование тестовой части всех трёх баз данных (Таблица 2.6).

Таблица 2.6 – Результаты экспериментальных работ с альтернативными архитектурами нейронных сетей на 3 наборах обучающей выборки для противодействия методу синтеза изображений (дипфейк).

| **Модель нейронной сети** | **Обучающая выборка (точность, %)** | | |
|---|---|---|---|
| | **Replay-Attack** | **SiW** | **ROSE** |
| AlexNet | 95,2 % | 54,6 % | 55,8 % |
| VGG | 95,8 % | 73,3 % | 51,6 % |
| DenseNet | 92,9 % | 75,1 % | 61,8 % |
| Сверточная нейронная сеть | 97,8 % | 77,7 % | 79,7 % |

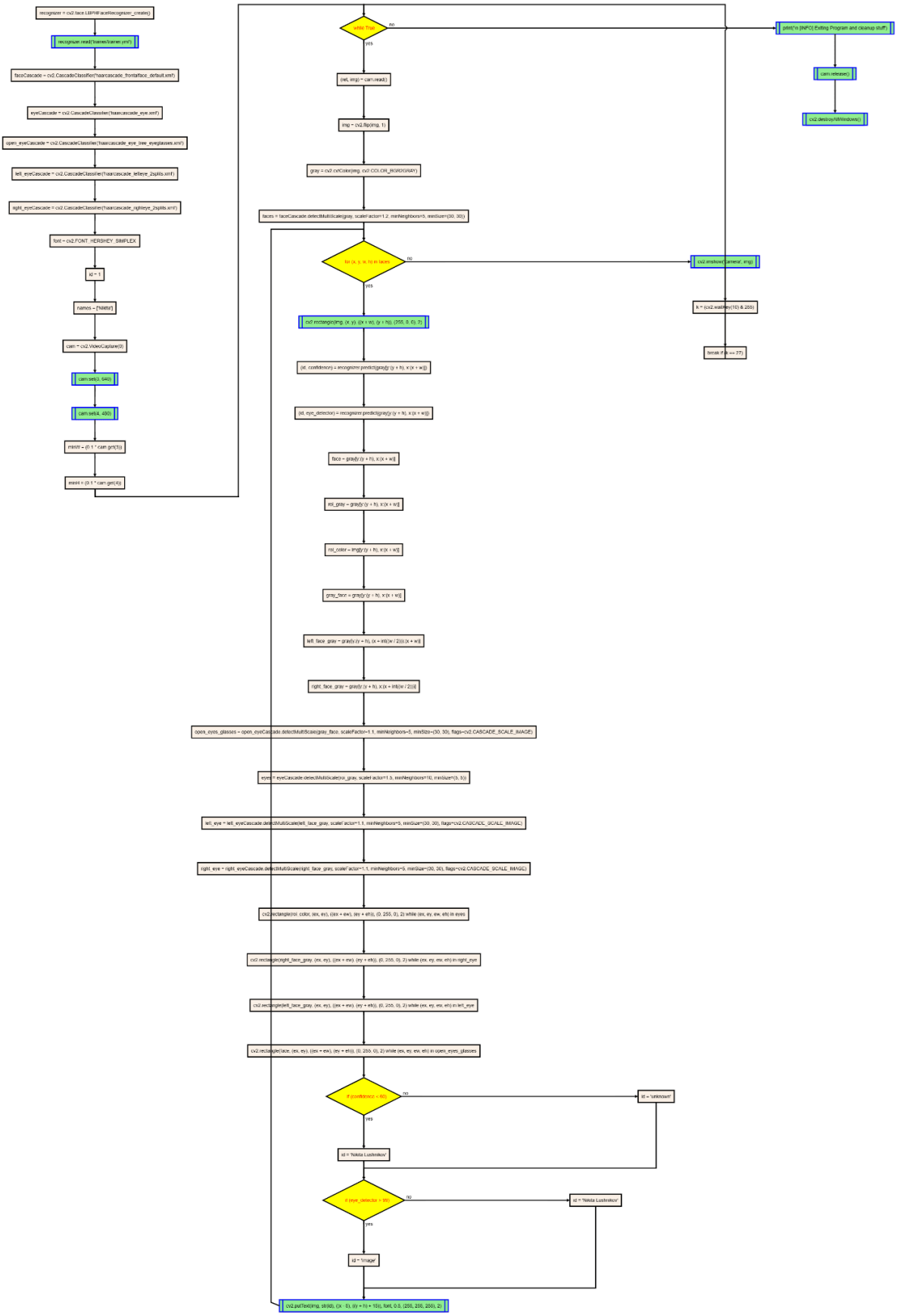


Рисунок 2.14 – Блок-схема программной реализации модуля противодействия методике синтеза изображений (дипфейк) (на основе pyflowchart)

## Выводы и результаты по второй главе

Рассмотрены основные алгоритмы распознавания пользователей информационных систем по изображению лица. Сравнительный анализ алгоритмов позволяет сделать выбор наиболее эффективного метода по извлечению индивидуальных характеристик изображений пользователей информационных систем.

Применены предварительная обработка и фильтры морфологического преобразования изображений пользователей информационной системы.

Реализовано извлечение локальных бинарных шаблонов и признаков гистограммы градиентов, а также графически представлены направления градиентов изображений и гистограммы локальных бинарных шаблонов, направлений и величин градиентов.

Описаны основные аспекты видеоаналитики, представлена иллюстрация алгоритма анализа видеопотока.

Применены фильтры видеозаписи с изображением лиц пользователей информационной системы на основе метода лапласиан гауссиан.

Сформирована обучающая выборка изображений пользователей информационной системы. Представлена архитектура искусственной нейронной сети распознавания пользователей информационных систем по изображению лица и реализован процесс обучения данной нейронной сети с применением составленного программного кода.

Рассмотрены основные методы фильтрации изображений лица. Описан принцип работы адаптивного фильтра Калмана в процессе обработки изображений. Применен метод фильтрации Калмана, который предназначен для шумоподавления изображений пользователей информационной системы.

Таким образом, создана модель распознавания пользователей информационных систем по изображению лица на основе извлечения DLBP и признаков HOG. Данный метод извлечения признаков изображения лица имеет высокие результаты точности обработки данных и, соответственно,

является довольно эффективным инструментом при проведении процедур аутентификации и идентификации.

Создан программный модуль фильтрации изображений лица в процессе распознавания пользователей информационных систем по индивидуальным биометрическим характеристикам. Данная программа позволяет повысить эффективность при распознавании пользователей информационной системы по изображению лица и снизить показатели ошибки первого рода, ошибки второго рода.

Создан программный модуль распознавания пользователей информационных систем на видеозаписи в режиме реального времени, предназначенный для противодействия методике синтеза изображения (дипфейк) и для противодействия распознавания пользователей информационной системы по изображениям, распечатанным на бумажном носителе и сохраненных на других устройствах.

Данная модель использует предварительную обработку и морфологические преобразования изображений лица. В рамках проведенных экспериментальных работ получены высокие показатели помехоустойчивости, выраженные в виде значения критерия среднеквадратичной ошибки MSE = 0,0019.

# ГЛАВА 3. РАЗРАБОТКА И ИССЛЕДОВАНИЕ МОДЕЛИ РАСПОЗНАВАНИЯ ПОЛЬЗОВАТЕЛЕЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО ГОЛОСУ

В данном разделе рассмотрены основные алгоритмы извлечения акустических признаков, особенности речевого сигнала и процессы преобразования аудиофрагментов. Сформирована база биометрических персональных данных (голос) пользователей информационной системы, разработаны две архитектуры искусственных нейронных сетей, предназначенных для распознавания пользователей информационной системы на основе извлеченных акустических признаков. Произведен сравнительный анализ разработанных моделей искусственных нейронных сетей с различными наборами извлеченных биометрических характеристик и приведены основные результаты, а также показатели эффективности работы программного комплекса.

## 3.1. Алгоритм выделения акустических признаков

На начальном этапе производится запись считываемого аудиофайла:

```
def __init__(self):

        self.open = True
        self.rate = 28800
        self.frames_per_buffer = 1024
        self.channels = 1
        self.format = pyaudio.paInt16
        self.audio_filename = "temp_audio2.wav"
        self.audio = pyaudio.PyAudio()
```

Сохранение аудиофайла:

```
def file_manager(filename):

      local_path = os.getcwd()

      if os.path.exists(str(local_path) + "/temp_audio2.wav"):
            os.remove(str(local_path) + "/temp_audio2.wav")
```

Вычисление мощности звука и октав голоса первой аудиозаписи с выводом подробного графика в формате изображения:

```
y,sr=librosa.load(r'temp_audio.wav')

y_harmonic, y_percussive = librosa.effects.hpss(y)

chroma=librosa.feature.chroma_cens(y=y_harmonic, sr=sr)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chroma,y_axis='chroma', x_axis='time')
plt.colorbar()

chroma_mean=np.mean(chroma,axis=1)
chroma_std=np.std(chroma,axis=1)
#plot the summary
octave=['C','C#','D','D#','E','F','F#','G','G#','A','A#','B']
plt.figure(figsize=(15,5))
plt.title('Mean CENS')
sns.barplot(x=octave,y=chroma_mean)

plt.figure(figsize=(15,5))
plt.title('SD CENS 1')
```

```
sns.barplot(x=octave,y=chroma_std)
#Generate the chroma Dataframe
chroma_df=pd.DataFrame()
for i in range(0,12):
    chroma_df['chroma_mean_'+str(i)]=chroma_mean[i]
for i in range(0,12):
    chroma_df['chroma_std_'+str(i)]=chroma_mean[i]
chroma_df.loc[0]=np.concatenate((chroma_mean,chroma_std),axis=0)
chroma_df
plt.savefig('График1.png')
shape1=np.concatenate((chroma_mean,chroma_std),axis=0)
print(shape1)
```

Вычисление мощности звука и октав голоса второй аудиозаписи с выводом подробного графика в формате изображения:

```
y,sr=librosa.load(r'temp_audio2.wav')

y_harmonic, y_percussive = librosa.effects.hpss(y)

chroma=librosa.feature.chroma_cens(y=y_harmonic, sr=sr)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chroma,y_axis='chroma', x_axis='time')
plt.colorbar()

chroma_mean=np.mean(chroma,axis=1)
chroma_std=np.std(chroma,axis=1)
#plot the summary
octave=['C','C#','D','D#','E','F','F#','G','G#','A','A#','B']
plt.figure(figsize=(15,5))
plt.title('Mean CENS')
sns.barplot(x=octave,y=chroma_mean)
```

```
plt.figure(figsize=(15,5))
plt.title('SD CENS 2')
sns.barplot(x=octave,y=chroma_std)
#Generate the chroma Dataframe
chroma_df=pd.DataFrame()
for i in range(0,12):
    chroma_df['chroma_mean_'+str(i)]=chroma_mean[i]
for i in range(0,12):
    chroma_df['chroma_std_'+str(i)]=chroma_mean[i]
chroma_df.loc[0]=np.concatenate((chroma_mean,chroma_std),axis=0)
chroma_df
plt.savefig('График2.png')
shape2=np.concatenate((chroma_mean,chroma_std),axis=0)
print(shape2)
```

Сравнение двух аудиозаписей:

```
a = distance.euclidean(shape1, shape2)
print(a)

if a >= 0.6:
    print("Пользователь не идентифицирован, повторите попытку")
else:
    print("Вы являетесь пользователем системы")
```

Вывод видеоканала совместно с аудиоканалом при параллельной фиксации считываемого изображения:

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
faceCascade=cv2.CascadeClassifier("haarcascade_frontalface_default.xml
")
eyeCascade = cv2.CascadeClassifier("haarcascade_eye.xml")
```

```
open_eyeCascade=cv2.CascadeClassifier("haarcascade_eye_tree_eyeglasses
.xml")

left_eyeCascade=cv2.CascadeClassifier("haarcascade_lefteye_2splits.xml
")

right_eyeCascade=cv2.CascadeClassifier("haarcascade_righteye_2splits.x
ml")


font = cv2.FONT_HERSHEY_SIMPLEX


id = 1


names = ['Unknown','Nikita']


cam = cv2.VideoCapture(0)

cam.set(3, 640) # set video widht

cam.set(4, 480) # set video height
```

Для извлечения речевых сегментов необходимо рассчитать амплитудный спектр [112]:

$$A_k = \left| \sum_{t=0}^{T-1} w_t^{hamm} \times x_t e^{-\frac{2\pi i}{T} kt} \right| \quad k = 0, \dots, \frac{T}{2} \tag{77}$$

где $T$ – размер кадра, $w_t^{hamm}$ – окно Хэмминга, $x_t$ – значение входного сигнала, $k$ – индекс частоты, $t$ – количество кадров.

Далее определяется энергия сигнала для каждого кадра [111]:

$$E = \sqrt{\sum_{K_{beg} \le k \le K_{end}} A_k^2} \tag{78}$$

$$K_{beg} = \frac{300}{\hat{f}} T \tag{79}$$

$$K_{end} = \frac{1500}{\hat{f}} T \tag{80}$$

где $\hat{f}$ – частота дискретизации входного сигнала.

Прямым методом оценки эффективности является основная метрика оценки эффективности информационной системы – коэффициент ошибок диаризации [12]:

$$DER = \frac{\sum_{seg}(T(seg)\times \max(N_{ref}(seg), N_{sys}(seg)) - N_{correct}(seg))}{\sum_{seg} T(seg)\times N_{ref}(seg)} \times 100\% \quad (81)$$

где $T(seg)$ – длительность речевого сегмента $seg$, $N_{ref}(seg)$ – количество пользователей информационной системы, голос которых присутствует на речевом сегменте $seg$ в соответствии с эталонной разметкой, $N_{sys}(seg)$ – количество пользователей информационной системы, голос которых присутствует на речевом сегменте $seg$ в соответствии с результатом работы оцениваемой системы, $N_{correct}(seg)$ – количество достоверно отнесенных к речевому сегменту $seg$ пользователей.

Доля ошибок распознавания включает в себя суммирование трех типов ошибок: ложноположительных ошибок $E_{FA}$, ложноотрицательных ошибок $E_{miss}$ и ошибок разделения пользователей $E_{spkr}$ [116]:

$$E_{FA} = \frac{\sum_{N_{sys}(seg)>N_{ref}(seg)} T(seg) \times \quad (N_{sys}(seg) - N_{ref}(seg)}{\sum_{seg} T(seg)\times N_{ref}(seg)} \quad (82)$$

$$E_{miss} = \frac{\sum_{N_{ref}(seg)>N_{sys}(seg)} T(seg) \times \quad (N_{ref}(seg) - N_{sys}(seg)}{\sum_{seg} T(seg)\times N_{ref}(seg)} \quad (83)$$

$$E_{spkr} = \frac{\sum_{seg}(T(seg)\times \min(N_{ref}(seg), N_{sys}(seg)) - N_{correct}(seg))}{\sum_{seg} T(seg)\times N_{ref}(seg)} \quad (84)$$

Прямой метод расчета эффективности систем разделения пользователей информационной системы заключается в вычислении двух значений: средней чистоты кластера (ACP) и средней чистоты говорящего (ASP) [13,14]:

$$ACP_c = \frac{\sum_{s=1}^{S} n_{sc}^2}{\left(N_c^{cluster}\right)^2} \quad (85)$$

$$ACP = \frac{1}{N}\sum_{c=1}^{M} ACP_c \times N_c^{cluster} \quad (86)$$

$$ASP_s = \frac{\sum_{c=1}^{M} n_{sc}^2}{\left(N_c^{cluster}\right)^2} \quad (87)$$

$$ASP = \frac{1}{N}\sum_{s=1}^{S} ASP_s \times N_s^{speaker} \quad (88)$$

где $S$ – количество пользователей в соответствии с эталонной разметкой, $M$ – полученное в результате работы системы количество кластеров, $n_{sc}$ – количество данных в кластере $c$, которые принадлежат пользователю $s$, $N_c^{cluster} = \sum_{s=1}^{S} n_{sc}$ – количество данных в кластере $c$, $N_s^{speaker} = \sum_{c=1}^{M} n_{sc}$ – количество данных, принадлежащих пользователю $s$, $N = \sum_{s=1}^{S}\sum_{c=1}^{M} n_{sc}$ – количество всех данных.

В качестве итоговой агрегированной оценки системы разделения пользователей информационной системы используется среднее геометрическое значение ASP и ACP, обозначаемое как $K$ [14]:

$$K = \sqrt{ACP \times ASP} \quad (89)$$

Сравнение аудиозаписей и вывод результатов осуществляется следующим образом:

```
'exec(%matplotlib inline)'
import os
import librosa
import librosa.display
import IPython
import numpy as np
import pandas as pd
import scipy
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.spatial import distance
import platform
import wx
```

```
import sys
import ctypes
import subprocess
from tkinter import *
from tkinter import messagebox as mb

y,sr=librosa.load(r'temp_audio.wav')

y_harmonic, y_percussive = librosa.effects.hpss(y)

chroma=librosa.feature.chroma_cens(y=y_harmonic, sr=sr)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chroma,y_axis='chroma', x_axis='time')
plt.colorbar()

chroma_mean=np.mean(chroma,axis=1)
chroma_std=np.std(chroma,axis=1)
octave=['C','C#','D','D#','E','F','F#','G','G#','A','A#','B']
plt.figure(figsize=(15,5))
plt.title('Mean CENS')
sns.barplot(x=octave,y=chroma_mean)

plt.figure(figsize=(15,5))
plt.title('SD CENS 1')
sns.barplot(x=octave,y=chroma_std)
#Generate the chroma Dataframe
chroma_df=pd.DataFrame()
for i in range(0,12):
    chroma_df['chroma_mean_'+str(i)]=chroma_mean[i]
for i in range(0,12):
```

```
chroma_df['chroma_std_'+str(i)]=chroma_mean[i]
chroma_df.loc[0]=np.concatenate((chroma_mean,chroma_std),axis=0)
chroma_df
plt.savefig('SD CENS 1.png')
shape1=np.concatenate((chroma_mean,chroma_std),axis=0)
print(shape1)

y,sr=librosa.load(r'temp_audio2.wav')

y_harmonic, y_percussive = librosa.effects.hpss(y)

chroma=librosa.feature.chroma_cens(y=y_harmonic, sr=sr)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chroma,y_axis='chroma', x_axis='time')
plt.colorbar()
chroma_mean=np.mean(chroma,axis=1)
chroma_std=np.std(chroma,axis=1)
octave=['C','C#','D','D#','E','F','F#','G','G#','A','A#','B']
plt.figure(figsize=(15,5))
plt.title('Mean CENS')
sns.barplot(x=octave,y=chroma_mean)

plt.figure(figsize=(15,5))
plt.title('SD CENS 2')
sns.barplot(x=octave,y=chroma_std)
chroma_df=pd.DataFrame()
for i in range(0,12):
    chroma_df['chroma_mean_'+str(i)]=chroma_mean[i]
for i in range(0,12):
    chroma_df['chroma_std_'+str(i)]=chroma_mean[i]
chroma_df.loc[0]=np.concatenate((chroma_mean,chroma_std),axis=0)
```

```
chroma_df

plt.savefig('SD CENS 2.png')

shape2=np.concatenate((chroma_mean,chroma_std),axis=0)

print(shape2)


a = distance.euclidean(shape1, shape2)

print(a)

if a >= 0.4:

    class MyForm(wx.Frame):

        def __init__(self):

            no_caption = (wx.MAXIMIZE_BOX | wx.RESIZE_BORDER

                          | wx.SYSTEM_MENU | wx.CLOSE_BOX |
wx.CLIP_CHILDREN)

            wx.Frame.__init__(self, None, title='Безопасность входа',
style=no_caption)

            self.panel = wx.Panel(self, -1)

            self.Maximize(True)

            wx.StaticText(self.panel, -1, "ВЫ НЕ ПРОШЛИ
АУТЕНТИФИКАЦИЮ!",

                          (850, 450))
       self.button2=wx.Button(self.panel, -1, "Выйти из системы",

                          (810, 600))

            self.Bind(wx.EVT_BUTTON, self.OnClose, self.button2)

            self.button2.SetDefault()


            button=wx.Button(self.panel, -1, "Ввести пароль",

                          (1000, 600))

            self.Bind(wx.EVT_BUTTON, self.newwindow, button)


            loc = wx.IconLocation(r'C:\Windows\System32\credwiz.exe',
0)

            self.SetIcon(wx.Icon(loc))


        def OnClose(self, event):
```

```
        sys.exit(0)

        def newwindow(self,event):
            secondWindow = window2()
            secondWindow.Show()

    class window2(wx.Dialog):
        def __init__(self):

            wx.Dialog.__init__(self, None, title="Логин")
            self.logged_in = False

            user_sizer = wx.BoxSizer(wx.HORIZONTAL)

            user_lbl = wx.StaticText(self, label="Имя пользователя:")
            user_sizer.Add(user_lbl, 0, wx.ALL|wx.CENTER, 9)
            self.user = wx.TextCtrl(self, style=wx.TE_PROCESS_ENTER)
            self.user.Bind(wx.EVT_TEXT_ENTER, self.onLogin)
            user_sizer.Add(self.user, 0, wx.ALL, 9)

             p_sizer = wx.BoxSizer(wx.HORIZONTAL)

             p_lbl = wx.StaticText(self, label="Пароль:")
             p_sizer.Add(p_lbl, 0, wx.ALL|wx.CENTER, 9)
             self.password = wx.TextCtrl(self,
style=wx.TE_PASSWORD|wx.TE_PROCESS_ENTER)
             self.password.Bind(wx.EVT_TEXT_ENTER, self.onLogin)
             p_sizer.Add(self.password, 0, wx.ALL, 9)

             main_sizer = wx.BoxSizer(wx.VERTICAL)
             main_sizer.Add(user_sizer, 0, wx.ALL, 9)
             main_sizer.Add(p_sizer, 0, wx.ALL, 9)
```

```
            btn = wx.Button(self, label="Подтвердить")
            btn.Bind(wx.EVT_BUTTON, self.onLogin)
            main_sizer.Add(btn, 0, wx.ALL|wx.CENTER, 9)

            self.SetSizer(main_sizer)

        def onLogin(self, event):

            nikita_name = "NikDL"
            user_name = self.user.GetValue()
            nikita_password = "Nik00"
            user_password = self.password.GetValue()
            if (user_name == nikita_name and user_password ==
nikita_password):
                    subprocess.Popen(['sub2.exe'],
stdout=subprocess.DEVNULL)
            else:
                    ctypes.windll.user32.MessageBoxW(0, "Вы
неправильно ввели логин или пароль", "Неавторизованный пользователь",
0)
                    subprocess.Popen(['form2.exe'],
stdout=subprocess.DEVNULL)

    if __name__ == '__main__':
        app = wx.App(False)
        frame = MyForm().Show()
        app.MainLoop()


