МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное

учреждение высшего образования

«Омский государственный университет им. Ф.М. Достоевского»

Кафедра информационной безопасности

УТВЕРЖДАЮ

И.о. заведующего кафедрой

\_\_\_\_\_\_\_\_\_\_\_ Богаченко Н. Ф.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025 г.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА РАСПОЗНАВАНИЯ ЛИЦ

Выпускная квалификационная работа

по направлению 10.03.01 – «Информационная безопасность»

Научный руководитель:

канд. технических наук, доцент

\_\_\_\_\_\_\_\_\_\_\_ Мироненко А. Н.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Выполнил:

студент группы МББ-101-О-01

\_\_\_\_\_\_\_\_\_\_\_ Волков В. В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Омск  
2025

**Содержание**

[Введение 3](#_Toc200760013)

[Глава 1. Анализ современных методов и технологий распознавания лиц 6](#_Toc200760014)

[1.1. Основы биометрической идентификации 6](#_Toc200760015)

[1.2. Алгоритмы и методы распознавания лиц 8](#_Toc200760016)

[1.3. Обзор программных решений и библиотек 10](#_Toc200760017)

[Глава 2. Проектирование и разработка системы распознавания лиц 15](#_Toc200760018)

[2.1. Анализ требований к системе 15](#_Toc200760019)

[2.2. Архитектура программной системы 17](#_Toc200760020)

[2.3. Проектирование базы данных 20](#_Toc200760021)

[2.4. Алгоритм работы системы распознавания 23](#_Toc200760022)

[2.5. Разработка системы аудита безопасности 26](#_Toc200760023)

[2.6. Реализация пользовательского интерфейса 29](#_Toc200760024)

[Заключение 34](#_Toc200760025)

[Список литературы 35](#_Toc200760026)

# Введение

В современном мире обеспечение информационной безопасности становится одной из приоритетных задач как для государственных учреждений, так и для коммерческих организаций. Традиционные методы идентификации пользователей, основанные на паролях, магнитных картах или специальных устройствах, не всегда обеспечивают необходимый уровень защиты и могут быть скомпрометированы. В связи с этим активно развиваются биометрические технологии, которые используют уникальные физиологические характеристики человека для его надежной идентификации.

Среди различных биометрических методов технология распознавания лиц занимает особое место благодаря отсутствию необходимости физического контакта с устройством, удобству использования и высокой точности идентификации. Современные алгоритмы машинного обучения и глубокие нейронные сети позволяют достигать точности распознавания свыше девяносто пяти процентов, что делает данную технологию применимой в критически важных системах безопасности.

Особую актуальность приобретают системы распознавания лиц, интегрированные с комплексными средствами мониторинга и аудита безопасности. Согласно требованиям федерального закона № 152-ФЗ “О персональных данных” и стандартов в области информационной безопасности, системы обработки биометрических данных должны обеспечивать полное протоколирование всех операций и предоставлять возможность анализа событий безопасности.

Анализ существующих коммерческих решений показывает, что большинство систем распознавания лиц либо имеют ограниченную функциональность в области аудита безопасности, либо требуют значительных финансовых затрат на внедрение и сопровождение. Это создает потребность в разработке доступных, но функционально полных решений, которые могут быть адаптированы под специфические требования различных организаций.

Цель работы состоит в разработке автоматизированной системы распознавания лиц с интегрированной подсистемой аудита безопасности, обеспечивающей эффективную биометрическую идентификацию пользователей и полное протоколирование событий безопасности.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ современных методов и технологий распознавания лиц, выбрать наиболее подходящие алгоритмы и программные библиотеки для реализации системы.
2. Спроектировать архитектуру программной системы, включающие модули распознавания лиц, управления пользователями и аудита безопасности.
3. Разработать алгоритмы обработки видеопотока в режиме реального времени с механизмами защиты от ложных срабатываний.
4. Создать подсистему управления пользователями с возможностями регистрации, редактирования и удаления учетных записей.
5. Реализовать систему аудита безопасности, обеспечивающую протоколирование всех событий и формирование отчетов для анализа.
6. Разработать графический интерфейс пользователя, обеспечивающий удобное взаимодействие со всеми функциями системы.
7. Провести тестирование разработанной системы и оценить эффективность её функционирования.

Объектом исследования являются системы биометрической идентификации на основе технологий распознавания лиц.

Предметом исследования выступают методы и алгоритмы автоматизированного распознавания лиц в системах контроля доступа с функциями аудита безопасности.

Разработанная система будет полезна для образовательных учреждений, малых и средних предприятий, государственных организаций, которым необходимо обеспечить контролируемый доступ к помещениям или информационным ресурсам с одновременным ведением детального аудита всех операций. Система может применяться в учебных целях для демонстрации принципов работы биометрических технологий и методов обеспечения информационной безопасности.

# Глава 1. Анализ современных методов и технологий распознавания лиц

## 1.1. Основы биометрической идентификации

Биометрическая идентификация представляет собой процесс автоматического распознавания личности человека на основе его уникальных физиологических и поведенческих характеристик. В отличие от традиционных методов аутентификации, которые основаны на том, что человек знает (пароль) или имеет (ключ, карта), биометрические системы используют то, чем человек является по своей природе.

Основные принципы биометрических систем базируются на том, что каждый человек обладает набором уникальных характеристик, которые остаются неизменными на протяжении длительного времени и могут быть измерены с помощью технических средств. Для использования в системах идентификации биометрические характеристики должны удовлетворять следующим требованиям: универсальность (присутствие у всех людей), уникальность (различие между людьми), постоянство (неизменность во времени), измеримость (возможность количественной оценки).

Процесс биометрической идентификации включает в себя несколько этапов. На этапе регистрации