if a < 0.4:
    subprocess.Popen(['sub2.exe'], stdout=subprocess.DEVNULL)
```

## 3.2. Выбор параметров речевого сигнала: мел-частотных спектральных коэффициентов, коэффициентов линейного предсказания, перцепционных коэффициентов линейного предсказания, частоты спектрального центроида, Q-константных кепстральных коэффициентов, - используемых для формирования биометрического образа пользователя по голосу

Основная задача блока извлечения признаков — генерация цепочки векторов признаков из исходного сигнала. Сегмент извлечения признаков сканирует входной сигнал с помощью кратковременного скользящего окна, в пределах которого генерируется один вектор признаков (Рисунок 3.1).

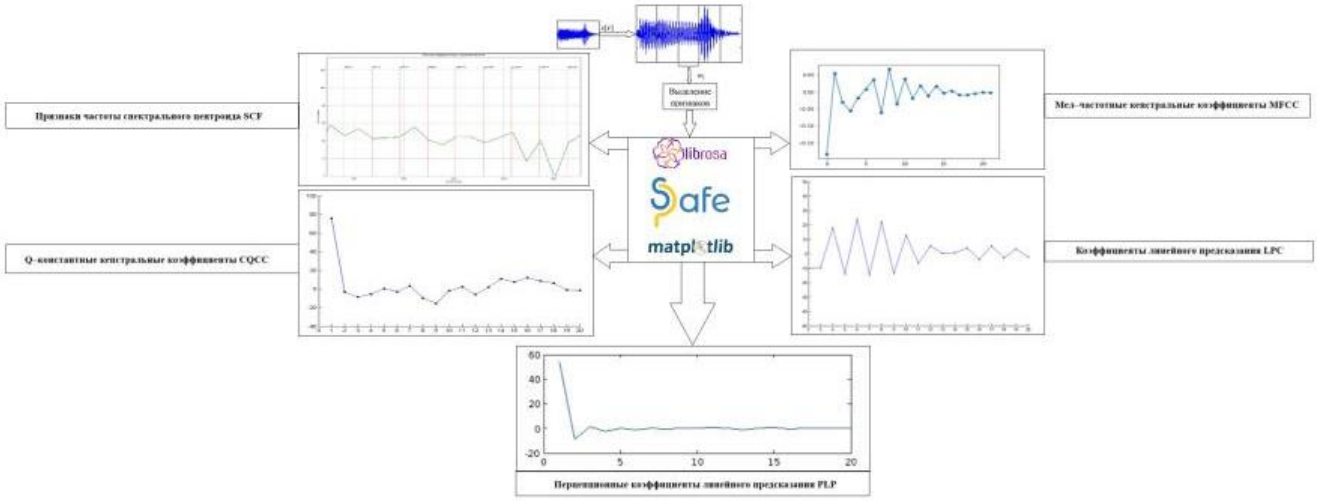


Рисунок 3.1 – Последовательность выделения акустических признаков

Основная концепция методов линейного предсказания заключается в возможности аппроксимировать текущий отсчет речевого сигнала линейной комбинацией предыдущих отсчетов [92].

Передаточная функция линейной системы, которая включает возбуждение и речевой сигнал, отражена следующим образом:

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} \tag{90}$$

где $G$ – коэффициент усиления возбуждения.

Преимущество кепстральных коэффициентов перед коэффициентами линейного предсказания и перцептивными коэффициентами линейного

предсказания заключается в простоте их реализации при сохранении схожей производительности распознавания.

Существует также группа признаков временной области. К ним относятся частота нулевого перехода (ZCR) и кратковременная энергия (STE) [105]. Первый метод позволяет быстро оценить спектральные характеристики речевого сигнала. Вычисление ZCR позволяет отличить вокализованные звуки от невокализованных, поскольку высокие частоты приводят к большему количеству нулевых переходов, а низкие — к меньшему.

## 3.3. Обработка и нормализация сформированного биометрического образа пользователей по голосу

Для уменьшения возможных акустических и канальных искажений, аддитивного шума, необходимо обработать акустические характеристики:

$$|F_k(t)| = |G_k|(|S_k(t)| + |N_k|), k = 0, \dots, \frac{T}{2} \quad (91)$$

где $|S_k(t)|$ – амплитудный спектр речевого сигнала, $|N_k|$ – амплитудный спектр аддитивного шума, $|G_k|$ – амплитудно–частотная характеристика (АЧХ) канальных искажений.

При подсчете логарифма энергии в банке фильтров канальные искажения являются частью аддитивного сигнала:

$$E_s^{Bark}(t) = \log\left(\sum_{k=0}^{\frac{T}{2}} |F_k(t)|\, H_s^{Bark}(f_k)\right) = \log\left(\hat{G}_s\right) + \log\left(\hat{S}_s(t) + \hat{N}_s\right) \quad (92)$$

где $\hat{S}_s(t), \hat{N}_s, \hat{G}_s$ – результат применения s–го фильтра к спектру сигнала, шума и АЧХ канала:

$$\hat{S}_s(t) = \sum_{k=0}^{\frac{T}{2}} |S_k(t)|\, H_s^{Bark}(f_k) \quad (93)$$

$$\widehat{N}_s(t) = \sum_{k=0}^{\frac{T}{2}} |N_k(t)|\, H_s^{Bark}(f_k) \tag{94}$$

$$\widehat{G}_s(t) = \sum_{k=0}^{\frac{T}{2}} |G_k(t)|\, H_s^{Bark}(f_k) \tag{95}$$

Аддитивный шум является триггером для нелинейных изменений результирующих коэффициентов [77]:

$$\tilde{C}_l(t) = \frac{\hat{C}_l(t)}{\sigma C_l} \tag{96}$$

где $\sigma C_l$ – среднее–квадратичное отклонение значения коэффициента $C_l(t)$ на всех речевых сегментах фонограммы.

Существует множество методов шумоподавления, предназначенных для устранения ненужных шумов и помех. К ним относятся метод спектрального вычитания [84], а также метод удаления шума и спектрального представления, совместно разработанный с PLP, известный как RASTA (representations of relative spectra) (Рисунок 3.2) [90].

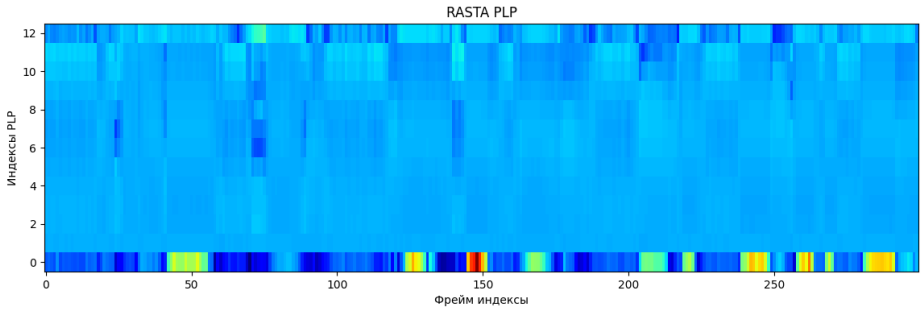


Рисунок 3.2 – Метод шумоочистки и представления спектров на основе PLP

## 3.4. Обоснование выбора и использование нейронной сети для распознавания пользователей информационной системы по голосу

В рамках данного исследования сформирована база данных аудиозаписей, которая состоит из трех папок: test (тестовый набор), train (тренировочный набор) и val (валидационный набор). Каждая из этих папок разбивается на два класса и содержит аудиофрагменты длиной в 5, 8 и 12

секунд. Общий объем биометрической базы данных составляет 450 аудиозаписей пользователей информационной системы [51,52]. Для реализации программного комплекса распознавания пользователей информационных систем по голосу, который будет представлять собой эффективное и усовершенствованное решение [19], разработаны две архитектуры искусственной нейронной сети с разным подбором акустических признаков на входном слое.

Первая архитектура искусственной нейронной сети, построенная на основе акустических признаков речевого сигнала MFCC, LPC и PLP, состоит из 13 нейронов входного слоя (№ ПС – № пользователя информационной системы, извлеченные коэффициенты MFCC, LPC, PLP), 13 нейронов одного скрытого слоя (количество пользователей информационной системы, промежуточные результаты активации коэффициентов MFCC, LPC, PLP). Выходной слой состоит из результатов, который определяет авторизованных пользователей информационной системы (АП) и неавторизованных пользователей (НАП) (Рисунок 3.3) [42].

Вторая архитектура искусственной нейронной сети, построенная на основе акустических признаков речевого сигнала MFCC, LPC, PLP, CQCC и SCF, состоит из 19 нейронов входного слоя (№ ПС – № пользователя информационной системы, извлеченные коэффициенты MFCC, LPC, PLP, CQCC, SCF), 20 нейронов одного скрытого слоя (количество пользователей информационной системы, промежуточные результаты активации коэффициентов MFCC, LPC, PLP, CQCC, SCF). Выходной слой состоит из результатов, который определяет авторизованных пользователей информационной системы (АП) и неавторизованных пользователей (НАП) (Рисунок 3.4) [31].

Значение выходного нейрона получено в результате применения функции активации входного значения нейрона $f(x) = \frac{1}{1+e^{-x}}$.

Для определения уровня погрешности при работе модели следует воспользоваться вычислением евклидова расстояния $d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$ [59].

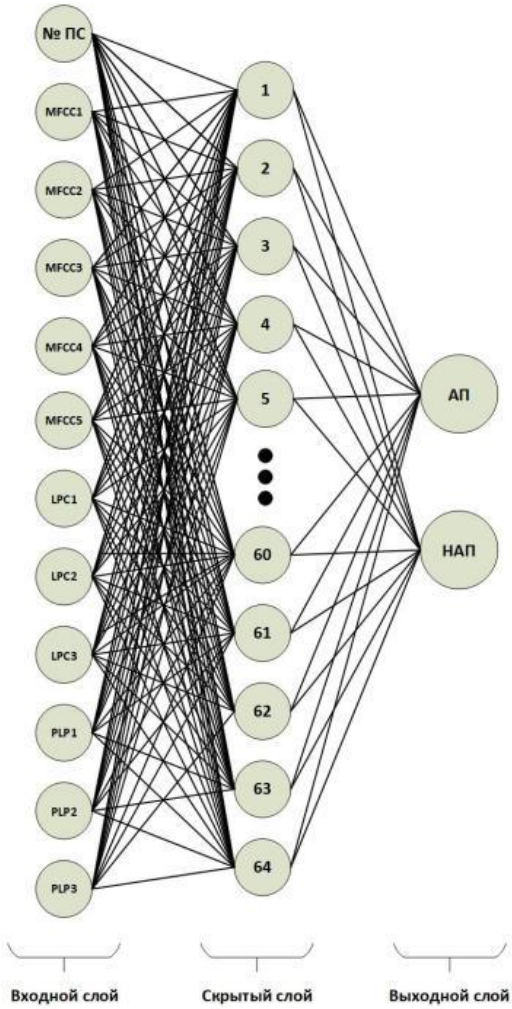


Рисунок 3.3 – Архитектура искусственной нейронной сети в модели распознавания пользователей информационных систем по голосу на основе MFCC, LPC, PLP

По результатам обучения искусственных нейронных сетей следует сделать вывод: система распознавания личности нуждается в укрупнении как базы данных, так и извлекаемых признаков, которые являются индивидуальными характеристиками пользователей информационных систем [64].

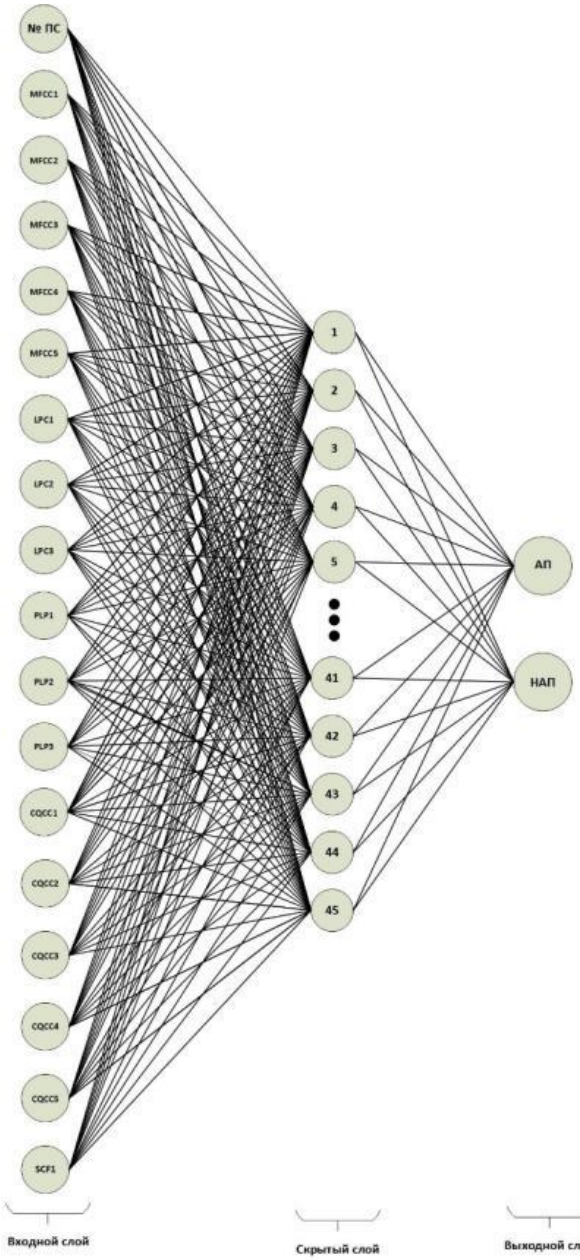


Рисунок 3.4 – Архитектура искусственной нейронной сети распознавания пользователей информационных систем по голосу на основе MFCC, LPC, PLP, CQCC, SCF

## 3.5. Модель распознавания пользователей информационной системы по голосу

Последовательность разработанной модели распознавания пользователей информационной системы по голосу включает интерпретирована следующими этапами (Рисунок 3.5):

1) Обработка входных данных.

2) Применение метода шумоочистки и представления спектров речевых сигналов.

3) Извлечение акустических признаков и формирование вектора признаков.

4) Составление архитектуры сверточной нейронной сети.

5) Формирование обучающей выборки и применение сверточной нейронной сети.

6) Результаты распознавания пользователей по голосу.

В ходе исследования была реализована архитектура нейронной сети Wav2vec и проведено обучение на различных наборах данных (обучающие выборки) (Рисунок 3.6) [64].

Блок-схема алгоритма обучения сверточной нейронной сети модели биометрической аутентификации пользователей информационной системы по голосу, отличающегося извлечением дополнительных речевых признаков, представлена на рисунке 3.7. Извлечение дополнительных речевых признаков позволяет минимизировать возможность синтеза голоса (дипфейк).

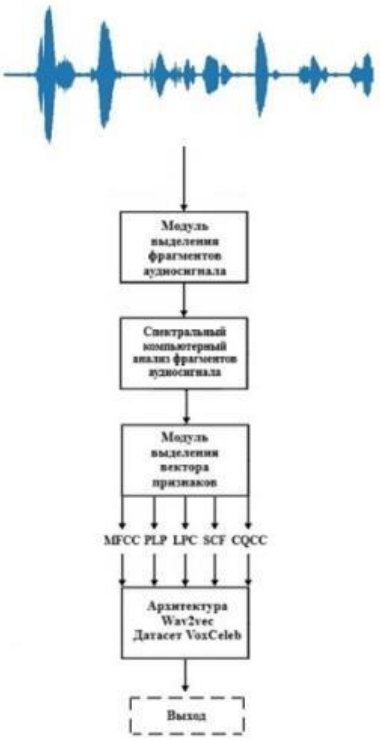


Рисунок 3.5 – Схема модели биометрической аутентификации при распознавании пользователей по голосу

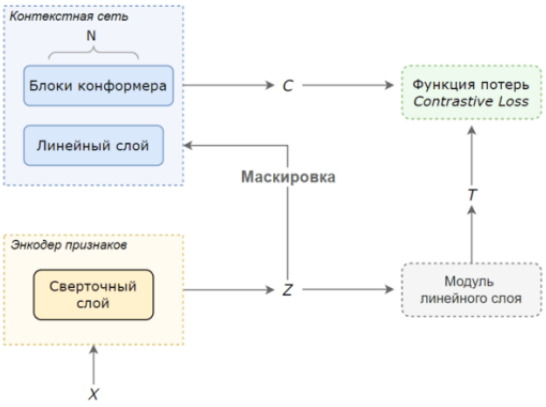


Рисунок 3.6 – Схема обучения нейронной сети Wav2vec для модели биометрической аутентификации по голосу

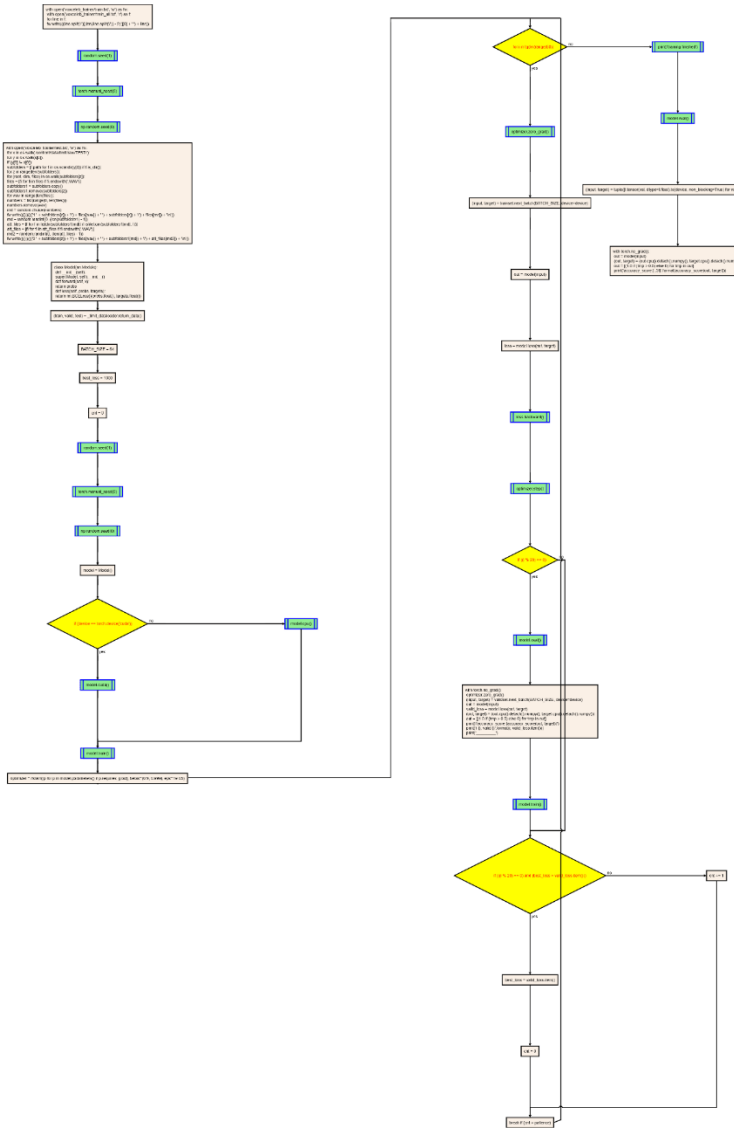

Рисунок 3.7 - Блок-схема алгоритма обучения сверточной нейронной сети модели биометрической аутентификации пользователей информационной системы по голосу (на основе pyflowchart)

Объем собственной обучающей выборки для каждой архитектуры нейронной сети по акустическим признакам составляет 450 аудиозаписей двух пользователей информационной системы. Длительность каждой аудиозаписи составляет 8, 15 и 25 секунд соответственно. Аудиозапись длительностью 8 секунд представлена в виде файла с записанным голосом, в котором производится счет от одного до пяти. В аудиозаписи длительностью 15 секунд производится счет от одного до десяти, а в аудиозаписи длительностью 25 секунд – от одного до двадцати.

Для минимизации различного рода канальных и акустических искажений, а также аддитивного шума акустических признаков следует произвести обработку (на примере PLP).

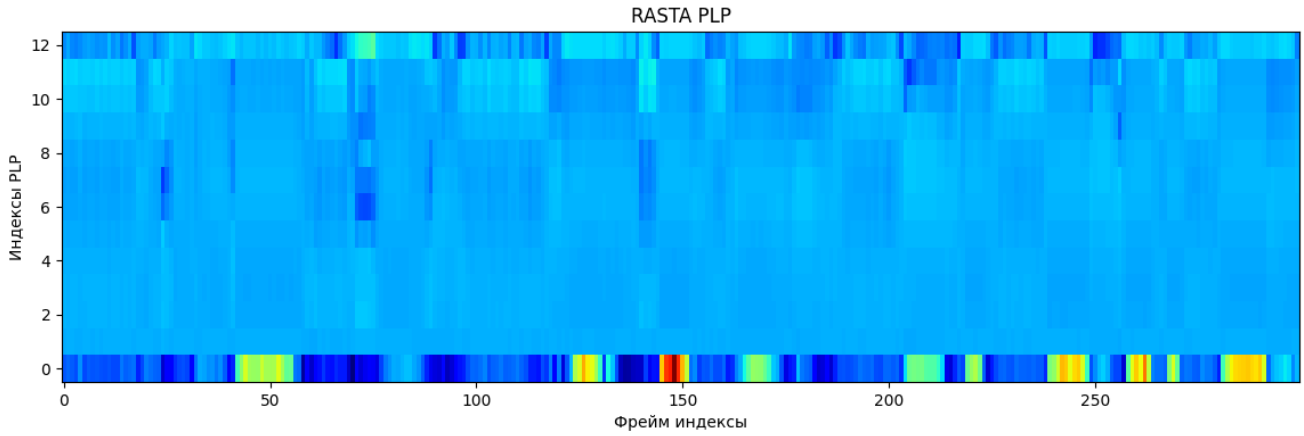


Рисунок 3.8 – Метод шумоочистки и представления спектров на основе PLP

В качестве итоговых значений берутся первые несколько коэффициентов дискретного косинусного преобразования (Рисунок 3.9). Выбрано 12 коэффициентов.

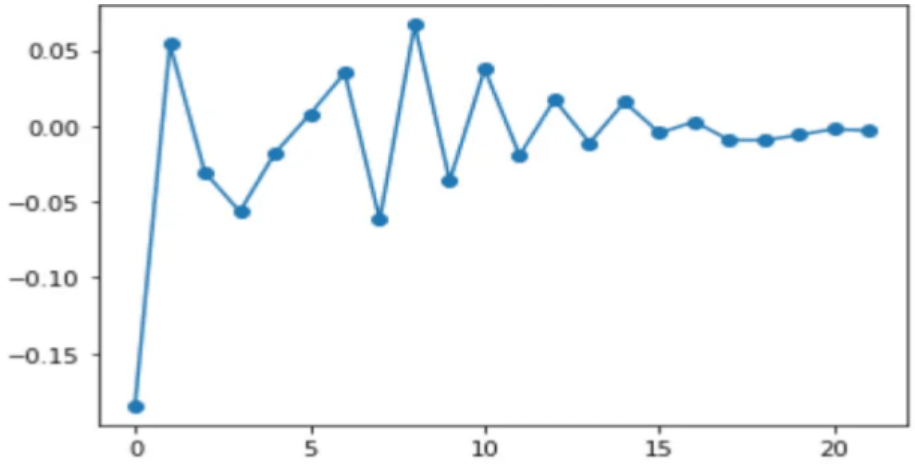


Рисунок 3.9 – Мел–частотные кепстральные коэффициенты аудиофрагмента

Элементы матрицы $\{\varphi_{ij}^{auto}\}_{i,j=1}^{P}$ являются автокорреляционными коэффициентами, $\alpha_p$ – коэффициент минимума общей ошибки предсказания [104].

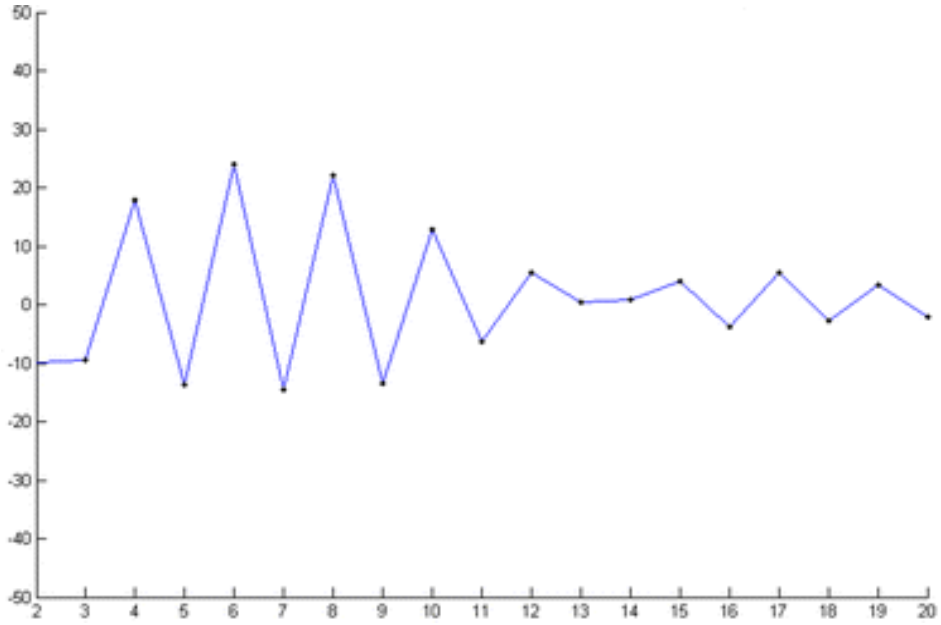


Рисунок 3.10 – Коэффициенты линейного предсказания аудиофрагмента

В качестве итоговых перцепционных коэффициентов линейного предсказания аудиозаписей выбраны коэффициенты линейного предсказания, подсчитанные для величин $\Phi_s^{Bark}$ (Рисунок 3.11).

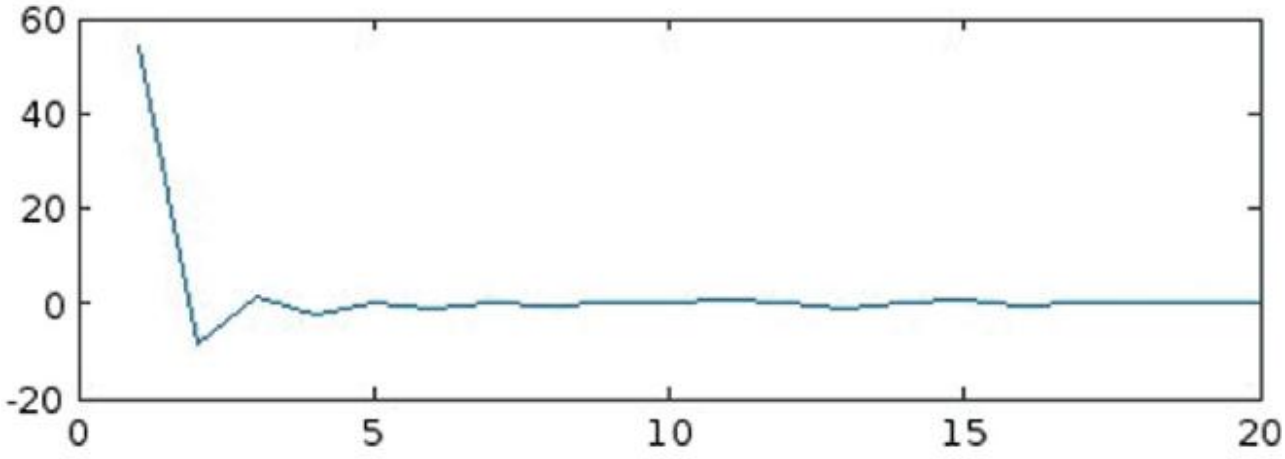


Рисунок 3.11 – Перцепционные коэффициенты линейного предсказания аудиофрагмента

Амплитуда спектрального центроида – это амплитуда в частотной позиции спектрального центроида, которая также несет формантную информацию, необходимую для распознавания личности. (Рисунок 3.12).

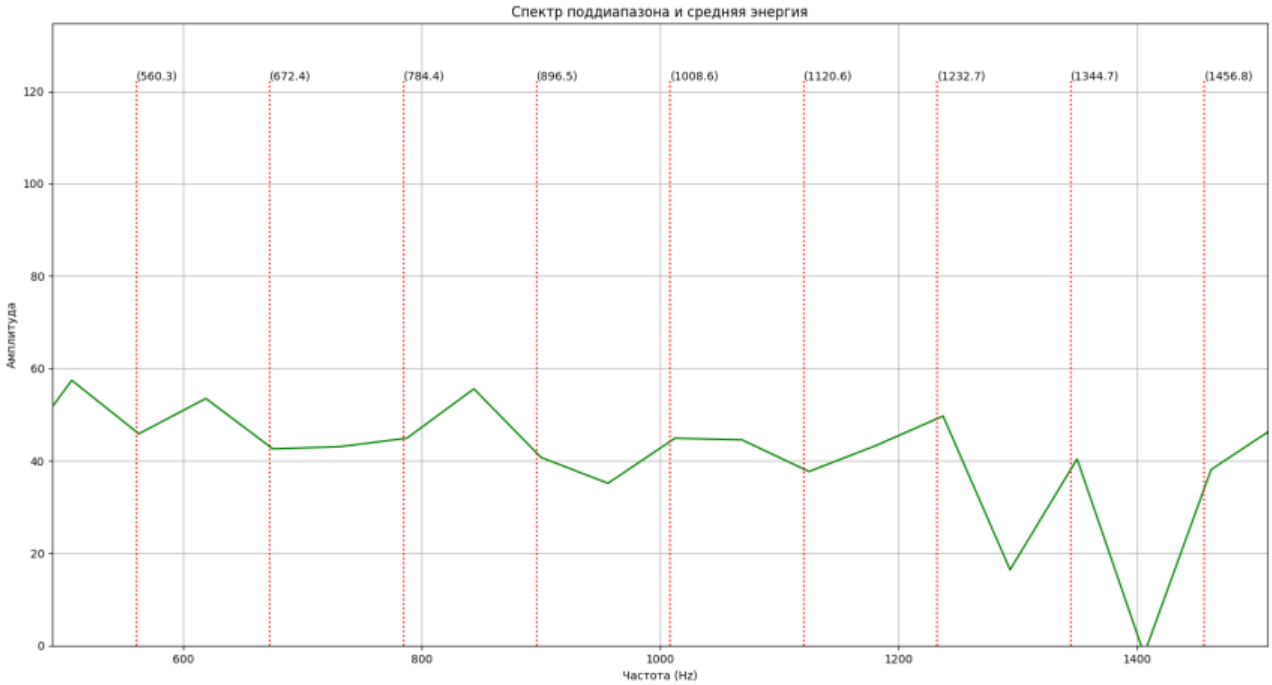


Рисунок 3.12 – Спектр поддиапазона и средняя энергия аудиофрагмента

Для получения коэффициентов необходимо применить дискретное косинусное преобразование для вычисленных значений энергий фильтров к

логарифму энергетического спектра, полученного константным Q–преобразованием (Рисунок 3.13) [98].

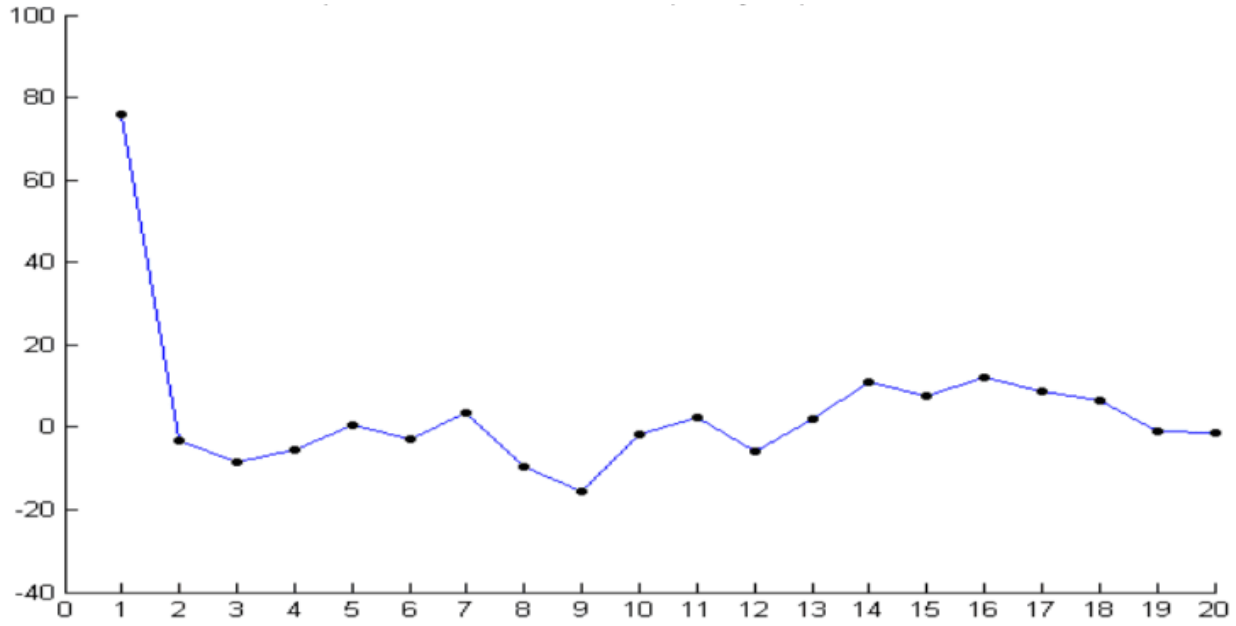


Рисунок 3.13 – Q–константные кепстральные коэффициенты аудиофрагмента

**Выводы и результаты по третьей главе**

В рамках данного исследования реализован алгоритм извлечения мел-частотных кепстральных коэффициентов MFCC, коэффициентов линейного предсказания LPC, перцепционных коэффициентов линейного предсказания PLP, дополнительных речевых признаков: Q–константных кепстральных коэффициентов CQCC, признаков частоты спектрального центроида SCF.

Сформирована обучающая выборка извлеченных биометрических признаков голоса. Представлены архитектуры нейронных сетей распознавания пользователей информационных систем по голосу, проведен сравнительный анализ представленных архитектур и реализован процесс обучения нейронных с применением составленного программного кода.

Приведены основные показатели эффективности, которые отражают функциональность примененных архитектур нейронных сетей.

Таким образом, создана модель распознавания пользователей информационных систем по голосу на основе биометрических характеристик.

Разработан программный модуль распознавания пользователей информационных систем по голосу, который позволит защитить биометрические персональные данные пользователей информационной системы от несанкционированного доступа.

Разработана модель биометрической аутентификации пользователей информационной системы по голосу с применением метода шумоочистки и представления спектров речевых сигналов, позволяющая в составе мультимодальной биометрической системы уменьшить показатели ошибок первого и второго рода при распознавании пользователей информационной системы. Предложенная модель позволяет повысить точность распознавания пользователей по голосу и уменьшить показатели функции потерь на основании примененных средств обработки аудиосигнала. Показатель ошибки первого рода FRR = 0,11 % и показатель ошибки второго рода FAR = 0,01 %.

# ГЛАВА 4. АППРОБАЦИЯ РАЗРАБОТАННЫХ МОДЕЛЕЙ РАСПОЗНАВАНИЯ, ФОРМИРОВАНИЕ СИСТЕМЫ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ КОНКАТЕНАЦИИ ВЕКТОРА ПРИЗНАКОВ И СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ, ОЦЕНКА ЕЕ ЭФФЕКТИВНОСТИ

В данном разделе рассмотрены структура и архитектура нейронной сети на основе конкатенации. Приведены результаты и показатели нейронной сети на основе конкатенации. Приведены показатели модели распознавания пользователей информационной системы по изображению лица и модели распознавания пользователей информационной системы по голосу.

## 4.1. Результаты распознавания пользователей информационной системы по изображению лица

На начальном этапе включаем все необходимые для реализации импортируемые элементы.

```
'exec(%matplotlib inline)'
import librosa
import librosa.display
import IPython
import numpy as np
import pandas as pd
import scipy
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
import face_recognition
import dlib
import platform
import wx
```

```
import sys
import pyaudio
import wave
import threading
import time
import subprocess
import os
from tqdm import tqdm
from collections import defaultdict
from skimage import io
from scipy.spatial import distance
from tkinter import *
from tkinter import messagebox as mb
from scipy.spatial import distance
```

Далее прикрепляем к нейронным сетям наименования с содержанием датасетов.

```
sp = dlib.shape_predictor('shape_predictor_68_face_landmarks. dat')
facerec = dlib.face_recognition_model_v1('dlib_face_recognition_
resnet_model_v1.dat')
detector = dlib.get_frontal_face_detector()
```

Следующим шагом будет являться загрузка считываемого изображения, создание окна данного изображения с работой детектора.

```
img = io.imread('lushnikov_passport.jpg')

win1 = dlib.image_window()
win1.clear_overlay()
win1.set_image(img)
```

```
dets = detector(img, 1)
```

Затем создается массив данных после работы детектора. После создания массива окно со считываемым изображением удаляется.

```
for k, d in enumerate(dets):
    print("Detection {}: Left: {} Top: {} Right: {} Bottom:
{}".format(
        k, d.left(), d.top(), d.right(), d.bottom()))
    shape = sp(img, d)
    win1.clear_overlay()
    win1.add_overlay(d)
    win1.add_overlay(shape)
    del win1
```

В конце необходимо вывести итог считывания. Данное действие выполняет дескриптор.

```
face_descriptor1 = facerec.compute_face_descriptor(img, shape)

print(face_descriptor1)
```

Последующее считывание сверяемого изображения имеет практически аналогичный характер. Помимо всего вышеперечисленного в следующем действии необходимо добавить считывание изображения с веб-камеры.

```
cap = cv2.VideoCapture(0)
for i in range(220):
    cap.read()
ret, frame = cap.read()
cv2.imwrite('cam.jpg', frame)
cap.release()
```

Затем мы производим все те же действия, что произвели ранее с первым изображением. В данном случае актуальным снимком будет изображение cam.jpg.

```
img = io.imread('cam.jpg')


win2 = dlib.image_window()
win2.clear_overlay()
win2.set_image(img)


dets = detector(img, 1)


for k, d in enumerate(dets):
    print("Detection {}: Left: {} Top: {} Right: {} Bottom:
{}".format(
        k, d.left(), d.top(), d.right(), d.bottom()))
    shape = sp(img, d)
    win2.clear_overlay()
    win2.add_overlay(d)
    win2.add_overlay(shape)
    del win2


face_descriptor2 = facerec.compute_face_descriptor(img, shape)


print(face_descriptor2)
```

После считывания изображений их необходимо сверить.

```
a = distance.euclidean(face_descriptor1, face_descriptor2)
```

Отталкиваясь от полученного результата подсчитанного расстояния между дескрипторами, функция if определяет итоги реализации данной

программы. В случае, если расстояние больше или равно 0.6, то выводится следующее окно.

```
if a >= 0.6:
    class MyForm(wx.Frame):
        def __init__(self):
            no_caption = (wx.MAXIMIZE_BOX | wx.RESIZE_BORDER
                          | wx.SYSTEM_MENU | wx.CLOSE_BOX |
wx.CLIP_CHILDREN)
            wx.Frame.__init__(self, None, title='Безопасность входа',
style=no_caption)
            self.panel = wx.Panel(self, -1)
            self.Maximize(True)
            wx.StaticText(self.panel, -1, "ВЫ НЕ ПРОШЛИ
АУТЕНТИФИКАЦИЮ!",
                          (650, 280))
            self.button2=wx.Button(self.panel, -1, "Выйти из системы",
                          (610, 430))
            self.Bind(wx.EVT_BUTTON, self.OnClose, self.button2)
            self.button2.SetDefault()

            button=wx.Button(self.panel, -1, "Ввести пароль",
                          (800, 430))
            self.Bind(wx.EVT_BUTTON, self.newwindow, button)
             loc = wx.IconLocation(r'C:\Windows\System32 \credwiz.exe',
 0)
             self.SetIcon(wx.Icon(loc))

         def OnClose(self, event):
             sys.exit(0)

         def newwindow(self,event):
             secondWindow = window2()
             secondWindow.Show()
```

Для повышения точности распознавания личности пользователя информационной системы необходимо применить морфологическое преобразование изображений лица. Основным результатом применения фильтра Калмана является минимизация дисперсионной ошибки.

С применением адаптивного фильтра Калмана обрабатываемые точки новых изображений лица пользователей всегда находятся рядом с точками предыдущих изображений лица этих же пользователей, поэтому резких скачков характеристик изображений (при составленном прогнозе об их изменении) не происходит.

Для проведения фильтрации использовалась обучающая выборка, предназначенная для компиляции модели нейронной сети при распознавании пользователей информационной системы по изображению лица (Рисунок 4.1) [18].

Рисунок 4.1 – Результаты фильтрации белого шума на изображении

Результатом применения адаптивного фильтра Калмана является разработанное на языке программирования Python 3.8 программное обеспечение, предназначенное для обработки и шумоподавления изображений в процессе распознавания пользователей информационной системы (Рисунок 4.2).

К основным показателям эффективности работы фильтра Калмана следует отнести точность оценки и прогнозирования [61]. При проведении

тестирования программного обеспечения было выявлено: данный алгоритм фильтрации обладает высокой степенью подавления цифрового шума, являясь средством предобработки и морфологического преобразования изображений.

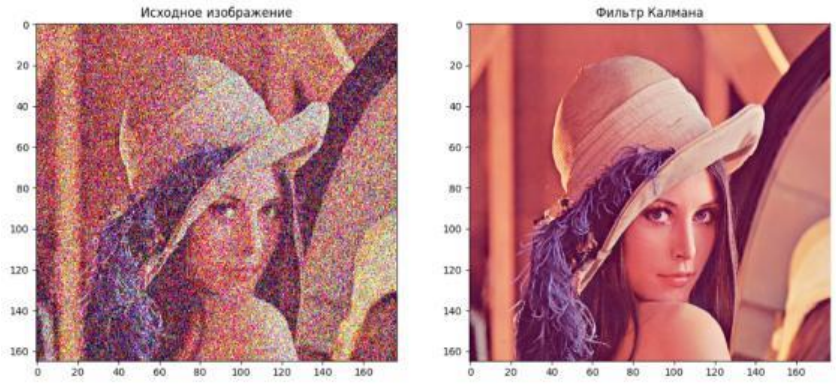


Рисунок 4.2 – Результаты фильтрации белого шума на изображении

Для морфологического преобразования изображений обучающей выборки применяется блочная обработка с изменением размерности и последующим извлечением биометрических признаков изображений лица (Рисунок 4.3).

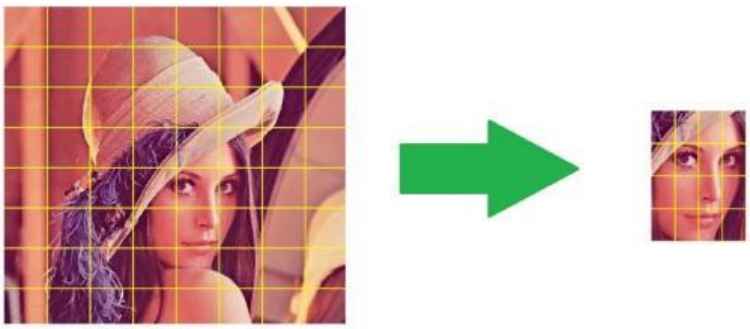

Рисунок 4.3 – Квадратичная матрица разных размерностей с применением фильтра Калмана при распознавании личности

В рамках данного исследования был проведен анализ воздействия цифрового шума на исходное изображение, а также приведены гистограммы RGB-коэффициентов [61]. В начале был произведен анализ зашумленного изображения (Рисунок 4.4).

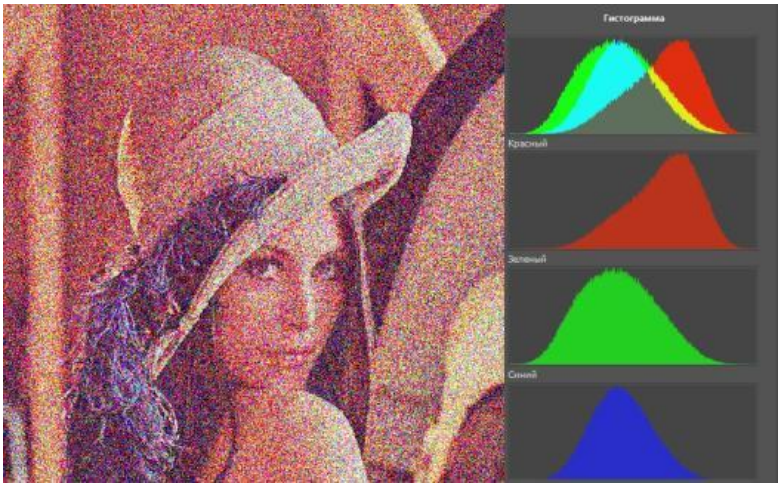


Рисунок 4.4 – Гистограмма RGB-коэффициентов зашумленного изображения

В ходе исследования было выявлено, что на изображении воспринимается 1/3 цветовой информации участка изображения, а 2/3 отсекается. Для получения более чётких изображений была произведена корректировка RGB-коэффициентов в ходе фильтрации (Рисунок 4.5). Таким образом, применение фильтра Калмана способствует снижению резкости изображения. Данный подход позволяет избавиться от некоторых дефектных пикселей при использовании изображения в разных графических форматах.

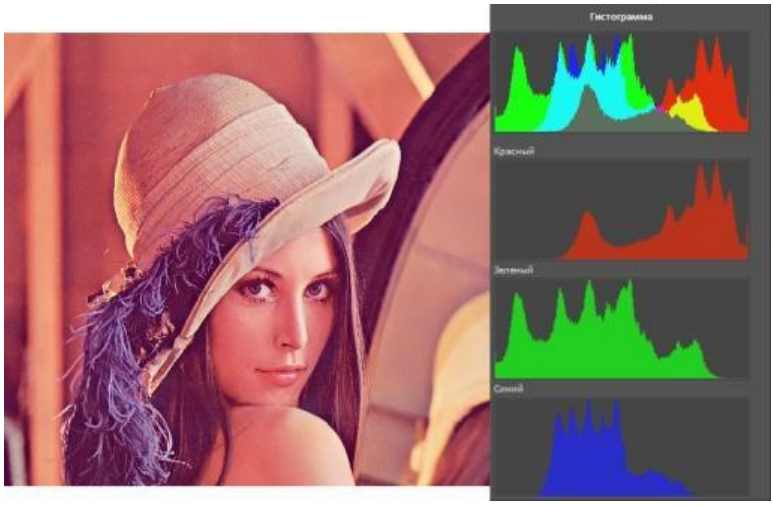


Рисунок 4.5 – Применение фильтрации при анализе RGB-коэффициентов в изображении

Также в исследовании были проведены эксперименты по применению фильтра Калмана в состоянии ночного освещения (Рисунок 4.6). При ночном освещении наиболее часто проявляются дефектные пиксели. Не все устройства обладают возможностью автоматической калибровки

обнаруженных дефектов матрицы в связи с ограничениями аппаратных ресурсов.

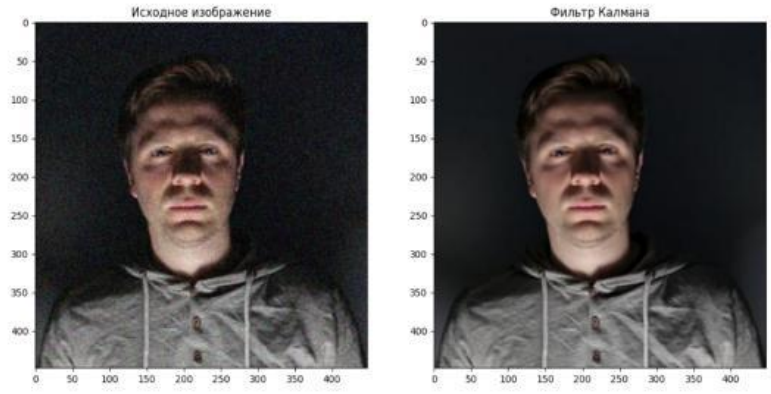


Рисунок 4.6 – Описание спектральных характеристик изображения с выведенным результатом применения фильтрации

При обработке информации изображение делится на блоки, для каждого из которых рассчитывается собственная гистограмма LBP (Рисунок 4.7).

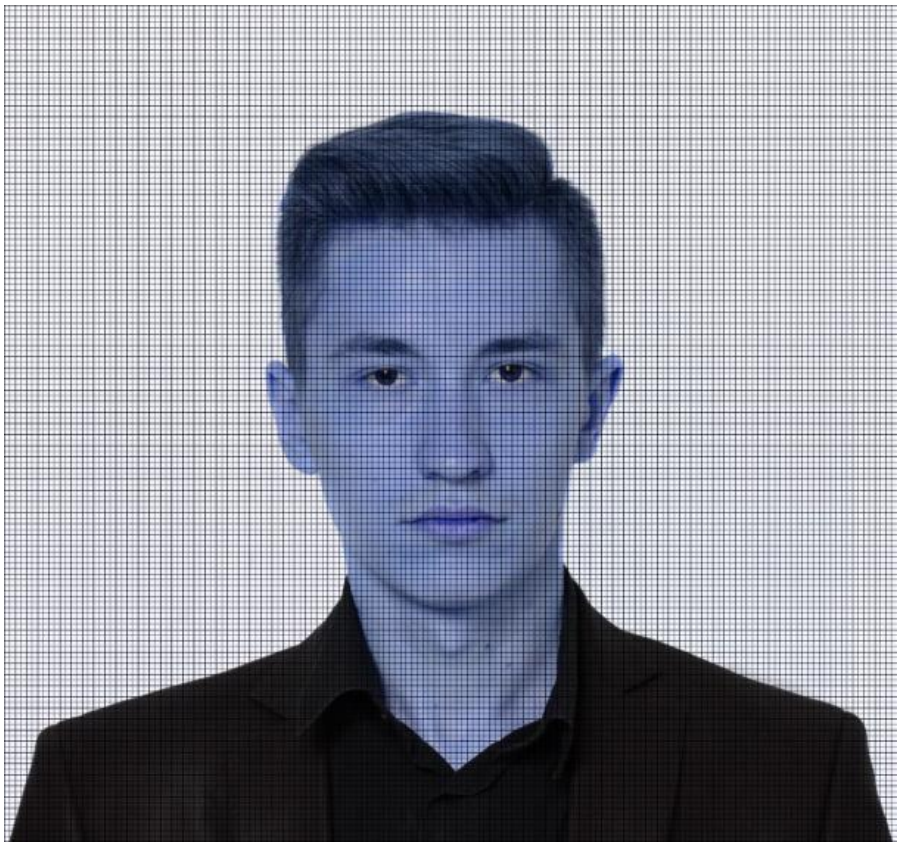

Рисунок 4.7 – Пиксельное представление изображения

Значения в гистограмме LBP представлены с помощью исходного программного кода на языке программирования Python 3.8 с применением библиотек PIL, numpy и matplotlib (Рисунок 4.8).

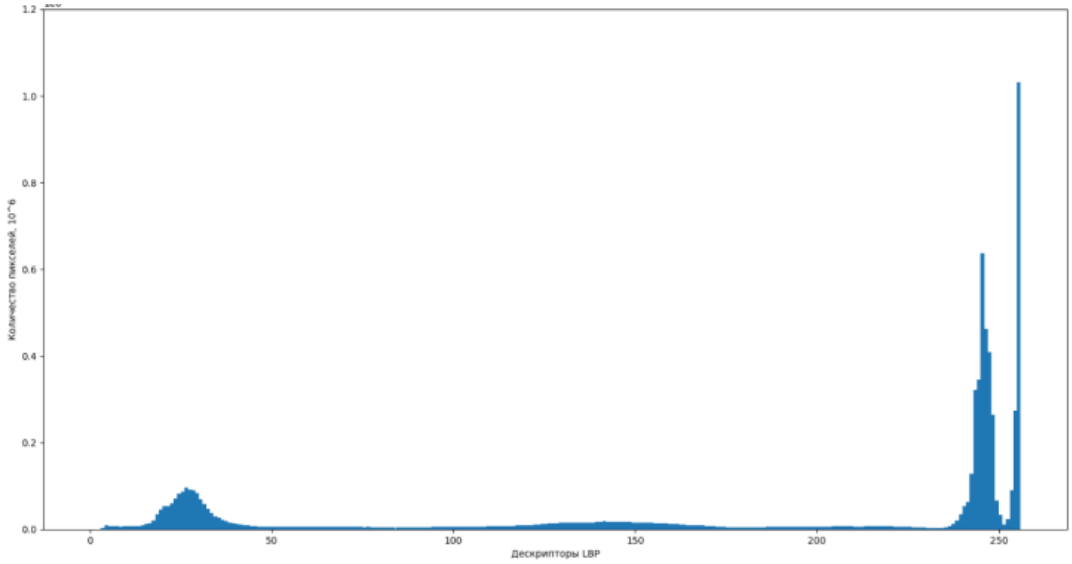


Рисунок 4.8 – Гистограмма LBP для всех пикселей изображения

Чтобы извлечь только LBP пользователя информационной системы, следует обрезать изображение и выделить лицо (Рисунок 4.9).

Рисунок 4.9 – Пиксельное представление изображения лица пользователя информационной системы

В результате выделения лица пользователя информационной системы на изображении есть возможность вычислить значимые (доминантные) DLBP. Значения коэффициентов LBP пользователя информационной системы в данном исследовании являются значимыми (доминантными) коэффициентами DLBP (Рисунок 4.10). Для подсчета значимых (доминантных) DLBP следует сохранить новое изображение и после этого произвести анализ с помощью построенной гистограммы.

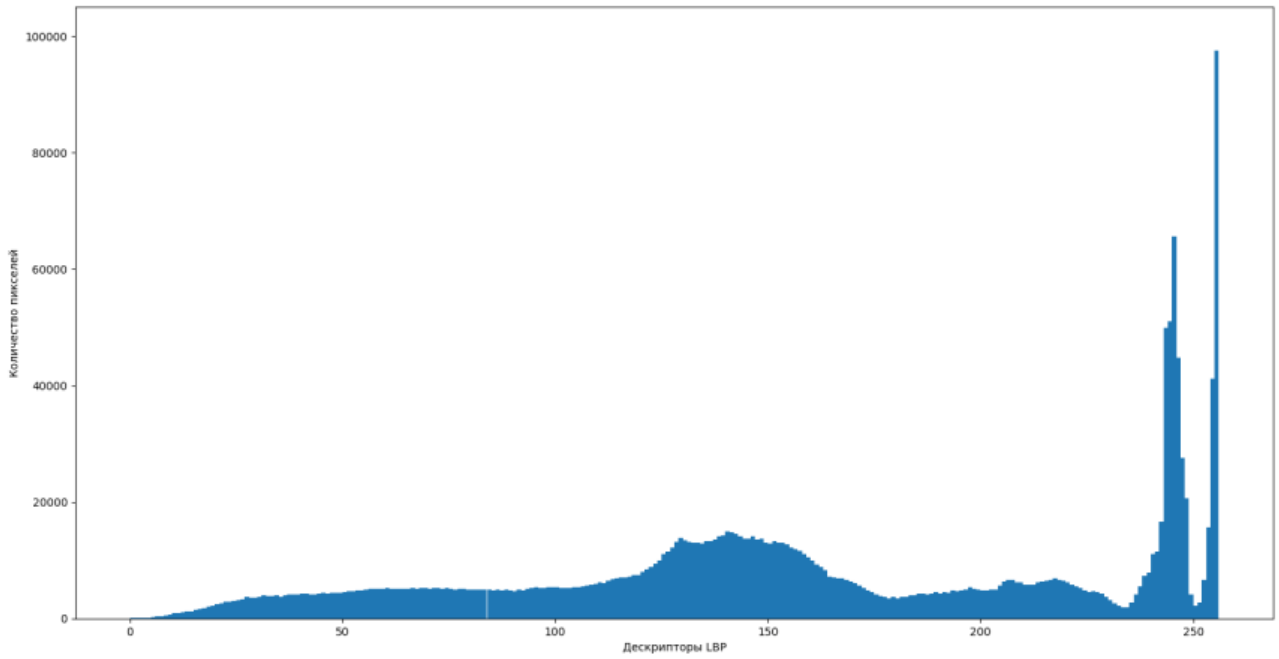


Рисунок 4.10 – Гистограмма коэффициентов DLBP пользователя информационной системы

Полученные гистограммы коэффициентов LBP пользователей информационной системы далее являются составной частью входного слоя искусственной нейронной сети [32].

Гистограмма ориентированных градиентов (Histogram of oriented gradients, HOG) – это коэффициент используемой функции в компьютерном зрении и обработке изображений с целью обнаружения объектов [26]. Методика подсчитывает появление градиентной ориентации в локализованных частях изображения. Данный метод сопоставим методу коэффициентов масштабно–инвариантного преобразования признаков и контекстов формы.

После фильтрации и предварительной обработки изображения пользователя информационной системы представляется векторное направление каждого пикселя ячейки в гистограмме градиентов HOG (Рисунок 4.11).

Для учета изменений освещения и контрастности силы градиента должны быть локально нормализованы, что требует группировки ячеек вместе в более крупные, пространственно связанные блоки. HOG представляет собой

конкатенированный вектор компонентов нормализованных гистограмм ячеек из всех областей блока.

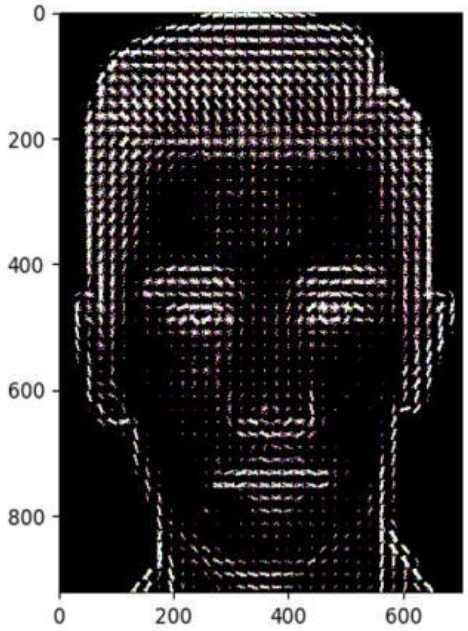


Рисунок 4.11 – Гистограмма градиентов HOG изображения пользователя информационной системы

Матрица градиентов является функцией углового расположения пар соседних элементов разрешения, а также функцией расстояния между ними. Для каждого такого направления введены матрицы распределения градиентов, содержащие относительные частоты наличия на изображении соседних элементов со значениями яркости элементов разрешения. Далее величина и направление каждого градиента будут представлены в виде двух матриц размерностью $8 \, x \, 8$ с углами в диапазоне от 0 до 180 градусов. Данные показатели сортируются в гистограмме, которая состоит из 9 бинов (Рисунок 4.12). Бины – это один из способов агрегирования точечных объектов для наблюдения закономерностей в мелких и крупных масштабах для повышения скорости обработки данных [85].

В итоге, получившиеся матрицы и будут составлять полный ансамбль матриц градиентов.

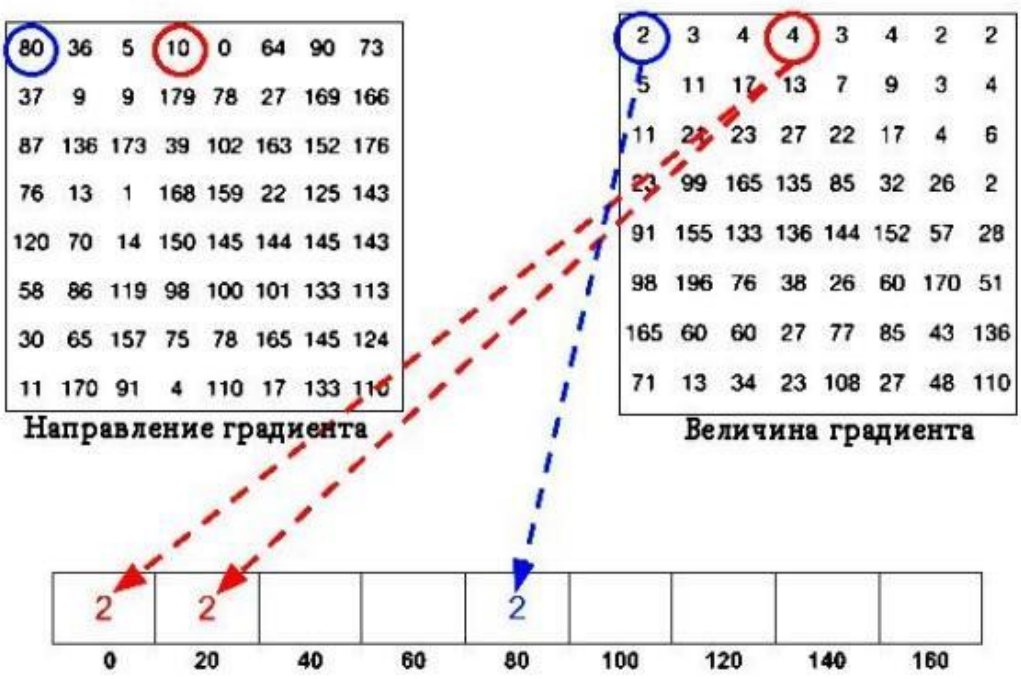


Рисунок 4.12 – Совмещение величины и направления градиента

Если угол больше 160 градусов, он находится между 160 и 180. Угол огибает, делая эквивалент 0 и 180. Например, пиксель с углом 165 градусов вносит пропорциональный вклад в бин 0 градусов и бин 160 градусов.

Полученные показатели всех пикселей изображения пользователя информационной системы в ячейках $8\,x\,8$ складываются для создания 9–биновой гистограммы (Рисунок 4.13).

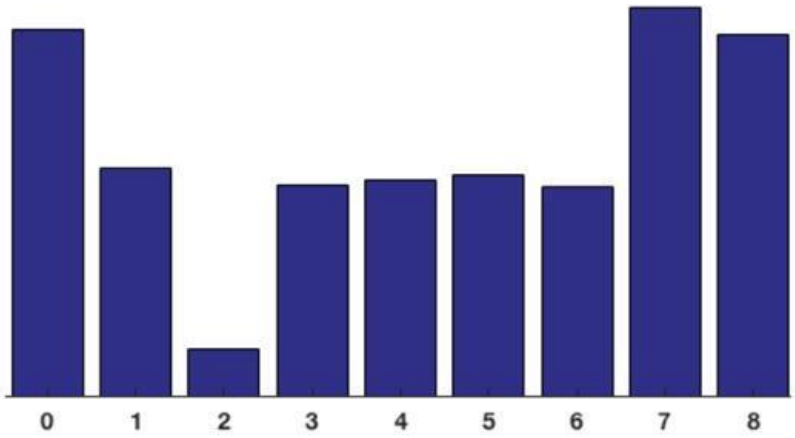


Рисунок 4.13 – Гистограмма показателей всех пикселей изображения пользователя информационной системы

Гистограмма имеет большой вес около 0 и 180 градусов [26]. Таким образом, градиенты изображения указывают либо вверх, либо вниз.

Для определения наиболее эффективного алгоритма идентификации личности приведены полученные результаты сравнения ранее рассмотренных методов на различных наборах данных. На основе проведенных экспериментов были выбраны следующие наборы данных:

– фронтальные изображения 1196 человек базы данных FERET;

– 194 фронтальных изображения людей в различных условиях освещения;

– сформированная база данных, состоящая из 600 изображений каждого пользователя системы.

К критериям оценки эффективности методов следует отнести частоту распознавания, которая вычисляется по следующей формуле [94]:

$$E = \frac{N_p}{N} \tag{97}$$

где $N_p$ – количество верно распознанных фрагментов, $N$ – количество распознаваемых фрагментов.

Таблица 4.1 – Полученные результаты сравнения методов модели распознавания пользователей информационных систем по изображению лица

| Метод | Результат распознавания, % | | |
|---|---|---|---|
| | Набор 1 | Набор 2 | Набор 3 |
| Метод Виолы–Джонса | 90,47 | 88,14 | 85,00 |
| Метод гибкого сравнения на графах | 86,96 | 81,96 | 75,00 |
| PCA | 87,94 | 88,05 | 89,00 |
| LBP | 92,05 | 92,75 | 94,00 |

Сравнение качества распознавания в зависимости от уровня предобработки изображений на примере DLBP и сгенерированного обучающего набора представлено в таблице 4.2.

Таблица 4.2 – Сравнение качества распознавания лиц в зависимости от предобработки

| Предварительная обработка | Точность, 1 чел. | Точность, 2 чел. | Средняя точность |
|---|---|---|---|
| Нет | 0.9103 | 0.9177 | 0.9140 |
| Вырезание лица | 0.9430 | 0.9663 | 0.9547 |
| Трансформация и вырезание лица | 0.9680 | 0.9790 | 0.9684 |

Сравнение качества распознавания пользователей на основе извлеченных признаков проводилось с использованием библиотек Python, состоящих из различных строительных блоков. Это сравнение позволяет оценить устойчивость систем распознавания к качеству предобработки изображений. Результаты сравнения представлены в таблице 4.3. Значения определялись с использованием характеристик центрального и графического процессоров.

Таблица 4.3 – Сравнение качества распознавания лиц различным составом конвейеров

| Библиотеки | Точность, 1 чел. | Точность, 2 чел. | Средняя точность |
|---|---|---|---|
| Openface, DLBP | 0.9240 | 0.9418 | 0.9329 |
| PCA, face align, keras FaceNet | 0.9493 | 0.9940 | 0.9717 |
| PCA, face align, pytorch | 0.9413 | 0.9768 | 0.9663 |
| PCA, face align, Dlib | 0.9310 | 0.9727 | 0.9518 |

| | | | |
|---|---|---|---|
| DLBP, face align, keras FaceNet | 0.9680 | 0.9894 | 0.9822 |
| DLBP, face align, pytorch | 0.9663 | 0.9857 | 0.9760 |
| DLBP, face align, Dlib | 0.9680 | 0.9790 | 0.9724 |

## 4.2. Результаты распознавания пользователей информационной системы по голосу

В ходе исследования была реализована архитектура нейронной сети Wav2vec и проведено обучения на различных наборах данных (обучающие выборки) (Рисунок 4.14) [64].

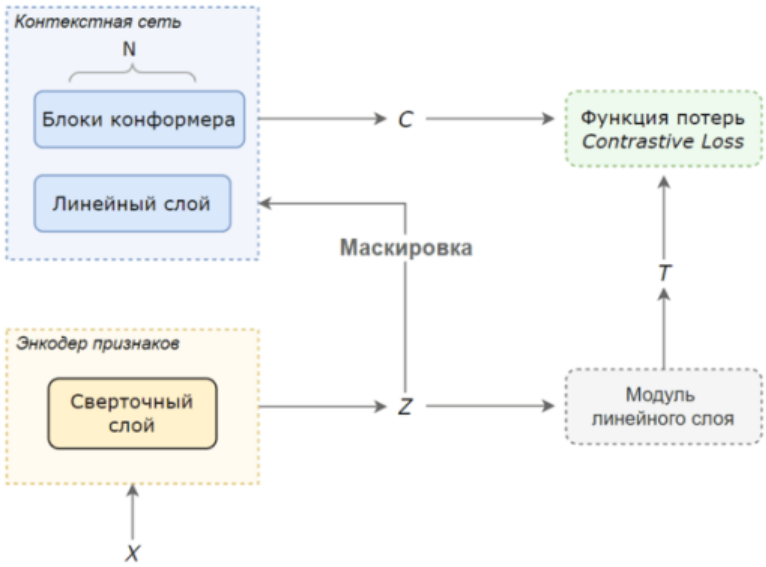


Рисунок 4.14 – Схема обучения нейронной сети Wav2vec для модели биометрической аутентификации по голосу

Для корректности оценки эффективности разработанных комплексов распознавания пользователей информационных систем по голосу приведены показатели ошибки разделения пользователей, показатели функции ошибок, скорости обучения и точности с применением датасета VoxCeleb для двух пользователей информационной системы (Рисунок 4.15, Рисунок 4.16, Рисунок 4.17).

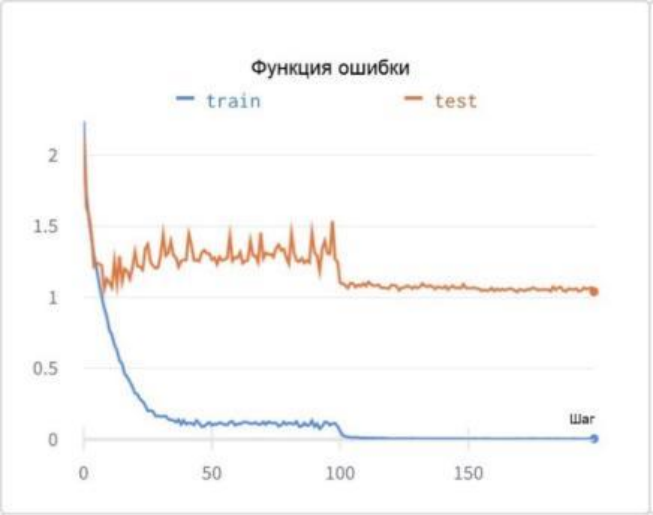


Рисунок 4.15 – Показатели функции ошибки сверточной нейронной сети Wav2vec на двух пользователей с применением датасета VoxCeleb

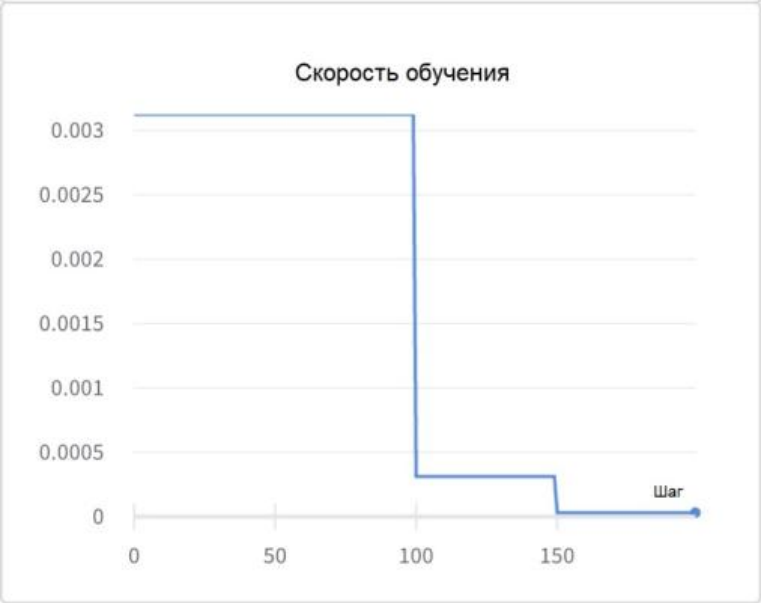


Рисунок 4.16 – Скорость обучения сверточной нейронной сети Wav2vec на двух пользователей с применением датасета VoxCeleb

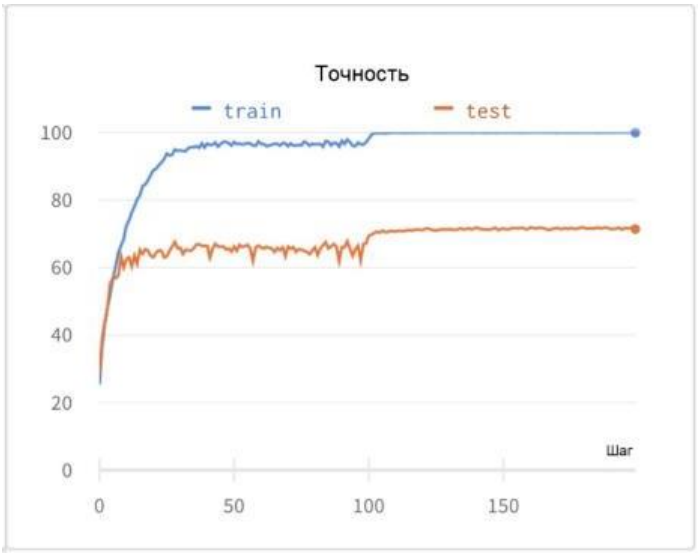


Рисунок 4.17 – Показатели точности обучения сверточной нейронной сети Wav2vec на двух пользователей с применением датасета VoxCeleb

Объем собственной обучающей выборки для каждой архитектуры нейронной сети по акустическим признакам составляет 450 аудиозаписей двух пользователей информационной системы. Длительность каждой аудиозаписи составляет 8, 15 и 25 секунд соответственно. Аудиозапись длительностью 8 секунд представлена в виде файла с записанным голосом, в котором производится счет от одного до пяти. В аудиозаписи длительностью 15 секунд производится счет от одного до десяти, а в аудиозаписи длительностью 25 секунд – от одного до двадцати.

## 4.3. Архитектура нейронной сети на основе конкатенации и экспериментальные исследования по распознаванию пользователей на ее основе

В данном исследовании использовалось объектно-ориентированное программирование (ООП) на языке программирования Python 3.11 (Рисунок 4.18 и Рисунок 4.19).


Рисунок 4.18 - UseCase-модель мультимодальной системы биометрической аутентификации


Рисунок 4.19 - Модель терминов мультимодальной системы биометрической аутентификации

При разработке программного обеспечения были применены следующие принципы ООП: инкапсуляция и наследование (Рисунок 4.20 и Рисунок 4.21).

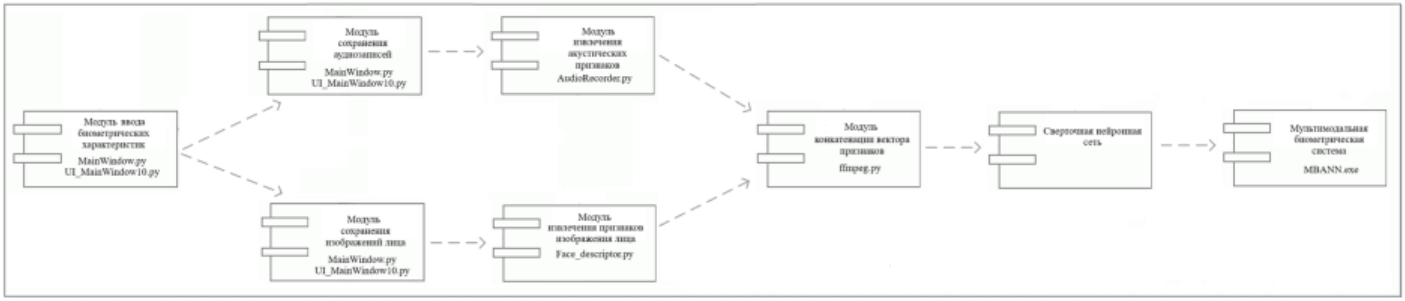


Рисунок 4.20 - Диаграмма компонентов мультимодальной системы биометрической аутентификации

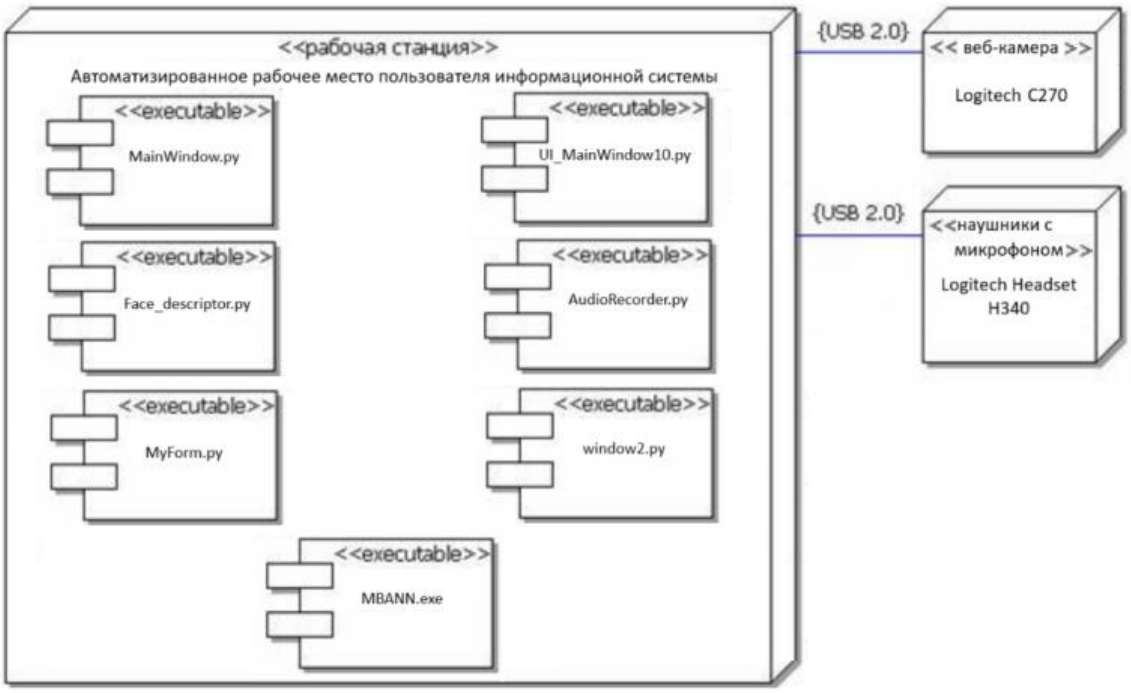


Рисунок 4.21 - Диаграмма развертывания мультимодальной системы биометрической аутентификации

Использовались такие библиотеки, как numpy, matplotlib, opencv, scimage, librosa, spafe, PyQt5. Программное обеспечение состоит из модуля ввода/вывода информации (Рисунок 4.22 и Рисунок 4.23).

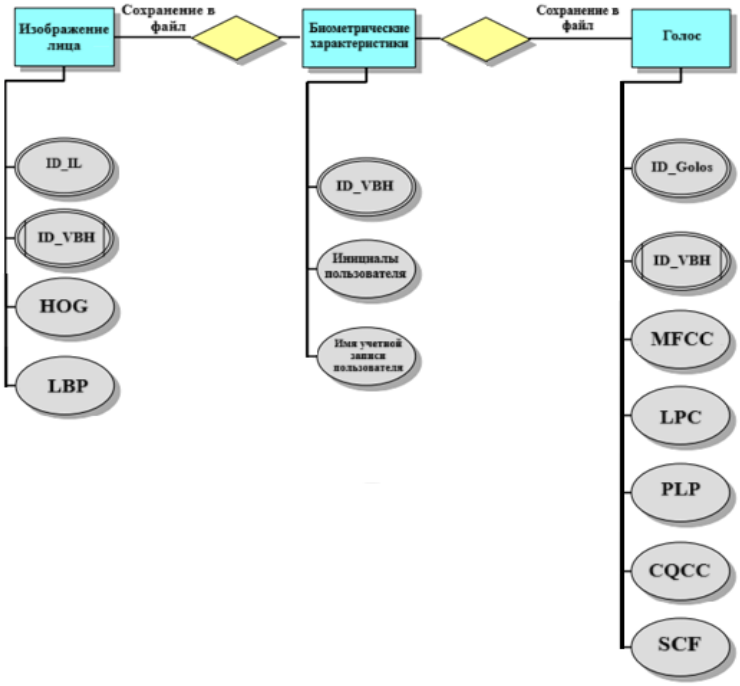


Рисунок 4.22 - Диаграмма развертывания мультимодальной системы биометрической аутентификации

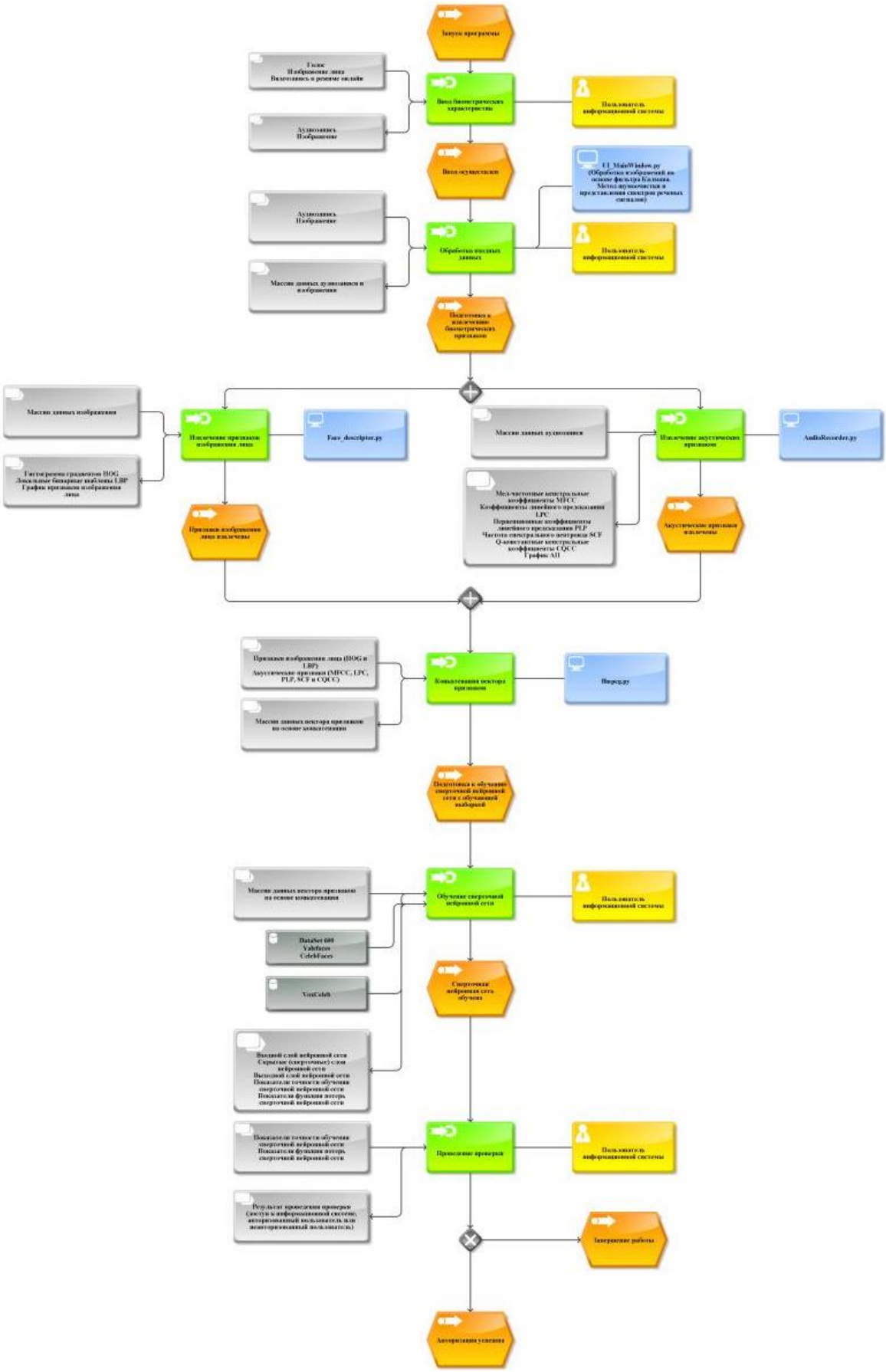


Рисунок 4.23 - EPC-модель мультимодальной системы биометрической аутентификации

Для программной реализации и применения мультимодальной системы биометрической аутентификации пользователей информационной системы, на основе нейронных сетей с различными биометрическими признаками разработана блок-программы и структура принятия решения (Рисунок 4.24, Рисунок 4.25, Рисунок 4.26). В результате анализа архитектур нейронных сетей при распознавании пользователей информационной системы по изображению лица и речевому сигналу формируются два вектора признаков одинаковой размерности. Далее векторы объединяются в результирующий вектор размера 512. При данном способе конкатенации векторов влияние каждой модальности на принятие решения является равнозначным.

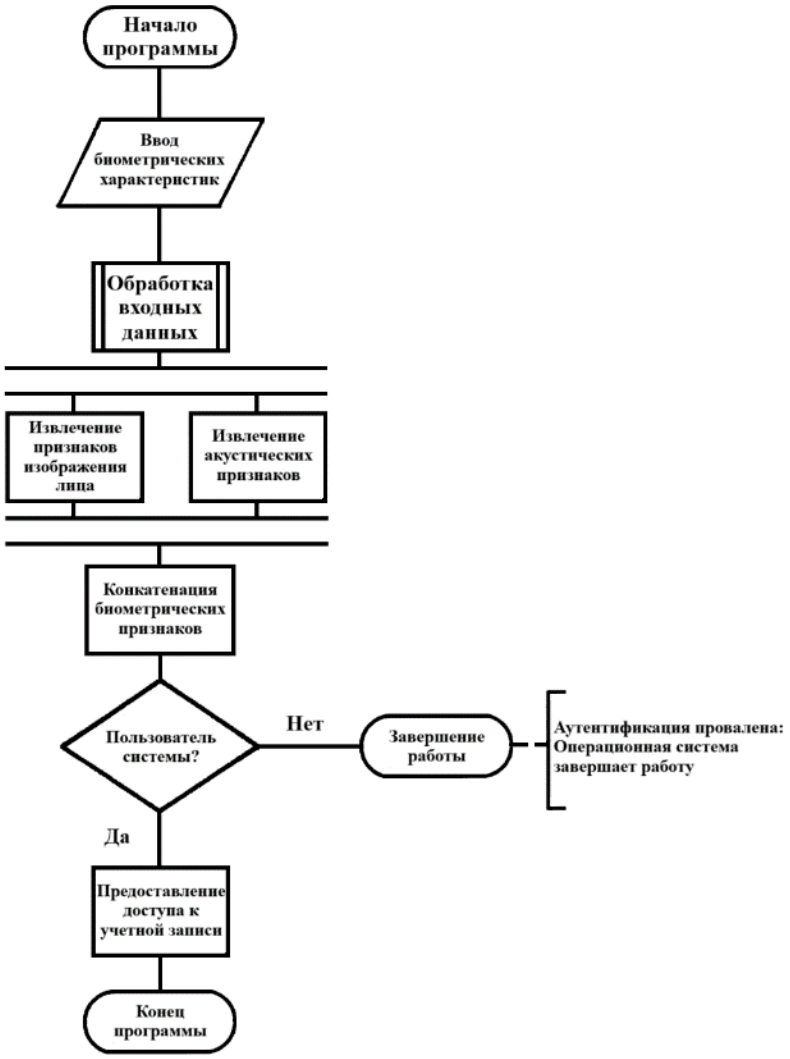


Рисунок 4.24 – Блок-схема программы мультимодальной биометрической системы аутентификации пользователей информационной системы

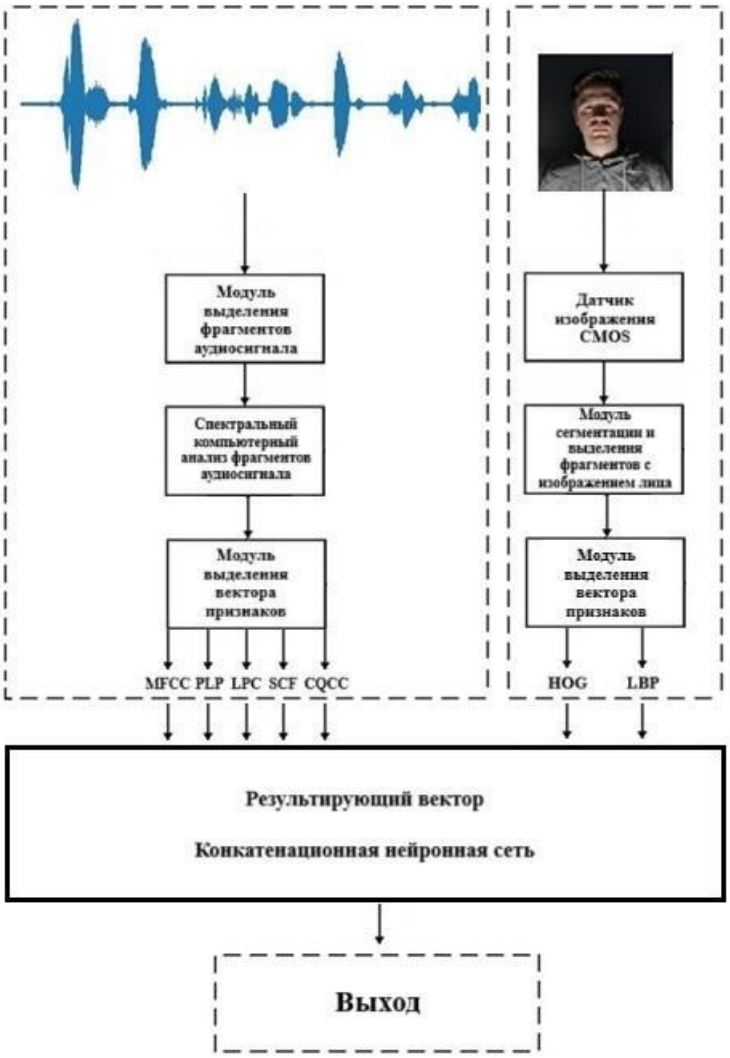


Рисунок 4.25 - Схема биометрической системы аутентификации пользователей информационной системы

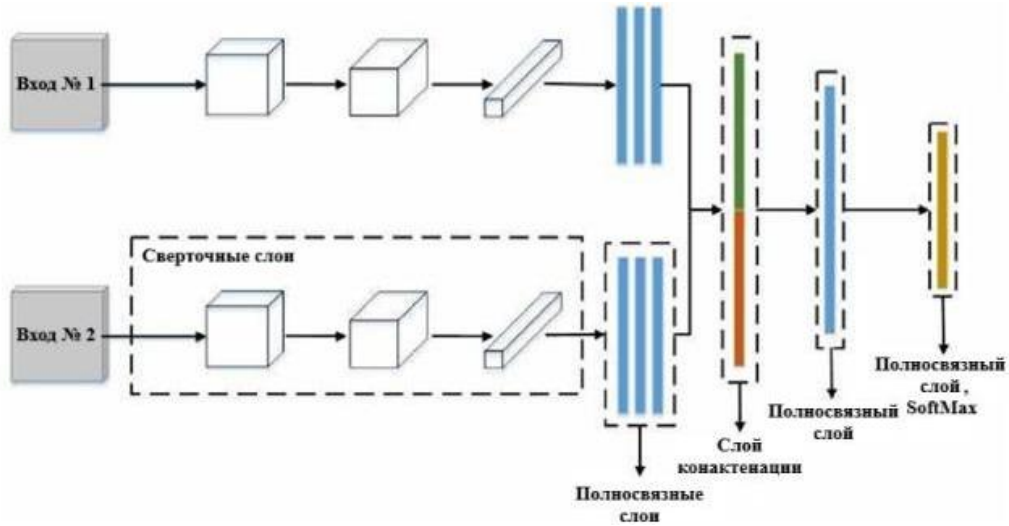


Рисунок 4.26 - Схема объединения признаков на основе конкатенации

## 4.4. Оценка эффективности распознавания пользователей на основе разработанного биометрического образа

По результатам обучения нейронных сетей были получены показатели эффективности работы программного комплекса системы биометрической аутентификации при распознавании пользователей по голосу на основе разработанного биометрического образа на различных наборах данных всех извлеченных акустических признаков (Таблица 4.4). В ходе исследования было выявлено, что для повышения точности распознавания пользователей информационной системы необходимо использовать все извлекаемые акустические признаки.

Таблица 4.4 – Оценка эффективности работы программного комплекса

| Набор данных | MFCC | | LPC | | PLP | | CQCC | | SCF | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $E_{spkr}(\%)$ | K | $E_{spkr}(\%)$ | K | $E_{spkr}(\%)$ | K | $E_{spkr}(\%)$ | K | $E_{spkr}(\%)$ | K |
| DataSet 100 | 8,26 | 0,862 | 7,60 | 0,873 | 9,42 | 0,845 | 7,52 | 0,793 | 8,34 | 0,853 |
| DataSet 300 | 8,01 | 0,82 | 7,99 | 0,832 | 8,87 | 0,839 | 7,49 | 0,769 | 8,1 | 0,821 |
| DataSet 450 | 7,67 | 0,79 | 7,8 | 0,801 | 8,34 | 0,821 | 7,21 | 0,742 | 7,99 | 0,8 |

После полученных результатов обучения на различных архитектурах нейронной сети с различными наборами обучающих выборок был проведен сравнительный анализ биометрических параметров (Рисунок 4.27) и приведено сравнение архитектур нейронных сетей в задаче распознавания пользователей информационной системы по голосу (Рисунок 4.28).

Сравнительный анализ биометрических параметров

| Характеристики | Сетчатка | Отпечатки пальцев | Радужная оболочка | Геометрия руки | Подпись | Лицо | Голос |
|---|---|---|---|---|---|---|---|
| Точность | Очень высокая | Высокая | Очень высокая | Высокая | Высокая | Высокая | Высокая |
| Простота использования | Низкая | Высокая | Средняя | Высокая | Высокая | Средняя | Высокая |
| Источники ошибок | Очки | Сухость, загрязнение, возраст | Освещение | Травма руки, возраст | Изменение подписи | Освещение, возраст, очки, волосы | Шум, простуда |
| Долговременная стабильность | Высокая | Высокая | Высокая | Средняя | Средняя | Средняя | Средняя |
| Приемлемость для пользователей | Средняя | Средняя | Средняя | Средняя | Высокая | Средняя | Высокая |

Рисунок 4.27 – Сравнительный анализ биометрических параметров

Результат идентификации личности в ходе проведения экспериментов

| № п/п | Условия тестирования | МСП | СНС |
|---|---|---|---|
| 1 | Здоровье человека в норме, помехи окружающей среды отсутствуют | + | + |
| 2 | Здоровье человека в норме, эксперимент проводился в условиях уличного шума | + | + |
| 3 | Здоровье человека в норме, эксперимент проводился около оживленной проезжей дороги | - | + |
| 4 | Здоровье человека в норме, эксперимент проводился вблизи шумной стройки | - | - |
| 5 | В голосе присутствует хрипота из-за болезни человека, помехи окружающей среды отсутствуют | + | + |
| 6 | Осипший голос из-за болезни, эксперимент проводился около оживленной проезжей дороги | - | - |
| 7 | Незначительная картавость речи, помехи окружающей среды отсутствуют | + | + |

Рисунок 4.28 – Сравнение архитектур нейронных сетей в задаче распознавания пользователей информационной системы по голосу

## Выводы и результаты по четвертой главе

В рамках данного исследования разработана архитектура сверточной нейронной сети на основе конкатенации вектора признаков изображений лица и голоса.

Сформирована обучающая выборка извлеченных биометрических признаков. Проведен сравнительный анализ представленных архитектур и реализован процесс обучения нейронных с применением составленного программного кода.

Приведены основные показатели эффективности, которые отражают функциональность архитектуры сверточной нейронной сети на основе конкатенации вектора признаков.

Разработан программный модуль распознавания пользователей информационных систем по биометрическим характеристикам, который позволит защитить биометрические персональные данные пользователей информационной системы от несанкционированного доступа.

Предложены принципы, решения и алгоритмы предварительной обработки изображения лица и голоса, обучения сверточной нейронной сети в составе мультимодальной биометрической системы аутентификации, позволяющие минимизировать возможность синтеза изображений и голоса (дипфейк). Сверточная нейронная сеть на основе конкатенации векторов признаков (изображение лица и голос) достигает показателей точности обучения 99,31 % и функции потерь = 0,032, что позволяет повысить

надежность процедуры распознавания пользователей информационной системы.

Разработаны мультимодальная система биометрической аутентификации пользователей информационной системы на основе нейронных сетей и программный комплекс данной системы, позволяющие осуществить разграничение доступа пользователей информационной системы.

# ЗАКЛЮЧЕНИЕ

Работа посвящена разработке мультимодальной системы биометрической аутентификации пользователей информационной системы на основе сверточной нейронной сети. Модели мультимодальной системы биометрической аутентификации пользователей информационной системы обладают уникальными наборами извлекаемых биометрических характеристик. Развитие инновационных средств защиты информации и наукоемких решений в рассматриваемой области является основополагающим аспектом, который направлен на повышение уровня защищенности данных и создание высоконадежных систем.

Процедуры идентификации и аутентификации пользователей являются одними из важнейших механизмов защиты современных информационных систем, реализуются на первичных этапах их обороны, в связи с чем данные механизмы наиболее часто подвергаются различным атакам со стороны злоумышленников. Стойкость ко взлому процедур идентификации и аутентификации субъектов доступа во многом определяет общий уровень защищенности всей информационной системы, поэтому качеству реализации данных процедур всегда уделяется особое внимание. В настоящее время биометрические системы становятся одними из наиболее перспективных средств аутентификации пользователей. Данные системы обеспечивают удобство аутентификации человека с одной стороны и высокий уровень безопасности с другой. В отличие от паролей и носителей информации, которые могут быть утеряны, украдены или скомпрометированы, биометрические системы основаны на человеческих уникальных характеристиках, которые являются неотъемлемой частью пользователя информационной системы. Это делает подделку или хищение аутентифицирующей информации практически невозможной.

Один из подходов к повышению качества работы биометрических систем, уменьшению количества ошибочных отказов и подтверждений,

устойчивости к подделкам, заключается в применении мультимодальности - использование для принятия решений нескольких биометрических характеристик, например, изображения лица и голоса человека. Однако, применение данного подхода на практике характеризуется рядом сложностей, требующих решения. В частности, необходимо разработать подходы к комбинированию различных биометрических признаков, к формированию итогового решения об аутентификации пользователя на основании нескольких биометрических характеристик. Также необходимо исследовать этапы предобработки биометрических характеристик для устранения возможных шумов.

Основная задача исследований в рассматриваемой предметной области состоит в высоконадежном распознавании личности (высоконадежной биометрической аутентификацией). На данный момент в области биометрических технологий наиболее актуальной проблемой является проблема обеспечения защищенности мультимодальных биометрических систем от несанкционированного доступа. К числу нерешенных задач следует отнести повышение надежности распознавания пользователей информационной системы на основе нейронных сетей, поддержание высокого уровня точности процедуры биометрической аутентификации и показателей достоверности.

В первой главе описана актуальность темы исследования, изучены основные нормативно–правовые документы, стандарты в области биометрической идентификации и аутентификации пользователей, а также проведен анализ научных публикаций в данной области. Рассмотрены особенности распознавания пользователей информационных систем по биометрическим признакам на основе мультимодальных и мультибиометрических технологий. Проведен анализ статистических показателей рынка биометрии.

Во второй главе рассмотрены основные алгоритмы распознавания пользователей информационных систем по изображению лица, применены

предварительная обработка и морфологические преобразования изображений лица, построены гистограммы локальных бинарных шаблонов и направленных градиентов. Разработана архитектура искусственной нейронной сети модели распознавания пользователей информационной системы по изображению лица и проведено обучение. Реализован модуль, предназначенный для противодействия методике синтеза изображения (дипфейк).

В третьей главе проведен анализ речевого сигнала, извлечены основные акустические признаки и изучены применяемые алгоритмы обработки аудиозаписей при прохождении систем аутентификации. Приведены результаты и показатели эффективности разработанной системы на основе искусственной нейронной сети. Рассмотрены отличия двух разработанных архитектур искусственных нейронных сетей для распознавания пользователей информационных систем по голосу.

В четвертой главе рассмотрены архитектура и алгоритм нейронной сети биометрической системы аутентификации на основе конкатенации. Описана структура и порядок действий мультимодальной системы биометрической аутентификации пользователей информационной системы. Представлены основные характеристики функционирования разработанного программного комплекса.

В ходе диссертационного исследования сформирован программный комплекс, направленный на решение проблемы распознавания пользователей информационных систем. Основные выводы и результаты диссертационной работы можно сформулировать следующим образом:

1. Разработана модель биометрической аутентификации пользователей информационной системы по изображению лица с применением фильтра Калмана, позволяющая в составе мультимодальной биометрической системы повысить помехоустойчивость алгоритма распознавания пользователей информационной системы. Данная модель использует предварительную обработку и морфологические преобразования

изображений лица. В рамках проведенных экспериментальных работ получены высокие показатели помехоустойчивости, выраженные в виде значения критерия среднеквадратичной ошибки MSE = 0,0019.

Разработана модель биометрической аутентификации пользователей информационной системы по голосу с применением метода шумоочистки и представления спектров речевых сигналов, позволяющая в составе мультимодальной биометрической системы уменьшить показатели ошибок первого и второго рода при распознавании пользователей информационной системы. Предложенная модель позволяет повысить точность распознавания пользователей по голосу и уменьшить показатели функции потерь на основании примененных средств обработки аудиосигнала. Показатель ошибки первого рода FRR = 0,11 % и показатель ошибки второго рода FAR = 0,01 %.

2. Предложены алгоритмы обучения сверточной нейронной сети в составе мультимодальной биометрической системы аутентификации, позволяющие минимизировать возможность синтеза изображений и голоса (дипфейк). Сверточная нейронная сеть на основе конкатенации векторов признаков (изображение лица и голос) достигает показателей точности обучения 99,31 % и функции потерь = 0,032, что позволяет повысить надежность процедуры распознавания пользователей информационной системы.

3. Разработан программный комплекс мультимодальной системы биометрической аутентификации пользователей информационной системы на основе сверточных нейронных сетей, который позволяет разграничить доступ пользователям информационной системы.

4. Осуществлена оценка эффективности разработанных решений и полученных результатов мультимодальной системы биометрической аутентификации пользователей информационной системы. В ходе проведения исследования мультимодальной системы биометрической аутентификации

пользователей информационной системы были получены показатели точности 98 %.

В рамках проведенных экспериментальных работ получены высокие показатели помехоустойчивости, выраженные в виде значения критерия среднеквадратичной ошибки MSE = 0,0019. Предложенная модель биометрической аутентификации пользователей информационной системы по голосу позволяет повысить точность распознавания пользователей по голосу и уменьшить показатели функции потерь на основании примененных средств обработки аудиосигнала. Показатель ошибки первого рода FRR = 0,11 % и показатель ошибки второго рода FAR = 0,01 %. Сверточная нейронная сеть на основе конкатенации векторов признаков (изображение лица и голос) достигает показателей точности обучения 99,31 % и функции потерь = 0,032, что позволяет повысить надежность процедуры распознавания пользователей информационной системы.

В ходе проведения исследования мультимодальной системы биометрической аутентификации пользователей информационной системы были получены показатели точности 98 %.

Модели и алгоритмы мультимодальной системы биометрической аутентификации пользователей информационной системы внедрены в автоматизированные рабочие места ООО «Информзащита» Уфа и ООО «ИТ Энигма Уфа», применена в процессах Координационного центра ФГБОУ ВО «Уфимский университет науки и технологий».

Разработанное решение и полученные результаты диссертационной работы принадлежат автору и являются уникальными.

**Перспективы дальнейшей разработки темы.** По теме диссертационного исследования в дальнейшем запланированы следующие мероприятия:

1. Биометрическая аутентификация по поведенческим характеристикам, что позволит противодействовать синтезу изображений (дипфейк).

# СПИСОК ЛИТЕРАТУРЫ

1. Об информации, информационных технологиях и о защите информации [Текст]: Федеральный закон от 27 июля 2006 г. № 149–ФЗ // Собрание законодательства РФ. – 2006. – № 13. – С. 12–13.

2. Об осуществлении идентификации и (или) аутентификации физических лиц с использованием биометрических персональных данных, о внесении изменений в отдельные законодательные акты Российской Федерации и признании утратившими силу отдельных положений законодательных актов Российской Федерации [Текст]: Федеральный закон от 29 декабря 2022 г. № 572–ФЗ // Собрание законодательства РФ. – 2022. – № 10. – С. 71–81.

3. О персональных данных [Текст]: Федеральный закон от 27 июля 2006 г. № 152–ФЗ // Собрание законодательства РФ. – 2006. – № 31 (1 я.). – ст. 3451.

4. Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных [Текст]: постановление Правительства РФ от 01 ноября 2012 г. № 1119 // Собрание законодательства РФ. – 2012. – № 45. – ст. 6257.

5. Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах [Текст]: приказ ФСТЭК России от 11 фев. 2013 г. № 17 // Российская газета. – 2013.

6. ГОСТ Р 58833–2020 «Защита информации. Идентификация и аутентификация. Общие положения» – Москва: Стандартинформ, 2020. – С. 11–16.

7. ГОСТ Р 52633.5–2011 «Защита информации. Техника защиты информации. Автоматическое обучение нейросетевых преобразователей биометрия–код доступа» – Москва: Стандартинформ, 2012. – 16 с.

8. ГОСТ Р 54411–2018 «Информационные технологии. Биометрия.

Мультимодальные и другие мультибиометрические технологии» – Москва: Стандартинформ, 2018. – 27 с.

9. ISO/IEC 24745:2011 Information Technology. Security Techniques. Biometric Information Protection – Technical Committee: ISO/IEC JTC 1/SC 27 Information Security, Cybersecurity and Privacy Protection, 2011, p. 50 – URL: https://www.iso.org/standard/52946.html (дата обращения: 08.06.2023).

10. ISO/IEC 24761:2009 Information Technology. Security Techniques. Authentication Context for Biometrics – Technical Committee: ISO/IEC JTC 1/SC 27 Information Security, Cybersecurity and Privacy Protection, 2009, p. 50 – URL: https://www.iso.org/standard/52946.html (дата обращения: 08.06.2023).

11. ISO/IEC 19792:2009 Information Technology. Security Techniques. Security Evaluation of Biometrics – Technical Committee: ISO/IEC JTC 1/SC 27 Information Security, Cybersecurity and Privacy Protection, 2009, p. 37 – URL: https://www.iso.org/standard/52946.html (дата обращения: 08.06.2023).

12. NIST SP 800–63–3 «Digital Identity Guidelines».

13. NIST SP 800–63A «Digital Identity Guidelines: Enrollment and Identity Proofing».

14. NIST SP 800–63B «Digital Identity Guidelines: Authentication and Lifecycle Management».

15. Акилин Г.А., Грицкевич Е.В. Особенности имитационного моделирования информационных систем, использующих биометрическую идентификацию по лицу // Сборник статей по материалам международного научного конгресса «Интерэкспо Гео–Сибирь» – Новосибирск: СГУГиТ, 2019, С.61–65.

16. Анисимова, А. С. Интеллектуальная система биометрической аутентификации пользователя по динамической рукописной подписи / А. С. Анисимова, И. В. Аникин // Международный форум Kazan Digital Week–2022: Сборник материалов Международного форума, Казань, 21–24 сентября 2022 года / Под общей редакцией Р.Н. Минниханова. – Казань: Научный центр безопасности жизнедеятельности, 2022. – С. 280–285.

17. Арсентьев Д.А., Бирюкова Т.С. Метод гибкого сравнения на графах как алгоритм распознавания образов // Вестник МГУП имени Ивана Федорова. – 2015. – № 6. – С. 74–75.

18. Багров Н. Ю. Полуавтоматический метод сбора выборок для обучения алгоритма идентификации лиц / Н. Ю. Багров, А. С. Конушин, В. С. Конушин // Программирование. – 2019. – № 3. – С. 57-63.

19. Биометрическая идентификация. Подрядчики по количеству проектов внедрений (ИБ – Биометрическая идентификация). 2022 год / Tadviser. Государство. Бизнес. – URL: https://www.tadviser.ru/index.php/ИБ_–__Биометрическая_идентификация?cache=no&ptype= integrator#ttop (дата обращения: 08.06.2023).

20. Ван Лянпэн, Петросян О.Г. Распознавание лиц на основе классификации вейвлет признаков путем вейвлет нейронных сетей // Информатизация образования и науки. 2018, № 4 (40).

21. Васильев В. И., Ильясов Б. Г. Интеллектуальные системы управления. Теория и практика. Учебное пособие – М.: Радиотехника, 2009 г. – 388 с.

22. Вульфин А.М. Интеллектуальный анализ видеоданных в системе контроля соблюдения правил промышленной безопасности [Электронный ресурс] // Моделирование, оптимизация и информационные технологии. – 2020. – № 8(2). – С. 1–16. – Режим доступа: https://moit.vivt.ru/wpcontent/uploads /2020/05/Vulfin_2_20_1.pdf

23. Голеусов Я. А. О показателях эффективности систем биометрической аутентификации и идентификации // Актуальные проблемы авиации и космонавтики. 2016. № 12. – URL: https://cyberleninka.ru/article/n/o–pokazatelyah–effektivnosti–sistem–biometricheskoy–autentifikatsii–i–identifikatsii (дата обращения: 08.06.2023).

24. Головко В. А. Нейронные сети: обучение, организация и применение. Книга 4. Нейрокомпьютеры и их применение – Москва, ИПРЖР, 2001 г.

25. Гонсалес Р., Вудс Р. Цифровая обработка изображений / Пер. с англ. — М.: Техносфера, 2006. 1072 с.

26. Горбунов А.Л. Визуальная когерентность в дополненной реальности. Advanced Engineering Research (Rostov-on-Don). 2023;23(2):180-190.

27. Гринчук О.В., Цурков В.И. Обучение мультимодальной нейронной сети для определения подлинности изображений // Известия Российской академии наук. Теория и системы управления. 2020, № 4. С. 103–109.

28. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах: Учеб. пособие. – Новосибирск.: Изд-во НГТУ, 2003. – 352 с.

29. Гузаиров, М. Б. Аутентификация пользователей информационной системы по изображению лица / М. Б. Гузаиров, А. С. Исмагилова, Н. Д. Лушников // Моделирование, оптимизация и информационные технологии. – 2023. – Т. 11, № 4(43).

30. Гузаиров, М.Б. Конкатенация нейронных сетей в системе биометрической аутентификации пользователей компьютерной информационной системы / М.Б. Гузаиров, А.С. Исмагилова, Н.Д. Лушников // Инженерный вестник Дона. – 2025. – № 5(125).

31. Девицына С.Н., Елецкая Т.А., Балабанова Т.Н., Гахова Н.Н. Разработка интеллектуальной системы биометрической идентификации пользователя // Научные ведомости. Серия: Экономика. Информатика. 2019, Т.46, №1. С.148–160.

32. Иванов В. В., Лубова Е. С., Черкасов Д. Ю. Аутентификация и авторизация // Проблемы Науки. 2017. № 2 (84). – URL: https://cyberleninka.ru/article/n/autentifikatsiya–i–avtorizatsiya (дата обращения: 08.06.2023).

33. Исмагилов, Р. Ф. Конструирование модели обучающей нейронной сети для биометрической многофакторной аутентификации пользователя информационной системы / Р. Ф. Исмагилов, Н. Д. Лушников, А. С.

Исмагилова // Вопросы защиты информации. – 2023. – № 1(140). – С. 19–23. – DOI 10.52190/2073–2600_2023_1_19.

34. Исмагилов, Р. Ф. Разработка приложения для мониторинга и выявления противоправного контента / Н. Д. Лушников, Р. Ф. Исмагилов // Информационные технологии обеспечения комплексной безопасности в цифровом обществе: сборник материалов V Всероссийской молодежной научно–практической конференции, Уфа, 20–21 мая 2022 года. – Уфа: Башкирский государственный университет, 2022. – С. 237–240.

35. Исмагилова, А. С. Алгоритм шифрования биометрических данных пользователя / А. С. Исмагилова, Н. Д. Лушников // Информационная безопасность: Сборник докладов Всероссийской Школы молодых ученых, Новосибирск, 14–18 ноября 2022 года. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2022. – С. 19–23. – DOI 10.55648/978–5–91434–080–0–2022–19–23.

36. Исмагилова, А. С. Комплексная биометрическая аутентификация пользователей информационной системы с применением нейронных сетей / А. С. Исмагилова, Н. Д. Лушников // Инженерный вестник Дона. – 2024. – № 1(109). – С. 178-188.

37. Исмагилова, А. С. Многофункциональное ПО для защиты учетных записей пользователей с использованием биометрических технологий / А. С. Исмагилова, Н. Д. Лушников // Защита информации. Инсайд. – 2021. – № 2(98). – С. 28–31.

38. Исмагилова А.С., Андреев М.Ф., Лушников Н.Д. Обработка сетевых пакетов в ядре Linux для противодействия атакам типа «отказ в обслуживании» / А. С. Исмагилова, М.Ф. Андреев, Н. Д. Лушников // Сборник статей по материалам IV Международной научной конференции, посвящённой памяти доктора технических наук, профессора А.А. Тарасова и доктора технических наук, старшего научного сотрудника О.В. Казарина. – Москва, 2023. – С. 169 – 172.

39. Исмагилова, А. С. Программная реализация защиты от

несанкционированного доступа / А. С. Исмагилова, Н. Д. Лушников // Безопасность информационных технологий. – 2023. – Т. 30, № 1. – С. 81–91. – DOI 10.26583/bit.2023.1.06.

40. Караваев Д.А. Вейвлет–подобная архитектура комплекснозначной сверточной нейронной сети для синтеза комплексных сигналов // Вестник кибернетики. 2020, № 2. С. 20–31.

41. Кручинина Е.В. Видеоидентификация – ключ в мире адресных услуг // Системы безопасности. 2016, №6. С. 110–111.

42. Крылова И.Ю., Рудакова О.С. Биометрические технологии как механизм обеспечения информационной безопасности в цифровой экономике // Молодой ученый. 2018, № 45 (231). – С. 74–79.

43. Лакин Г.Ф. Биометрия. Учебное пособие. – М.: Высшая школа. – 1990. – 223 с.

44. Ложников П. С., Сулавко А. Е. Применение сетей квадратичных форм для распознавания субъектов по динамическим биометрическим образам // ОмГТУ. 2017. №4. – URL: https://cyberleninka.ru/article/n/primenenie–setey–kvadratichnyh–form–dlya–raspoznavaniya–subektov–po–dinamicheskim–biometricheskim–obrazam (дата обращения: 08.06.2023).

45. Ложников П. С. Распознавание пользователей в системах дистанционного образования: обзор // ОТО. 2001. № 2. – URL: https://cyberleninka.ru/article/n/raspoznavanie–polzovateley–v–sistemah–distantsionnogo–obrazovaniya–obzor (дата обращения: 08.06.2023).

46. Ложников П. С., Самотуга А. Е. Технология проверки целостности и аутентичности документов в гибридном документообороте // Известия ТулГУ. Технические науки. 2013. №3. – URL: https://cyberleninka.ru/article/n/tehnologiya–proverki–tselostnosti–i–autentichnosti–dokumentov–v–gibridnom–dokumentooborote (дата обращения: 08.06.2023).

47. Ложников П. С., Сулавко А. Е., Еременко А. В., Волков Д. А. Экспериментальная оценка надежности верификации подписи сетями

квадратичных форм, нечёткими экстракторами и персептронами // Информационно–управляющие системы. 2016. № 5. С. 73–85. doi:10.15217/issn1684–8853.2016.5.73.

48. Лушников, Н. Д. Видеоидентификация как средство защиты персональных устройств / Н. Д. Лушников // Информационные технологии обеспечения комплексной безопасности в цифровом обществе: сборник материалов IV Всероссийской молодежной научно–практической конференции с международным участием, Уфа, 21–22 мая 2021 года. – Уфа: Башкирский государственный университет, 2021. – С. 39–42. – DOI 10.33184/itokbco–2021–05–21.7.

49. Лушников, Н. Д. Система распознавания пользователей по извлекаемым признакам голоса с применением фильтра Калмана // Инженерный вестник Дона. – 2024. – № 2(110).

50. Лушников, Н. Д. Защита информационных ресурсов пользователей с помощью биометрической идентификации личности / Н. Д. Лушников // Этнополитический и религиозный экстремизм в России: социально–культурные истоки, угрозы распространения в информационной среде, методы противодействия: Сборник материалов Всероссийской молодежной научной школы–конференции, Уфа, 04–05 декабря 2020 года. – Уфа: ООО Издательство «Диалог», 2020. – С. 354–359.

51. Лушников, Н. Д. Обучение и создание весов нейронной сети с применением категориальной кросс–энтропии / Н. Д. Лушников, А. С. Исмагилова // Информационные технологии обеспечения комплексной безопасности в цифровом обществе: сборник материалов V Всероссийской молодежной научно–практической конференции, Уфа, 20–21 мая 2022 года. – Уфа: Башкирский государственный университет, 2022. – С. 30–32. – DOI 10.33184/itokbco–2022–05–20.6.

52. Лушников, Н. Д. Организация идентификации личности при помощи нейросетевых технологий / Н. Д. Лушников, А. С. Исмагилова // Актуальные проблемы прикладной математики, информатики и механики: сборник трудов

Международной научной конференции, Воронеж, 07–09 декабря 2020 года / ФГБОУ ВО «Воронежский государственный университет». – Воронеж: Научно–исследовательские публикации, 2021. – С. 575–581.

53. Лушников, Н. Д. Особенности голосовой идентификации в многофункциональном программном обеспечении с использованием нейронных сетей / Н. Д. Лушников, А. С. Исмагилова // Теория и практика обеспечения информационной безопасности: Сборник научных трудов по материалам всероссийской научно–теоретической конференции, Москва, 03 декабря 2021 года. – Москва: Московский технический университет связи и информатики, 2021. – С. 98–102.

54. Лушников, Н. Д. Особенности идентификации пользователя по видео в режиме онлайн / Н. Д. Лушников, А. С. Исмагилова // Теоретические и прикладные вопросы реализации проектов в области информационной безопасности: Материалы межвузовской научно–теоретической конференции (в рамках Сибирского форума «Информационная безопасность – 2021»), Новосибирск, 29 – 03 ноября 2021 года / Под редакцией А.В. Ефимова, Т.И. Монастырской, И.В. Балабан. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2021. – С. 84–89.

55. Малков, А. А., Кротов, Л. Н., Кротова, Е. Л. Численные методы анализа экспертных оценок в системах социальной аутентификации // Системы управления и информационные технологии, Воронеж, издательство «Научная книга», № 1 (47), 2012. С. 62–65.

56. Малыгина, Е. А., Иванов, А. И., Урнев, И. В. Обеспечение безопасности биометрических данных для систем контроля доступа к «облачным сервисам» // Сборник статей 32 Международной научно–технической конференции «Проблемы автоматизации и управления в технических системах». Том 1, издательство Пензенский государственный университет, 2017 г. С. 119–121.

57. Мамаев В. Многофакторная биометрическая идентификация // Системы безопасности. 2017, №5. С. 78–79.

58. Марпл С. Л. Цифровой спектральный анализ и его приложения. – М.: Мир, 1990.

59. Машкина, И. В. Автоматизация экспертного аудита информационной безопасности на основе использования искусственной нейронной сети / И. В. Машкина, А. Ю. Сенцова // Безопасность информационных технологий. – 2014. – Т. 21, № 2. – С. 65-70.

60. Немков Р.М. Исследование сверточной нейронной сети, обученной с помощью метода применения нестандартных рецептивных полей при распознавании изображений // Известия Южного федерального университета. 2015, № 7 (168).

61. Осовский, С. Нейронные сети для обработки информации: учебное пособие / С. Осовский – М.: Телеком, 2017.

62. Пименова М. Б. Применение фильтра Калмана в задачах трекинга воздушных объектов / М. Б. Пименова // Политехнический молодежный журнал. – 2019. – № 12(41). – С. 1-9.

63. Пчеловодова Н. Российский биометрический рынок в 2019–2022 годах. Результаты масштабного исследования J’son & Partners Consulting // Системы безопасности. 2019, № 2. С. 88–91.

64. Радиоавтоматика: Учеб. пособие для студ. вузов спец. «Радиотехника»/В. А. Бесекерский, А. А. Елисеев, А. В. Небылов и др.; Под ред. В. А. Бесекерского.— М.: Высш. шк., 1985. — 271 с.

65. Рахманенко, И. А. Автоматическая верификация диктора по произвольной фразе с применением сверточных глубоких сетей доверия / И. А. Рахманенко, А. А. Шелупанов, Е. Ю. Костюченко // Компьютерная оптика. – 2020. – Т. 44, № 4. – С. 596-605.

66. Рабинер Л. Р., Шафер Р. В. Цифровая обработка речевых сигналов. – М.: Радио и связь, 1981. – 496 с.

67. Сабанов А. Г. Аутентификация и системы разграничения логического доступа: концепция оценки доверия к результатам / Сабанов А. Г. // Защита информации. Инсайд. – 2021. – № 2. – С. 10–17.

68. Сабанов А. Г., Шубинский И. Б. Метод анализа технологических рисков первичной идентификации субъектов доступа / Сабанов А. Г., Шубинский И. Б. // Защита информации. Инсайд. – 2020. – № 3. – С. 57–61.

69. Свидетельство о государственной регистрации программы для ЭВМ № 2021614672 Российская Федерация. Аутентификация учетных записей пользователей с помощью биометрических технологий: № 2021613387: заявл. 15.03.2021: опубл. 29.03.2021 / Н. Д. Лушников, А. С. Исмагилова; заявитель федеральное государственное бюджетное образовательное учреждение высшего образования «Башкирский государственный университет».

70. Свидетельство о государственной регистрации программы для ЭВМ № 2023617147 Российская Федерация. Программное обеспечение для выявления противоправного контента: № 2023616049: заявл. 28.03.2023: опубл. 05.04.2023 / Р. Ф. Исмагилов, Н. Д. Лушников, А. С. Исмагилова; заявитель федеральное государственное бюджетное образовательное учреждение высшего образования «Уфимский университет науки и технологий».

71. Свидетельство о государственной регистрации программы для ЭВМ № 2020660303 Российская Федерация. Управление доступом при помощи нейронных сетей: № 2020618972: заявл. 12.08.2020: опубл. 01.09.2020 / Н. Д. Лушников, А. С. Исмагилова; заявитель федеральное государственное бюджетное образовательное учреждение высшего образования «Башкирский государственный университет».

72. Сердюк В. А. Организация и технологии защиты информации. Обнаружение и предотвращение информационных атак в автоматизированных системах предприятий [Текст]: учеб, пособие / В. А. Сердюк; Гос. ун–т – Высшая школа экономики. – М.: Изд. дом Гос. ун–та – Высшей школы экономики, 2011. – 572 с. – ISBN 978–5–7598–0698–1.

73. Сидоренко И. А., Кускова П. А. О спектральном анализе фонем с использованием звуковых редакторов [Текст] / Научные ведомости БелГУ, серия История. Политология. Экономика. Информатика. 2013, № 22 (165) – с.

246–250.

74. Сирота А. А., Иванков А. Ю. Блочные алгоритмы обработки изображений на основе фильтра Калмана в задаче построения сверхразрешения // Компьютерная оптика. – 2014. – Т. 38, № 1. – С. 118–126.

75. Сорокин В. Н. Распознавание личности по голосу: аналитический обзор [Текст] / В. Н. Сорокин, В. В. Вьюгин, А. А. Тананыкин // Информационные процессы. Том 12, №1. Институт проблем передачи информации, Российская академия наук, Москва. – 2012. – стр. 1–30.

76. Сорокин В. Н. Синтез речи – Наука, Москва, 1992 г.

77. Сорокин В. Н. Теория речеобразования – М.: Радио и связь, 1985 г. – 312 с.

78. Стругайло В. В. Обзор методов фильтрации и сегментации цифровых изображений / В. В. Стругайло // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. – 2012. – № 5. – С. 270-281.

79. Тропченко А.А., Тропченко А.Ю. Нейросетевые методы идентификации человека по изображению лица // Известия высших учебных заведений. Приборостроение. 2012, Т.55, №10.

80. Фисенко В.Т., Фисенко Т.Ю., Компьютерная обработка и распознавание изображений: учеб. пособие. – СПб: СПбГУ ИТМО, 2008. – 192 с.

81. Хайкин С. Нейронные сети: полный курс. 2-е изд.: пер. с англ. М.: И.Д. Вильямс, 2006. 1104 с.

82. Шаньгин В. Ф. Информационная безопасность компьютерных систем и сетей, учебное пособие. – М.: издательский дом «Форум»: Инфра–М, 2011. – 416 с.

83. Шелупанов А.А. Идентификация и аутентификация в цифровом мире: монография / А.Г. Сабанов, А.А. Шелупанов. – М.: Горячая линия – Телеком, 2022 – 356 с.: ил. – ISBN 978–5–9912–0976–2. – (Серия «Технологии доверенного взаимодействия»).

84. Щербань И. В., Доброходский В. В., Ефименко А. А. Online–

программа аутентификации, основанная на оконном преобразовании Фурье речевых фраз пользователя // Символ науки. 2016. № 6–1. URL: https://cyberleninka.ru/article/n/online–programma–autentifikatsii–osnovannaya–na–okonnom–preobrazovanii–furie–rechevyh–fraz–polzovatelya (дата обращения: 08.06.2023).

85. Anouar Ben Khalifa, Sami Gazzah, Najoua ESSOUKRI BEN AMARA. Adaptive Score Normalization: A Novel Approach for Multimodal Biometric Systems. – January 2013.

86. A. Rossand, A. K. Jain. Information Fusion in Biometrics // Proc. of AVBPA : conference. — June 2001. — P. 354—359.

87. A. S. Ismagilova and N. D. Lushnikov, «Learning Neural Network for Multifactor Authentication Using Biometric Technologies», IEEE, 2022 4th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), 2022, pp. 416–420, doi: 10.1109/SUMMA57301.2022.9973920.

88. A. Jain, K. Nandakumar, A. Ross. Score normalization in multimodal biometric systems // Pattern Recognition, Volume 38 Issue 12: journal. – December 2005. – P. 2270–2285.

89. Carey M., Parris E., Lloyd–Thomas H., Bennett S. (1996). Robust prosodic features for speaker identification. In: Proc. Intemat. Conf, on Spoken Language Processing (ICSLP), 1800–1803.

90. Deller J., Hansen J., Proakis J. (2000). Discrete–Time Processing of Speech Signals, second ed. IEEE Press, New York.

91. D. Jagadiswary, D. Saraswady. Biometric Authentication Using Fused Multimodal Biometric // Procedia Computer Science 85: journal. – June 2016. – P. 109–116.

92. Farrell K., Mammone R., Assaleh K. (1994). Speaker recognition using neural networks and conventional classifiers. IEEE Trans. Speech Audio Process., v.2, N. 1, 194– 205.

93. G. Doddington, W. Liggett, A. Martin, M. Przybocki and D.

Reynolds. Sheeps, goats, lambs and wolves: a statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation // Proc. of ICSLD 98: conference. – 1998.

94. Hsiao–Chuan Wang, Jyh–Min Cheng. A study on model–based equal error rate estimation for automatic speaker verification // INTERSPEECH: conference. – 2004.

95. Huang X., Acero A., Hon H.–W. (2001). Spoken Language Processing: a Guide to Theory, Algorithm, and System Development. Prentice–Hall, New Jersey.

96. Itoh K. (1992). Perceptual analysis of speaker identity. In: Saito, S. (Ed.), Speech Science and Technology. IOS Press, 133–145.

97. J. J. Hopfield «Neural networks and physical systems with emergent collective computational abilities», Proceedings of National Academy of Sciences, vol. 79 no. 8 pp. 2554–2558, April 1982. PNAS Reprint (Abstract) PNAS Reprint (PDF).

98. Jialiang Penga, Ahmed A. Abd El–Latifb, Qiong Li, Xiamu Niuc. Multimodal biometric authentication based on score level fusion of finger biometrics // Optik: journal. – December 2014. – P. 6891–6897.

99. Kohonen, T. (1989/1997/2001), Self–Organizing Maps, Berlin – New York: Springer–Verlag. First edition 1989, second edition 1997, third extended edition 2001, ISBN 0–387–51387–6, ISBN 3–540–67921–9.

100. Kuwabara H., Sagisaka Y. (1995). Acoustic characteristics of speaker individuality: Control and Conversion. Speech Communication, v.16, 165–173.

101. L. Latha, Sangarappan Thangasamy. Robust Way of Multimodal Biometric Score Normalization // Journal of Applied Security Research. – January 2012.

102. Lavner Y., Gath I., Rosenhouse J. (2000). The effects of acoustic modifications on the identification of familiar voices speaking isolated vowels. Speech Communication v.30, 9–26.

103. Lawrence R. Rabiner and Biing–Hwang Juang. Fundamentals of Speech Recognition. – Prentice Hall, 1993. – 496 p. – ISBN 978–0130151575.

104. Marcos Faundez–Zanuy. Biometric security technology // IEEE Aerospace and Electronic Systems Magazine: journal. – July 2006.

105. M. Indovina, U. Uludag, R. Snelick, A. Mink, A. K. Jain. Multimodal biometric authentication methods: A COTS approach // Proc. of Workshop on Multimodal User Authentication: workshop. – 2003. – P. 99–106.

106. Markel J., Oshika B., Gray Jr. A.H. (1977). Long–term feature averaging for speaker recognition. IEEE Trans. Acoustics, Speech, Signal Process., v.25, N4, 330– 337.

107. Massimiliano Todisco, Hector Delgado, Nicholas Evans, Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. In: Computer Speech & Language, v. 45, 2017, p. 516–535, ISSN 0885–2308, https://doi.org/10.1016/j.csl.2017.01.001.

108. Miyajima C., Watanabe H., Kitamura T., Katagiri S. (1999). Discriminative feature extraction – Optimization of Mel–cepstral features using second–order all–pass warping function. Proc. EUROSPEECH, II–779–I–782.

109. Neyman, J.; Pearson, E. S. (1933-02-16). On the problem of the most efficient tests of statistical hypotheses. Phil. Trans. R. Soc. Lond. A. 231 (694–706): 289–337.

110. Nolan F. (1983). The Phonetic Bases of Speaker Recognition. Cambridge University Press, Cambridge.

111. Ojala T., Pietikainen M., Harwood D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994, vol. 1, pp. 582–585.

112. Ojala T., Pietikainen M., Harwood D. A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition, 1996, vol. 29, pp. 51–59.

113. Patterson R. D., Holdsworth J. (1996). A functional model of neural activity patterns and auditory images. Advances in Speech, Hearing and Language Processing, v. 3, 547–563.

114. Pekhovsky T., Oparin 1. Maximum Likelihood Estimations for Session Independent Speaker Modeling // SPECOM–2009. Proc. XIII Intern. Conf. «Speech and Computer». St.–Petersburg, 2009. P. 267–270.

115. Prasanna S. R. M., Govind D. (2010). Analysis of Excitation Source Information in Emotional Speech. Interspeech, 781–784.

116. Reddy M. S. H., Prahallad K., Gangashetty S.V., Yegnanarayana B. (2010). Significance of Pitch Synchronous Analysis for Speaker Recognition using AANN Models. Interspeech, 669–672.

117. Rosenberg A., Siohan O., Parthasarathy S. (2000). Small group speaker identification with common password phrases. Speech Communication, v. 31, 131–140.

118. R. Snelick, M. Indovina, J. Yen, A. Mink. Multimodal Biometrics: Issues in Design and Testing // Proc. of the 5th International Conference on Multimodal Interfaces (ICMI 2003): conference. – November 2003.

119. R. Parkavi, K. R. Chandeesh Babu, J. Ajeeth Kumar. Multimodal Biometrics for user authentication // 2017 11th International Conference on Intelligent Systems and Control (ISCO): conference. – January 2017.

120. Sayed M. Performance of Convolutional Neural Networks for Human Identification by Gait Recognition // Journal of Artificial Intelligence. – 2018. – V. 11. – P. 30–38.

121. Sonmez K., Shriberg E., Heck L., Weintraub M. (1998). Modeling dynamic prosodic variation for speaker verification. In: Proc. Intemat. Conf, on Spoken Language Processing (ICSLP 1998), 3189–3192.

122. Sonmez M., Heck L., Weintraub M., Shriberg E. (1997). A lognormal tied mixture model of pitch for prosody–based speaker recognition. In: Proc. Fifth European Conf, on Speech Communication and Technology (Eurospeech), 1391–1394.

123. Soyuj Kumar Sahoo, Tarun Choubisa & S. R. Mahadeva Prasanna. Multimodal Biometric Person Authentication: A Review // IETE Technical Review: journal. – Sep 2014. – P. 54–75.

124. S. N. Garg, R. Vig, S.Gupta3. Multimodal Authentication System: An Overview Multimodal Authentication System: An Overview // International Science Press. – January 2012.

125. Takemoto H., Adachi S., Kitamura T., Mokhtari P., Honda K. (2006). Acoustic roles of the laryngeal cavity in vocal tract resonance. J. Acoust. Soc. Am., v.120, 2228–2239.

126. Viola P., Jones M. J. Robust real-time face detection // International Journal of Computer Vision. 2004. Vol. 57, no. 2, pp. 137-154.

127. Youssef Elmir, Zakaria Elberrichi, Reda Adjoudj. Score level fusion based multimodal biometric identification (Fingerprint & voice // 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT): conference. – March 2012.

128. Yongjin Lee, Kyunghee Lee, Hyungkeun Jee, Younhee Gil, Wooyong Choi, Dosung Ahn, Sungbum Pan. Fusion for Multimodal Biometric Identification // Audio– and Video–Based Biometric Person Authentication. – 2005. – P. 1071–1079.

**Приложение А: Свидетельства о государственной регистрации программы для ЭВМ и электронных ресурсов.**

РОССИЙСКАЯ ФЕДЕРАЦИЯ

# СВИДЕТЕЛЬСТВО

**о государственной регистрации программы для ЭВМ**

**№ 2020660303**

**Управление доступом при помощи нейронных сетей**

Правообладатель: ***федеральное государственное бюджетное образовательное учреждение высшего образования «Башкирский государственный университет» (RU)***

Авторы: ***Лушников Никита Дмитриевич (RU),***
***Исмагилова Альбина Сабирьянова (RU)***

Заявка № **2020618972**
Дата поступления **12 августа 2020 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***01 сентября 2020 г.***

*Руководитель Федеральной службы*
*по интеллектуальной собственности*

*Г.П. Ивлиев*

РОССИЙСКАЯ ФЕДЕРАЦИЯ

# СВИДЕТЕЛЬСТВО

**о государственной регистрации программы для ЭВМ**

**№ 2021614672**

**Аутентификация учетных записей пользователей с помощью биометрических технологий**

Правообладатель: ***федеральное государственное бюджетное образовательное учреждение высшего образования «Башкирский государственный университет» (RU)***

Авторы: ***Лушников Никита Дмитриевич (RU), Исмагилова Альбина Сабирьяновна (RU)***

Заявка № **2021613387**

Дата поступления **15 марта 2021 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***29 марта 2021 г.***

*Руководитель Федеральной службы по интеллектуальной собственности*

*Г.П. Ивлиев*

РОССИЙСКАЯ ФЕДЕРАЦИЯ

# СВИДЕТЕЛЬСТВО

**о государственной регистрации программы для ЭВМ**

**№ 2023686833**

**Биометрическая аутентификация пользователей информационной системы на основе искусственных нейронных сетей**

Правообладатели: ***Лушников Никита Дмитриевич (RU), Исмагилова Альбина Сабирьяновна (RU)***

Авторы: ***Лушников Никита Дмитриевич (RU), Исмагилова Альбина Сабирьяновна (RU)***

Заявка № **2023685836**
Дата поступления **22 ноября 2023 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***08 декабря 2023 г.***

*Руководитель Федеральной службы по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 429eb6a0fe3653164aaf96f83b73b4aa7
Владелец Зубов Юрий Сергеевич
Действителен с 10.05.2023 по 02.08.2024

*Ю.С. Зубов*

# Приложение Б: Акты внедрения.

«УТВЕРЖДАЮ»
Директор Координационного центра
_____ / Айдарбеков А. М.
«19» _____ 2025 г.

АКТ ВНЕДРЕНИЯ
о внедрении результатов диссертационной работы
Лушникова Никиты Дмитриевича на тему «Модели и алгоритмы мультимодальной биометрической аутентификации на основе сверточной нейронной сети»

Комиссия в составе: директор Координационного центра, Айдарбеков Айрат Минетдинович; заместитель директора Координационного центра, Великжанин Виктор Валентинович; ведущий специалист Координационного центра, Нурмухаметов Данир Расулевич, составила настоящий акт о том, что следующий результат диссертационной работы Лушникова Н.Д. «Модели и алгоритмы мультимодальной биометрической аутентификации на основе сверточной нейронной сети», представленной на соискание ученой степени кандидата технических наук, прошли апробацию, а также был использован в процессах Министерства науки и высшего образования Российской Федерации для разграничения доступа к автоматизированным рабочим местам и противодействия несанкционированному доступу:

- программный комплекс мультимодальной биометрической системы аутентификации пользователей информационной системы на основе сверточной нейронной сети.

Практическая ценность полученных результатов позволяет обеспечить бесперебойность функционирования информационных систем, минимизировать риски при воздействии угроз информационной безопасности на цифровую инфраструктуру.

Директор _____ /Айдарбеков А. М.

Зам. директора _____ /Великжанин В. В.

Вед. специалист _____ /Нурмухаметов Д. Р.

«УТВЕРЖДАЮ»
Генеральный директор
ООО «Информзащита» Уфа
_____/ Зубков С. В.
«12» 02 2025 г.

АКТ ВНЕДРЕНИЯ

о внедрении результатов диссертационной работы
Лушникова Никиты Дмитриевича на тему «Модели и алгоритмы мультимодальной биометрической аутентификации на основе сверточной нейронной сети»

Комиссия в составе: Генеральный директор ООО «Информзащита», Зубков Сергей Витальевич; Технический директор ООО «Информзащита», Новоселов Артем Владимирович; специалист ООО «Информзащита», Хакбердин Максим Азаматович, составила настоящий акт о том, что полученные в ходе исследования ключевые показатели диссертационной работы «Модели и алгоритмы мультимодальной биометрической аутентификации на основе сверточной нейронной сети» применяются на автоматизированных рабочих местах сотрудников компании и на контрольно-пропускных пунктах в системе контроля и управления доступом.

Полученные результаты диссертационной работы позволяют повысить уровень отказоустойчивости устройств сотрудников компании в целом, а также минимизировать риск потенциальных угроз в области информационной безопасности и разграничить доступ пользователей информационных систем.

Генеральный директор _____/Зубков С. В.

Технический директор _____/Новоселов А.В.

Специалист _____/ Хакбердин М.А.

«УТВЕРЖДАЮ»
Директор ООО «ИТ Энигма Уфа»
_____/ Оводов А. М.
«26» 02 2025 г.

АКТ ВНЕДРЕНИЯ

о внедрении результатов диссертационной работы
Лушникова Никиты Дмитриевича на тему «Модели и алгоритмы мультимодальной биометрической аутентификации на основе сверточной нейронной сети»

Комиссия в составе: директор ООО «ИТ Энигма Уфа», Оводов Александр Михайлович; заместитель директора ООО «ИТ Энигма Уфа», Кочетков Александр Александрович; специалист ООО «ИТ Энигма Уфа», Абдурахимов Исмоил Исроилович, составила настоящий акт о том, что полученные Лушниковым Н.Д. основные результаты кандидатской диссертации «Модели и алгоритмы мультимодальной биометрической аутентификации на основе сверточной нейронной сети» интегрированы и внедрены на автоматизированных рабочих местах организации для применения групповых политик на основе биометрической аутентификации пользователей информационных систем.

Результаты диссертационного исследования позволяют повысить уровень защиты информации и информационной безопасности компании в целом, а также минимизировать риск компрометации данных, содержащихся в информационных системах.

Директор _____/Оводов А. М.

Зам. директора _____/Кочетков А.А.

Специалист _____/Абдурахимов И.И.

## Приложение В: Листинг кода программного обеспечения.

```
from PyQt5.QtCore import QPropertyAnimation

from PyQt5.QtWidgets import QWidget, QVBoxLayout, QPushButton,
QApplication

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.Qt import *

import os

import subprocess



class Ui_MainWindow10(object):

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(900, 700)

        self.centralwidget = QtWidgets.QWidget(MainWindow)


        self.centralwidget.setStyleSheet("#centralwidget{\n"

"background-color: rgb(83, 0, 214);\n"

"}")

        self.centralwidget.setObjectName("centralwidget")

        self.verticalLayout =
QtWidgets.QVBoxLayout(self.centralwidget)

        self.verticalLayout.setObjectName("verticalLayout")


        self.frame = QtWidgets.QFrame(self.centralwidget)

        self.frame.setStyleSheet("QFrame {\n"

"background-color: rgb(83, 0, 214);\n"

"color: rgb(220, 220, 220);\n"

"border-radius: 10px\n"
```

```
"\n"
"}")
        self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
        self.frame.setObjectName("frame")

        self.progressBar = QtWidgets.QProgressBar(self.frame)
        self.progressBar.setGeometry(QtCore.QRect(0, 700, 181, 0))
        self.progressBar.setStyleSheet("""
        QProgressBar {
            background-color: rgb(83, 0, 214);
            color: rgb(83, 0, 214);
            border-style: none;
            border-radius: 10px;
            text-align: center;
            font-size: 30px;
        }
        """)
        self.progressBar.setObjectName("progressBar")
        self.progressBar.resize(self.width() - 120, 60)
        self.progressBar.move(550, 850)
        self.progressBar.setFormat('%p%')
        self.progressBar.setTextVisible(True)
        self.progressBar.setRange(0, 150)
        self.progressBar.setValue(40)

        self.label_4 = QtWidgets.QLabel(self.frame)
        self.label_4.setGeometry(QtCore.QRect(1700, 1010, 181, 31))
        self.label_4.setFont(
            QtGui.QFont('Times New Roman', 15)
        )
        self.label_4.setStyleSheet("color: rgb(255, 255, 255);")
```

```
        self.label_4.setObjectName("label_4")
        self.label = QtWidgets.QLabel(self.frame)
        self.label.setGeometry(QtCore.QRect(760, 75, 391, 231))
        self.label.setText("")
        self.label.setPixmap(QtGui.QPixmap("bsu3.jpg"))
        self.label.setAlignment(QtCore.Qt.AlignCenter)
        self.label.setObjectName("label")
        self.label_5 = QtWidgets.QLabel(self.frame)
        self.label_5.setGeometry(QtCore.QRect(320, 290, 391, 231))
        self.label_5.setFont(
            QtGui.QFont('Times New Roman', 27)
        )
        self.label_5.setStyleSheet("color: rgb(255, 255, 255);")
        self.label_5.setText("<strong>Биометрическая многофакторная
аутентификация с применением нейронных сетей </strong>")
        self.label_5.setWordWrap(True)
        self.label_5.setFixedWidth(1600)
        self.label_5.setObjectName("label_5")

        self.label_6 = QtWidgets.QLabel(self.frame)
        self.label_6.setGeometry(QtCore.QRect(570, 460, 391, 231))
        self.label_6.setFont(
            QtGui.QFont('Times New Roman', 24)
        )
        self.label_6.setStyleSheet("color: rgb(255, 255, 255);")
        self.label_6.setWordWrap(True)
        self.label_6.setFixedWidth(1600)
        self.label_6.setObjectName("label_6")

        self.verticalLayout.addWidget(self.frame)
        MainWindow.setCentralWidget(self.centralwidget)
```

```
        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow",
"MainWindow"))
        self.label_4.setText(_translate("MainWindow", "<strong>By
</strong> CYBERUPGRADE"))

class MainWindow(QtWidgets.QMainWindow, Ui_MainWindow10):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        self.resize(1920, 1080)

        # Класс анимации прозрачности окна
        self.animation = QPropertyAnimation(self, b'windowOpacity')
        self.animation.setDuration(1000)        # Продолжительность: 1
секунда

        # Выполните постепенное увеличение
        self.doShow()

        self.setupUi(self)

        self.counter = 0
        self.timer = QTimer()
        self.timer.timeout.connect(self.loading)
        self.timer.start(30)

        self.start_animation()
```

```
        self.setMinimumSize(QSize(300, 200))
        self.setWindowTitle("CYBERUPGRADE")

        self.bt1 = QPushButton('Аудио', self)
        self.bt1.clicked.connect(self.clickMethod1)
        self.bt1.resize(100,32)
        self.bt1.move(820, 700)
        self.bt2 = QPushButton('Фото', self)
        self.bt2.clicked.connect(self.clickMethod2)
        self.bt2.resize(100,32)
        self.bt2.move(1000, 700)
        self.bt3 = QPushButton('Выйти', self)
        self.bt3.clicked.connect(self.clickMethod3)
        self.bt3.resize(100,32)
        self.bt3.move(910, 760)
    def clickMethod1(self):
        self.label_6.setText("Внимание! Аудиозапись владельца ОС в
течение 15 секунд!")
        subprocess.Popen(['start1.exe'], stdout=subprocess.DEVNULL)
    def clickMethod2(self):
        self.label_6.setText("Смотрите на световой индикатор веб-
камеры до его потухания!")
        subprocess.Popen(['start2.exe'], stdout=subprocess.DEVNULL)
    def clickMethod3(self):
        self.close()

    def loading(self):
        self.progressBar.setValue(self.counter)
        self.counter += 1
        if self.counter == 151: self.timer.stop()

    def start_animation(self):
        opacity_effect = QtWidgets.QGraphicsOpacityEffect(self.label)
```

```
        self.label.setGraphicsEffect(opacity_effect)
        '''
        geometry_animation = QtCore.QPropertyAnimation(
            self.label,
            b"geometry",
            duration=4700,
            startValue=QtCore.QRect(200, -210, 671, 261),
            endValue=QtCore.QRect(42, 274, 391, 231),
        )
        '''
        opacity_animation = QtCore.QPropertyAnimation(
            opacity_effect,
            b"opacity",
            duration=6000,
            startValue=0.0,
            endValue=1.0
        )

        group = QtCore.QParallelAnimationGroup(self.label)
#       group.addAnimation(geometry_animation)
        group.addAnimation(opacity_animation)
        group.start(QtCore.QAbstractAnimation.DeleteWhenStopped)

    def doShow(self):
        try:
            self.animation.finished.disconnect(self.close)
        except:
            pass
        self.animation.stop()
        # Диапазон прозрачности постепенно увеличивается от 0 до 1.
        self.animation.setStartValue(0)
        self.animation.setEndValue(1)
```

```
            self.animation.start()


        def doClose(self):
            self.animation.stop()
            self.animation.finished.connect(self.close)  # Закройте окно,
когда анимация будет завершена
            # Диапазон прозрачности постепенно уменьшается с 1 до 0.
            self.animation.setStartValue(1)
            self.animation.setEndValue(0)
            self.animation.start()


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    w = MainWindow()
    w.showFullScreen()
    app.exec_()
```

Запись аудиозаписей

```
import pyaudio
import wave
import threading
import time
import subprocess
import os

class AudioRecorder():


    def __init__(self):

        self.open = True
        self.rate = 28800
        self.frames_per_buffer = 1024
        self.channels = 1
        self.format = pyaudio.paInt16
        self.audio_filename = "temp_audio2.wav"
        self.audio = pyaudio.PyAudio()
        self.stream = self.audio.open(format=self.format,
                                      channels=self.channels,
                                      rate=self.rate,
                                      input=True,
                                      frames_per_buffer =
self.frames_per_buffer)
        self.audio_frames = []

    def record(self):

        self.stream.start_stream()
```

```
        while(self.open == True):
            data = self.stream.read(self.frames_per_buffer)
            self.audio_frames.append(data)
            if self.open==False:

                break

    def stop(self):

        if self.open==True:

            self.open = False
            self.stream.stop_stream()
            self.stream.close()
            self.audio.terminate()

            waveFile = wave.open(self.audio_filename, 'wb')
            waveFile.setnchannels(self.channels)
waveFile.setsampwidth(self.audio.get_sample_size(self.format))
            waveFile.setframerate(self.rate)
            waveFile.writeframes(b''.join(self.audio_frames))
            waveFile.close()

        pass

    def start(self):

        audio_thread = threading.Thread(target=self.record)
        audio_thread.start()
```

```
def start_Arecording(filename):

      global audio_thread

      audio_thread = AudioRecorder()

      audio_thread.start()

      return filename

def start_audio_recording(filename):

      global audio_thread

      audio_thread = AudioRecorder()
      audio_thread.start()

      return filename

def stop_Arecording(filename):

      audio_thread.stop()

def file_manager(filename):

      local_path = os.getcwd()

      if os.path.exists(str(local_path) + "/temp_audio2.wav"):
            os.remove(str(local_path) + "/temp_audio2.wav")

if __name__== "__main__":
```

```
filename = "Default_user"
file_manager(filename)

start_Arecording(filename)
print ("Начало записи")

time.sleep(13)

stop_Arecording(filename)

print("Запись окончена!")
subprocess.Popen(['tab.exe'], stdout=subprocess.DEVNULL)
```

```
'exec(%matplotlib inline)'
import os
import librosa
import librosa.display
import IPython
import numpy as np
import pandas as pd
import scipy
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.spatial import distance
import platform
import wx
import sys
import ctypes
import subprocess
from tkinter import *
from tkinter import messagebox as mb

y,sr=librosa.load(r'temp_audio.wav')

y_harmonic, y_percussive = librosa.effects.hpss(y)

chroma=librosa.feature.chroma_cens(y=y_harmonic, sr=sr)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chroma,y_axis='chroma', x_axis='time')
plt.colorbar()

chroma_mean=np.mean(chroma,axis=1)
```

```
chroma_std=np.std(chroma,axis=1)
octave=['C','C#','D','D#','E','F','F#','G','G#','A','A#','B']
plt.figure(figsize=(15,5))
plt.title('Mean CENS')
sns.barplot(x=octave,y=chroma_mean)

plt.figure(figsize=(15,5))
plt.title('SD CENS 1')
sns.barplot(x=octave,y=chroma_std)
#Generate the chroma Dataframe
chroma_df=pd.DataFrame()
for i in range(0,12):
    chroma_df['chroma_mean_'+str(i)]=chroma_mean[i]
for i in range(0,12):
    chroma_df['chroma_std_'+str(i)]=chroma_mean[i]
chroma_df.loc[0]=np.concatenate((chroma_mean,chroma_std),axis=0)
chroma_df
plt.savefig('SD CENS 1.png')
shape1=np.concatenate((chroma_mean,chroma_std),axis=0)
print(shape1)

y,sr=librosa.load(r'temp_audio2.wav')

y_harmonic, y_percussive = librosa.effects.hpss(y)

chroma=librosa.feature.chroma_cens(y=y_harmonic, sr=sr)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chroma,y_axis='chroma', x_axis='time')
plt.colorbar()

chroma_mean=np.mean(chroma,axis=1)
chroma_std=np.std(chroma,axis=1)
```

```
octave=['C','C#','D','D#','E','F','F#','G','G#','A','A#','B']
plt.figure(figsize=(15,5))
plt.title('Mean CENS')
sns.barplot(x=octave,y=chroma_mean)

plt.figure(figsize=(15,5))
plt.title('SD CENS 2')
sns.barplot(x=octave,y=chroma_std)
chroma_df=pd.DataFrame()
for i in range(0,12):
    chroma_df['chroma_mean_'+str(i)]=chroma_mean[i]
for i in range(0,12):
    chroma_df['chroma_std_'+str(i)]=chroma_mean[i]
chroma_df.loc[0]=np.concatenate((chroma_mean,chroma_std),axis=0)
chroma_df
plt.savefig('SD CENS 2.png')
shape2=np.concatenate((chroma_mean,chroma_std),axis=0)
print(shape2)

a = distance.euclidean(shape1, shape2)
print(a)
if a >= 0.4:
    class MyForm(wx.Frame):
        def __init__(self):
            no_caption = (wx.MAXIMIZE_BOX | wx.RESIZE_BORDER
                          | wx.SYSTEM_MENU | wx.CLOSE_BOX |
wx.CLIP_CHILDREN)
            wx.Frame.__init__(self, None, title='Безопасность входа',
style=no_caption)
            self.panel = wx.Panel(self, -1)
            self.Maximize(True)
            wx.StaticText(self.panel, -1, "ВЫ НЕ ПРОШЛИ
АУТЕНТИФИКАЦИЮ!",
```

```
                (850, 450))
        self.button2=wx.Button(self.panel, -1, "Выйти из системы",
                (810, 600))
        self.Bind(wx.EVT_BUTTON, self.OnClose, self.button2)
        self.button2.SetDefault()

        button=wx.Button(self.panel, -1, "Ввести пароль",
                (1000, 600))
        self.Bind(wx.EVT_BUTTON, self.newwindow, button)

        loc = wx.IconLocation(r'C:\Windows\System32\credwiz.exe',
0)
        self.SetIcon(wx.Icon(loc))

    def OnClose(self, event):
        sys.exit(0)

    def newwindow(self,event):
        secondWindow = window2()
        secondWindow.Show()

class window2(wx.Dialog):
    def __init__(self):

        wx.Dialog.__init__(self, None, title="Логин")
        self.logged_in = False

        user_sizer = wx.BoxSizer(wx.HORIZONTAL)

        user_lbl = wx.StaticText(self, label="Имя пользователя:")
        user_sizer.Add(user_lbl, 0, wx.ALL|wx.CENTER, 9)
        self.user = wx.TextCtrl(self, style=wx.TE_PROCESS_ENTER)
```

```
            self.user.Bind(wx.EVT_TEXT_ENTER, self.onLogin)
            user_sizer.Add(self.user, 0, wx.ALL, 9)

            p_sizer = wx.BoxSizer(wx.HORIZONTAL)

            p_lbl = wx.StaticText(self, label="Пароль:")
            p_sizer.Add(p_lbl, 0, wx.ALL|wx.CENTER, 9)
            self.password = wx.TextCtrl(self,
style=wx.TE_PASSWORD|wx.TE_PROCESS_ENTER)
            self.password.Bind(wx.EVT_TEXT_ENTER, self.onLogin)
            p_sizer.Add(self.password, 0, wx.ALL, 9)

            main_sizer = wx.BoxSizer(wx.VERTICAL)
            main_sizer.Add(user_sizer, 0, wx.ALL, 9)
            main_sizer.Add(p_sizer, 0, wx.ALL, 9)

            btn = wx.Button(self, label="Подтвердить")
            btn.Bind(wx.EVT_BUTTON, self.onLogin)
            main_sizer.Add(btn, 0, wx.ALL|wx.CENTER, 9)

            self.SetSizer(main_sizer)

        def onLogin(self, event):

            nikita_name = "NikDL"
            user_name = self.user.GetValue()
            nikita_password = "Nik00"
            user_password = self.password.GetValue()
            if (user_name == nikita_name and user_password ==
nikita_password):
                    subprocess.Popen(['sub2.exe'],
stdout=subprocess.DEVNULL)
            else:
```

```
                    ctypes.windll.user32.MessageBoxW(0, "Вы
неправильно ввели логин или пароль", "Неавторизованный пользователь",
0)

                    subprocess.Popen(['form2.exe'],
stdout=subprocess.DEVNULL)


    if __name__ == '__main__':

        app = wx.App(False)

        frame = MyForm().Show()

        app.MainLoop()


if a < 0.4:

    subprocess.Popen(['sub2.exe'], stdout=subprocess.DEVNULL)
```

```
import cv2
import face_recognition
import numpy as np
import dlib
import platform
import wx
import sys
import ctypes
import os
import subprocess
from skimage import io
from scipy.spatial import distance
from tkinter import *
from tkinter import messagebox as mb


sp = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
facerec =
dlib.face_recognition_model_v1('dlib_face_recognition_resnet_model_v1.
dat')
detector = dlib.get_frontal_face_detector()


img = io.imread('cam.jpg')


dets = detector(img, 1)


for k, d in enumerate(dets):
    print("Detection {}: Left: {} Top: {} Right: {} Bottom:
{}".format(
        k, d.left(), d.top(), d.right(), d.bottom()))
    shape = sp(img, d)
```

```
face_descriptor1 = facerec.compute_face_descriptor(img, shape)

print(face_descriptor1)

cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
cap.set(3, 800)
cap.set(4, 600)

for i in range(220):
    cap.read()
ret, frame = cap.read()
cv2.imwrite('cam2.jpg', frame)
cap.release()

img2 = io.imread('cam2.jpg')

dets = detector(img2, 1)

for k, d in enumerate(dets):
    print("Detection {}: Left: {} Top: {} Right: {} Bottom:
{}".format(
        k, d.left(), d.top(), d.right(), d.bottom()))
    shape = sp(img2, d)

face_descriptor2 = facerec.compute_face_descriptor(img2, shape)

print(face_descriptor2)

a = distance.euclidean(face_descriptor1, face_descriptor2)

print (a)
if a >= 0.56:
```

```
    class MyForm(wx.Frame):
        def __init__(self):
            no_caption = (wx.MAXIMIZE_BOX | wx.RESIZE_BORDER
                          | wx.SYSTEM_MENU | wx.CLOSE_BOX |
wx.CLIP_CHILDREN)
            wx.Frame.__init__(self, None, title='Безопасность входа',
style=no_caption)
            self.panel = wx.Panel(self, -1)
            self.Maximize(True)
            wx.StaticText(self.panel, -1, "ВЫ НЕ ПРОШЛИ
АУТЕНТИФИКАЦИЮ!",
                          (850, 450))
            self.button2=wx.Button(self.panel, -1, "Выйти из системы",
                          (810, 600))
            self.Bind(wx.EVT_BUTTON, self.OnClose, self.button2)
            self.button2.SetDefault()

            button=wx.Button(self.panel, -1, "Ввести пароль",
                          (1000, 600))
            self.Bind(wx.EVT_BUTTON, self.newwindow, button)

            loc = wx.IconLocation(r'C:\Windows\System32\credwiz.exe',
0)
            self.SetIcon(wx.Icon(loc))

        def OnClose(self, event):
            sys.exit(0)

        def newwindow(self,event):
            secondWindow = window2()
            secondWindow.Show()

    class window2(wx.Dialog):
        def __init__(self):
```

```
        wx.Dialog.__init__(self, None, title="Логин")
        self.logged_in = False

        user_sizer = wx.BoxSizer(wx.HORIZONTAL)

        user_lbl = wx.StaticText(self, label="Имя пользователя:")
        user_sizer.Add(user_lbl, 0, wx.ALL|wx.CENTER, 9)
        self.user = wx.TextCtrl(self, style=wx.TE_PROCESS_ENTER)
        self.user.Bind(wx.EVT_TEXT_ENTER, self.onLogin)
        user_sizer.Add(self.user, 0, wx.ALL, 9)

        p_sizer = wx.BoxSizer(wx.HORIZONTAL)

        p_lbl = wx.StaticText(self, label="Пароль:")
        p_sizer.Add(p_lbl, 0, wx.ALL|wx.CENTER, 9)
        self.password = wx.TextCtrl(self,
style=wx.TE_PASSWORD|wx.TE_PROCESS_ENTER)
        self.password.Bind(wx.EVT_TEXT_ENTER, self.onLogin)
        p_sizer.Add(self.password, 0, wx.ALL, 9)

        main_sizer = wx.BoxSizer(wx.VERTICAL)
        main_sizer.Add(user_sizer, 0, wx.ALL, 9)
        main_sizer.Add(p_sizer, 0, wx.ALL, 9)

        btn = wx.Button(self, label="Подтвердить")
        btn.Bind(wx.EVT_BUTTON, self.onLogin)
        main_sizer.Add(btn, 0, wx.ALL|wx.CENTER, 9)

        self.SetSizer(main_sizer)

    def onLogin(self, event):
```

```
            nikita_name = "NikDL"
            user_name = self.user.GetValue()
            nikita_password = "Nik00"
            user_password = self.password.GetValue()
            if (user_name == nikita_name and user_password ==
nikita_password):
                    subprocess.Popen(['foto4.exe'],
stdout=subprocess.DEVNULL)
            else:
                    ctypes.windll.user32.MessageBoxW(0, "Вы
неправильно ввели логин или пароль", "Неавторизованный пользователь",
0)
                    subprocess.Popen(['form.exe'],
stdout=subprocess.DEVNULL)


    if __name__ == '__main__':
        app = wx.App(False)
        frame = MyForm().Show()
        app.MainLoop()


if a < 0.56:
    subprocess.Popen(['foto4.exe'], stdout=subprocess.DEVNULL)
```

Распознавание искомого пользователя (противодействие дипфейку)

```
import os

import cv2

import face_recognition

import numpy as np

import tkinter as tk

from tqdm import tqdm

from collections import defaultdict

from tkinter import messagebox


recognizer = cv2.face.LBPHFaceRecognizer_create()

recognizer.read('trainer/trainer.yml')

faceCascade =
cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

eyeCascade = cv2.CascadeClassifier("haarcascade_eye.xml")

open_eyeCascade =
cv2.CascadeClassifier("haarcascade_eye_tree_eyeglasses.xml")

left_eyeCascade =
cv2.CascadeClassifier("haarcascade_lefteye_2splits.xml")

right_eyeCascade =
cv2.CascadeClassifier("haarcascade_righteye_2splits.xml")


font = cv2.FONT_HERSHEY_SIMPLEX


id = 1


names = ['Nikita']


cam = cv2.VideoCapture(0, cv2.CAP_DSHOW)

cam.set(3, 640)

cam.set(4, 480)
```

```
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

while True:
    ret, img =cam.read()
    img = cv2.flip(img, 1)
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (30, 30),
       )

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
        id, eye_detector = recognizer.predict(gray[y:y+h,x:x+w])
        face = gray[y:y+h,x:x+w]
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        gray_face = gray[y:y+h,x:x+w]
        left_face_gray = gray[y:y+h, x+int(w/2):x+w]
        right_face_gray = gray[y:y+h, x:x+int(w/2)]

        open_eyes_glasses = open_eyeCascade.detectMultiScale(
            gray_face,
            scaleFactor=1.1,
            minNeighbors=5,
            minSize=(30, 30),
            flags = cv2.CASCADE_SCALE_IMAGE
            )
```

```
        eyes = eyeCascade.detectMultiScale(
            roi_gray,
            scaleFactor= 1.5,
            minNeighbors=10,
            minSize=(5, 5),
            )

        left_eye = left_eyeCascade.detectMultiScale(
            left_face_gray,
            scaleFactor=1.1,
            minNeighbors=5,
            minSize=(30, 30),
            flags = cv2.CASCADE_SCALE_IMAGE
            )

        right_eye = right_eyeCascade.detectMultiScale(
            right_face_gray,
            scaleFactor=1.1,
            minNeighbors=5,
            minSize=(30, 30),
            flags = cv2.CASCADE_SCALE_IMAGE
            )

        for (ex, ey, ew, eh) in eyes:
            cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0,
255, 0), 2)
        for (ex, ey, ew, eh) in right_eye:
            cv2.rectangle(right_face_gray, (ex, ey), (ex + ew, ey +
eh), (0, 255, 0), 2)
        for (ex, ey, ew, eh) in left_eye:
            cv2.rectangle(left_face_gray, (ex, ey), (ex + ew, ey +
eh), (0, 255, 0), 2)
```

```
        for (ex, ey, ew, eh) in open_eyes_glasses:
            cv2.rectangle(face, (ex, ey), (ex + ew, ey + eh), (0, 255,
0), 2)


        if (confidence < 60):
            id = 'Nikita Lushnikov'
        else:
            id = 'unknown'
        if (eye_detector > 99):
            id = 'Image'
        else:
            id = 'Nikita Lushnikov'
        cv2.putText(img, str(id), (x-5, y+h+15), font, 0.5,
(255,255,255), 2)


    cv2.imshow('Видео',img)


    k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
```

Форма аутентификации пользователя

```
import seaborn as sns
from scipy.spatial import distance
import platform
import wx
import os
import sys
import ctypes
import subprocess
from tkinter import *
from tkinter import messagebox as mb


class MyForm(wx.Frame):
    def __init__(self):
        no_caption = (wx.MAXIMIZE_BOX | wx.RESIZE_BORDER
                      | wx.SYSTEM_MENU | wx.CLOSE_BOX |
wx.CLIP_CHILDREN)
        wx.Frame.__init__(self, None, title='Безопасность входа',
style=no_caption)
        self.panel = wx.Panel(self, -1)
        self.Maximize(True)
        wx.StaticText(self.panel, -1, "ВЫ НЕ ПРОШЛИ АУТЕНТИФИКАЦИЮ!",
                      (850, 450))
        self.button2=wx.Button(self.panel, -1, "Выйти из системы",
                      (810, 600))
        self.Bind(wx.EVT_BUTTON, self.OnClose, self.button2)
        self.button2.SetDefault()

        button=wx.Button(self.panel, -1, "Ввести пароль",
                      (1000, 600))
        self.Bind(wx.EVT_BUTTON, self.newwindow, button)
```

```
        loc = wx.IconLocation(r'C:\Windows\System32\credwiz.exe', 0)
        self.SetIcon(wx.Icon(loc))

    def OnClose(self, event):
        subprocess.Popen(['off.exe'], stdout=subprocess.DEVNULL)

    def newwindow(self,event):
        secondWindow = window2()
        secondWindow.Show()

class window2(wx.Dialog):
    def __init__(self):

        wx.Dialog.__init__(self, None, title="Логин")
        self.logged_in = False

        user_sizer = wx.BoxSizer(wx.HORIZONTAL)

        user_lbl = wx.StaticText(self, label="Имя пользователя:")
        user_sizer.Add(user_lbl, 0, wx.ALL|wx.CENTER, 9)
        self.user = wx.TextCtrl(self, style=wx.TE_PROCESS_ENTER)
        self.user.Bind(wx.EVT_TEXT_ENTER, self.onLogin)
        user_sizer.Add(self.user, 0, wx.ALL, 9)

        p_sizer = wx.BoxSizer(wx.HORIZONTAL)

        p_lbl = wx.StaticText(self, label="Пароль:")
        p_sizer.Add(p_lbl, 0, wx.ALL|wx.CENTER, 9)
        self.password = wx.TextCtrl(self,
style=wx.TE_PASSWORD|wx.TE_PROCESS_ENTER)
        self.password.Bind(wx.EVT_TEXT_ENTER, self.onLogin)
```

```
        p_sizer.Add(self.password, 0, wx.ALL, 9)

        main_sizer = wx.BoxSizer(wx.VERTICAL)
        main_sizer.Add(user_sizer, 0, wx.ALL, 9)
        main_sizer.Add(p_sizer, 0, wx.ALL, 9)

        btn = wx.Button(self, label="Подтвердить")
        btn.Bind(wx.EVT_BUTTON, self.onLogin)
        main_sizer.Add(btn, 0, wx.ALL|wx.CENTER, 9)

        self.SetSizer(main_sizer)

    def onLogin(self, event):

        nikita_name = "NikDL"
        user_name = self.user.GetValue()
        nikita_password = "Nik00"
        user_password = self.password.GetValue()
        while True:
            p = user_name
            t = user_password
            if (p == nikita_name and t == nikita_password):
                        ctypes.windll.user32.MessageBoxW(0, "Вы вошли
в систему", "Система", 0)
                        subprocess.Popen(['foto4.exe'],
stdout=subprocess.DEVNULL)
            else:
                        ctypes.windll.user32.MessageBoxW(0, "Вы
неправильно ввели логин или пароль", "Неавторизованный пользователь",
0)
                        subprocess.Popen(['off.exe'],
stdout=subprocess.DEVNULL)

if __name__ == '__main__':
    app = wx.App(False)
    frame = MyForm().Show()
    app.MainLoop()
```

```
from PyQt5.QtCore import QPropertyAnimation

from PyQt5.QtWidgets import QWidget, QVBoxLayout, QPushButton,
QApplication

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.Qt import *

import os

import subprocess



class Ui_MainWindow10(object):

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(900, 700)

        self.centralwidget = QtWidgets.QWidget(MainWindow)


        self.centralwidget.setStyleSheet("#centralwidget{\n"

"background-color: rgb(83, 0, 214);\n"

"}")

        self.centralwidget.setObjectName("centralwidget")

        self.verticalLayout =
QtWidgets.QVBoxLayout(self.centralwidget)

        self.verticalLayout.setObjectName("verticalLayout")


        self.frame = QtWidgets.QFrame(self.centralwidget)

        self.frame.setStyleSheet("QFrame {\n"

"background-color: rgb(83, 0, 214);\n"

"color: rgb(220, 220, 220);\n"

"border-radius: 10px\n"
```

```
"\n"
"}")
        self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
        self.frame.setObjectName("frame")

        self.progressBar = QtWidgets.QProgressBar(self.frame)
        self.progressBar.setGeometry(QtCore.QRect(0, 700, 181, 0))
        self.progressBar.setStyleSheet("""
        QProgressBar {
            background-color: rgb(83, 0, 214);
            color: rgb(83, 0, 214);
            border-style: none;
            border-radius: 10px;
            text-align: center;
            font-size: 30px;
        }
        """)
        self.progressBar.setObjectName("progressBar")
        self.progressBar.resize(self.width() - 120, 60)
        self.progressBar.move(550, 850)
        self.progressBar.setFormat('%p%')
        self.progressBar.setTextVisible(True)
        self.progressBar.setRange(0, 150)
        self.progressBar.setValue(40)

        self.label_4 = QtWidgets.QLabel(self.frame)
        self.label_4.setGeometry(QtCore.QRect(1700, 1010, 181, 31))
        self.label_4.setFont(
            QtGui.QFont('Times New Roman', 15)
        )
        self.label_4.setStyleSheet("color: rgb(255, 255, 255);")
```

```
        self.label_4.setObjectName("label_4")
        self.label = QtWidgets.QLabel(self.frame)
        self.label.setGeometry(QtCore.QRect(760, 75, 391, 231))
        self.label.setText("")
        self.label.setPixmap(QtGui.QPixmap("bsu3.jpg"))
        self.label.setAlignment(QtCore.Qt.AlignCenter)
        self.label.setObjectName("label")
        self.label_5 = QtWidgets.QLabel(self.frame)
        self.label_5.setGeometry(QtCore.QRect(320, 290, 391, 231))
        self.label_5.setFont(
            QtGui.QFont('Times New Roman', 27)
        )
        self.label_5.setStyleSheet("color: rgb(255, 255, 255);")
        self.label_5.setText("<strong>Биометрическая многофакторная
аутентификация с применением нейронных сетей </strong>")
        self.label_5.setWordWrap(True)
        self.label_5.setFixedWidth(1600)
        self.label_5.setObjectName("label_5")

        self.label_6 = QtWidgets.QLabel(self.frame)
        self.label_6.setGeometry(QtCore.QRect(570, 460, 391, 231))
        self.label_6.setFont(
            QtGui.QFont('Times New Roman', 24)
        )
        self.label_6.setStyleSheet("color: rgb(255, 255, 255);")
        self.label_6.setWordWrap(True)
        self.label_6.setFixedWidth(1600)
        self.label_6.setObjectName("label_6")

        self.verticalLayout.addWidget(self.frame)
        MainWindow.setCentralWidget(self.centralwidget)
```

```
        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow",
"MainWindow"))
        self.label_4.setText(_translate("MainWindow", "<strong>By
</strong> CYBERUPGRADE"))

class MainWindow(QtWidgets.QMainWindow, Ui_MainWindow10):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        self.resize(1920, 1080)

        # Класс анимации прозрачности окна
        self.animation = QPropertyAnimation(self, b'windowOpacity')
        self.animation.setDuration(1000)        # Продолжительность: 1
секунда

        # Выполните постепенное увеличение
        self.doShow()

        self.setupUi(self)

        self.counter = 0
        self.timer = QTimer()
        self.timer.timeout.connect(self.loading)
        self.timer.start(30)

        self.start_animation()
```

```
        self.setMinimumSize(QSize(300, 200))
        self.setWindowTitle("CYBERUPGRADE")

        self.bt1 = QPushButton('Аудио', self)
        self.bt1.clicked.connect(self.clickMethod1)
        self.bt1.resize(100,32)
        self.bt1.move(820, 700)
        self.bt2 = QPushButton('Фото', self)
        self.bt2.clicked.connect(self.clickMethod2)
        self.bt2.resize(100,32)
        self.bt2.move(1000, 700)
        self.bt3 = QPushButton('Выйти', self)
        self.bt3.clicked.connect(self.clickMethod3)
        self.bt3.resize(100,32)
        self.bt3.move(910, 760)
    def clickMethod1(self):
        self.label_6.setText("Внимание! Аудиозапись владельца ОС в
течение 15 секунд!")
        subprocess.Popen(['start1.exe'], stdout=subprocess.DEVNULL)
    def clickMethod2(self):
        self.label_6.setText("Смотрите на световой индикатор веб-
камеры до его потухания!")
        subprocess.Popen(['start2.exe'], stdout=subprocess.DEVNULL)
    def clickMethod3(self):
        self.close()

    def loading(self):
        self.progressBar.setValue(self.counter)
        self.counter += 1
        if self.counter == 151: self.timer.stop()

    def start_animation(self):
        opacity_effect = QtWidgets.QGraphicsOpacityEffect(self.label)
```

```
        self.label.setGraphicsEffect(opacity_effect)
        '''
        geometry_animation = QtCore.QPropertyAnimation(
            self.label,
            b"geometry",
            duration=4700,
            startValue=QtCore.QRect(200, -210, 671, 261),
            endValue=QtCore.QRect(42, 274, 391, 231),
        )
        '''
        opacity_animation = QtCore.QPropertyAnimation(
            opacity_effect,
            b"opacity",
            duration=6000,
            startValue=0.0,
            endValue=1.0
        )

        group = QtCore.QParallelAnimationGroup(self.label)
#       group.addAnimation(geometry_animation)
        group.addAnimation(opacity_animation)
        group.start(QtCore.QAbstractAnimation.DeleteWhenStopped)

    def doShow(self):
        try:
            self.animation.finished.disconnect(self.close)
        except:
            pass
        self.animation.stop()
        # Диапазон прозрачности постепенно увеличивается от 0 до 1.
        self.animation.setStartValue(0)
        self.animation.setEndValue(1)
```

```
            self.animation.start()


        def doClose(self):
            self.animation.stop()
            self.animation.finished.connect(self.close)  # Закройте окно,
когда анимация будет завершена
            # Диапазон прозрачности постепенно уменьшается с 1 до 0.
            self.animation.setStartValue(1)
            self.animation.setEndValue(0)
            self.animation.start()


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    w = MainWindow()
    w.showFullScreen()
    app.exec_()
```

```
from PyQt5.QtCore import QPropertyAnimation

from PyQt5.QtWidgets import QWidget, QVBoxLayout, QPushButton,
QApplication

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.Qt import *

import os

import subprocess



class Ui_MainWindow10(object):

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(900, 700)

        self.centralwidget = QtWidgets.QWidget(MainWindow)


        self.centralwidget.setStyleSheet("#centralwidget{\n"

"background-color: rgb(83, 0, 214);\n"

"}")

        self.centralwidget.setObjectName("centralwidget")

        self.verticalLayout =
QtWidgets.QVBoxLayout(self.centralwidget)

        self.verticalLayout.setObjectName("verticalLayout")


        self.frame = QtWidgets.QFrame(self.centralwidget)

        self.frame.setStyleSheet("QFrame {\n"

"background-color: rgb(83, 0, 214);\n"

"color: rgb(220, 220, 220);\n"

"border-radius: 10px\n"
```

```
"\n"
"}")
        self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
        self.frame.setObjectName("frame")

        self.progressBar = QtWidgets.QProgressBar(self.frame)
        self.progressBar.setGeometry(QtCore.QRect(0, 700, 181, 0))
        self.progressBar.setStyleSheet("""
        QProgressBar {
            background-color: rgb(83, 0, 214);
            color: rgb(83, 0, 214);
            border-style: none;
            border-radius: 10px;
            text-align: center;
            font-size: 30px;
        }
        """)
        self.progressBar.setObjectName("progressBar")
        self.progressBar.resize(self.width() - 120, 60)
        self.progressBar.move(550, 850)
        self.progressBar.setFormat('%p%')
        self.progressBar.setTextVisible(True)
        self.progressBar.setRange(0, 150)
        self.progressBar.setValue(40)

        self.label_4 = QtWidgets.QLabel(self.frame)
        self.label_4.setGeometry(QtCore.QRect(1700, 1010, 181, 31))
        self.label_4.setFont(
            QtGui.QFont('Times New Roman', 15)
        )
        self.label_4.setStyleSheet("color: rgb(255, 255, 255);")
```

```
        self.label_4.setObjectName("label_4")
        self.label = QtWidgets.QLabel(self.frame)
        self.label.setGeometry(QtCore.QRect(760, 75, 391, 231))
        self.label.setText("")
        self.label.setPixmap(QtGui.QPixmap("bsu3.jpg"))
        self.label.setAlignment(QtCore.Qt.AlignCenter)
        self.label.setObjectName("label")
        self.label_5 = QtWidgets.QLabel(self.frame)
        self.label_5.setGeometry(QtCore.QRect(320, 290, 391, 231))
        self.label_5.setFont(
            QtGui.QFont('Times New Roman', 27)
        )
        self.label_5.setStyleSheet("color: rgb(255, 255, 255);")
        self.label_5.setText("<strong>Биометрическая многофакторная
аутентификация с применением нейронных сетей </strong>")
        self.label_5.setWordWrap(True)
        self.label_5.setFixedWidth(1600)
        self.label_5.setObjectName("label_5")

        self.label_6 = QtWidgets.QLabel(self.frame)
        self.label_6.setGeometry(QtCore.QRect(570, 460, 391, 231))
        self.label_6.setFont(
            QtGui.QFont('Times New Roman', 24)
        )
        self.label_6.setStyleSheet("color: rgb(255, 255, 255);")
        self.label_6.setWordWrap(True)
        self.label_6.setFixedWidth(1600)
        self.label_6.setObjectName("label_6")

        self.verticalLayout.addWidget(self.frame)
        MainWindow.setCentralWidget(self.centralwidget)
```

```
        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow",
"MainWindow"))
        self.label_4.setText(_translate("MainWindow", "<strong>By
</strong> CYBERUPGRADE"))

class MainWindow(QtWidgets.QMainWindow, Ui_MainWindow10):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        self.resize(1920, 1080)

        # Класс анимации прозрачности окна
        self.animation = QPropertyAnimation(self, b'windowOpacity')
        self.animation.setDuration(1000)        # Продолжительность: 1
секунда

        # Выполните постепенное увеличение
        self.doShow()

        self.setupUi(self)

        self.counter = 0
        self.timer = QTimer()
        self.timer.timeout.connect(self.loading)
        self.timer.start(30)

        self.start_animation()
```

```
        self.setMinimumSize(QSize(300, 200))
        self.setWindowTitle("CYBERUPGRADE")

        self.bt1 = QPushButton('Аудио', self)
        self.bt1.clicked.connect(self.clickMethod1)
        self.bt1.resize(100,32)
        self.bt1.move(820, 700)
        self.bt2 = QPushButton('Фото', self)
        self.bt2.clicked.connect(self.clickMethod2)
        self.bt2.resize(100,32)
        self.bt2.move(1000, 700)
        self.bt3 = QPushButton('Выйти', self)
        self.bt3.clicked.connect(self.clickMethod3)
        self.bt3.resize(100,32)
        self.bt3.move(910, 760)
    def clickMethod1(self):
        self.label_6.setText("Внимание! Аудиозапись владельца ОС в
течение 15 секунд!")
        subprocess.Popen(['start1.exe'], stdout=subprocess.DEVNULL)
    def clickMethod2(self):
        self.label_6.setText("Смотрите на световой индикатор веб-
камеры до его потухания!")
        subprocess.Popen(['start2.exe'], stdout=subprocess.DEVNULL)
    def clickMethod3(self):
        self.close()

    def loading(self):
        self.progressBar.setValue(self.counter)
        self.counter += 1
        if self.counter == 151: self.timer.stop()

    def start_animation(self):
        opacity_effect = QtWidgets.QGraphicsOpacityEffect(self.label)
```

```
        self.label.setGraphicsEffect(opacity_effect)
        '''
        geometry_animation = QtCore.QPropertyAnimation(
            self.label,
            b"geometry",
            duration=4700,
            startValue=QtCore.QRect(200, -210, 671, 261),
            endValue=QtCore.QRect(42, 274, 391, 231),
        )
        '''
        opacity_animation = QtCore.QPropertyAnimation(
            opacity_effect,
            b"opacity",
            duration=6000,
            startValue=0.0,
            endValue=1.0
        )

        group = QtCore.QParallelAnimationGroup(self.label)
#       group.addAnimation(geometry_animation)
        group.addAnimation(opacity_animation)
        group.start(QtCore.QAbstractAnimation.DeleteWhenStopped)

    def doShow(self):
        try:
            self.animation.finished.disconnect(self.close)
        except:
            pass
        self.animation.stop()
        # Диапазон прозрачности постепенно увеличивается от 0 до 1.
        self.animation.setStartValue(0)
        self.animation.setEndValue(1)
```

```
            self.animation.start()


        def doClose(self):
            self.animation.stop()
            self.animation.finished.connect(self.close)  # Закройте окно,
когда анимация будет завершена
            # Диапазон прозрачности постепенно уменьшается с 1 до 0.
            self.animation.setStartValue(1)
            self.animation.setEndValue(0)
            self.animation.start()


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    w = MainWindow()
    w.showFullScreen()
    app.exec_()
```

```
from PyQt5.QtCore import QPropertyAnimation

from PyQt5.QtWidgets import QWidget, QVBoxLayout, QPushButton,
QApplication

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.Qt import *

import os

import subprocess



class Ui_MainWindow10(object):

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(900, 700)

        self.centralwidget = QtWidgets.QWidget(MainWindow)


        self.centralwidget.setStyleSheet("#centralwidget{\n"

"background-color: rgb(83, 0, 214);\n"

"}")

        self.centralwidget.setObjectName("centralwidget")

        self.verticalLayout =
QtWidgets.QVBoxLayout(self.centralwidget)

        self.verticalLayout.setObjectName("verticalLayout")


        self.frame = QtWidgets.QFrame(self.centralwidget)

        self.frame.setStyleSheet("QFrame {\n"

"background-color: rgb(83, 0, 214);\n"

"color: rgb(220, 220, 220);\n"

"border-radius: 10px\n"
```

```
"\n"
"}")
        self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
        self.frame.setObjectName("frame")

        self.progressBar = QtWidgets.QProgressBar(self.frame)
        self.progressBar.setGeometry(QtCore.QRect(0, 700, 181, 0))
        self.progressBar.setStyleSheet("""
        QProgressBar {
            background-color: rgb(83, 0, 214);
            color: rgb(83, 0, 214);
            border-style: none;
            border-radius: 10px;
            text-align: center;
            font-size: 30px;
        }
        """)
        self.progressBar.setObjectName("progressBar")
        self.progressBar.resize(self.width() - 120, 60)
        self.progressBar.move(550, 850)
        self.progressBar.setFormat('%p%')
        self.progressBar.setTextVisible(True)
        self.progressBar.setRange(0, 150)
        self.progressBar.setValue(40)

        self.label_4 = QtWidgets.QLabel(self.frame)
        self.label_4.setGeometry(QtCore.QRect(1700, 1010, 181, 31))
        self.label_4.setFont(
            QtGui.QFont('Times New Roman', 15)
        )
        self.label_4.setStyleSheet("color: rgb(255, 255, 255);")
```

```
        self.label_4.setObjectName("label_4")
        self.label = QtWidgets.QLabel(self.frame)
        self.label.setGeometry(QtCore.QRect(760, 75, 391, 231))
        self.label.setText("")
        self.label.setPixmap(QtGui.QPixmap("bsu3.jpg"))
        self.label.setAlignment(QtCore.Qt.AlignCenter)
        self.label.setObjectName("label")
        self.label_5 = QtWidgets.QLabel(self.frame)
        self.label_5.setGeometry(QtCore.QRect(320, 290, 391, 231))
        self.label_5.setFont(
            QtGui.QFont('Times New Roman', 27)
        )
        self.label_5.setStyleSheet("color: rgb(255, 255, 255);")
        self.label_5.setText("<strong>Биометрическая многофакторная
аутентификация с применением нейронных сетей </strong>")
        self.label_5.setWordWrap(True)
        self.label_5.setFixedWidth(1600)
        self.label_5.setObjectName("label_5")

        self.label_6 = QtWidgets.QLabel(self.frame)
        self.label_6.setGeometry(QtCore.QRect(570, 460, 391, 231))
        self.label_6.setFont(
            QtGui.QFont('Times New Roman', 24)
        )
        self.label_6.setStyleSheet("color: rgb(255, 255, 255);")
        self.label_6.setWordWrap(True)
        self.label_6.setFixedWidth(1600)
        self.label_6.setObjectName("label_6")

        self.verticalLayout.addWidget(self.frame)
        MainWindow.setCentralWidget(self.centralwidget)
```

```
        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow",
"MainWindow"))
        self.label_4.setText(_translate("MainWindow", "<strong>By
</strong> CYBERUPGRADE"))

class MainWindow(QtWidgets.QMainWindow, Ui_MainWindow10):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
        self.resize(1920, 1080)

        # Класс анимации прозрачности окна
        self.animation = QPropertyAnimation(self, b'windowOpacity')
        self.animation.setDuration(1000)        # Продолжительность: 1
секунда

        # Выполните постепенное увеличение
        self.doShow()

        self.setupUi(self)

        self.counter = 0
        self.timer = QTimer()
        self.timer.timeout.connect(self.loading)
        self.timer.start(30)

        self.start_animation()
```

```
        self.setMinimumSize(QSize(300, 200))
        self.setWindowTitle("CYBERUPGRADE")

        self.bt1 = QPushButton('Аудио', self)
        self.bt1.clicked.connect(self.clickMethod1)
        self.bt1.resize(100,32)
        self.bt1.move(820, 700)
        self.bt2 = QPushButton('Фото', self)
        self.bt2.clicked.connect(self.clickMethod2)
        self.bt2.resize(100,32)
        self.bt2.move(1000, 700)
        self.bt3 = QPushButton('Выйти', self)
        self.bt3.clicked.connect(self.clickMethod3)
        self.bt3.resize(100,32)
        self.bt3.move(910, 760)
    def clickMethod1(self):
        self.label_6.setText("Внимание! Аудиозапись владельца ОС в
течение 15 секунд!")
        subprocess.Popen(['start1.exe'], stdout=subprocess.DEVNULL)
    def clickMethod2(self):
        self.label_6.setText("Смотрите на световой индикатор веб-
камеры до его потухания!")
        subprocess.Popen(['start2.exe'], stdout=subprocess.DEVNULL)
    def clickMethod3(self):
        self.close()

    def loading(self):
        self.progressBar.setValue(self.counter)
        self.counter += 1
        if self.counter == 151: self.timer.stop()

    def start_animation(self):
        opacity_effect = QtWidgets.QGraphicsOpacityEffect(self.label)
```

```
        self.label.setGraphicsEffect(opacity_effect)
        '''
        geometry_animation = QtCore.QPropertyAnimation(
            self.label,
            b"geometry",
            duration=4700,
            startValue=QtCore.QRect(200, -210, 671, 261),
            endValue=QtCore.QRect(42, 274, 391, 231),
        )
        '''
        opacity_animation = QtCore.QPropertyAnimation(
            opacity_effect,
            b"opacity",
            duration=6000,
            startValue=0.0,
            endValue=1.0
        )

        group = QtCore.QParallelAnimationGroup(self.label)
#       group.addAnimation(geometry_animation)
        group.addAnimation(opacity_animation)
        group.start(QtCore.QAbstractAnimation.DeleteWhenStopped)

    def doShow(self):
        try:
            self.animation.finished.disconnect(self.close)
        except:
            pass
        self.animation.stop()
        # Диапазон прозрачности постепенно увеличивается от 0 до 1.
        self.animation.setStartValue(0)
        self.animation.setEndValue(1)
```

```
        self.label.setGraphicsEffect(opacity_effect)
        '''
        geometry_animation = QtCore.QPropertyAnimation(
            self.label,
            b"geometry",
            duration=4700,
            startValue=QtCore.QRect(200, -210, 671, 261),
            endValue=QtCore.QRect(42, 274, 391, 231),
        )
        '''
        opacity_animation = QtCore.QPropertyAnimation(
            opacity_effect,
            b"opacity",
            duration=6000,
            startValue=0.0,
            endValue=1.0
        )

        group = QtCore.QParallelAnimationGroup(self.label)
#       group.addAnimation(geometry_animation)
        group.addAnimation(opacity_animation)
        group.start(QtCore.QAbstractAnimation.DeleteWhenStopped)

    def doShow(self):
        try:
            self.animation.finished.disconnect(self.close)
        except:
            pass
        self.animation.stop()
        # Диапазон прозрачности постепенно увеличивается от 0 до 1.
        self.animation.setStartValue(0)
        self.animation.setEndValue(1)
```

```
            self.animation.start()


        def doClose(self):
            self.animation.stop()
            self.animation.finished.connect(self.close)  # Закройте окно,
когда анимация будет завершена
            # Диапазон прозрачности постепенно уменьшается с 1 до 0.
            self.animation.setStartValue(1)
            self.animation.setEndValue(0)
            self.animation.start()


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    w = MainWindow()
    w.showFullScreen()
    app.exec_()
```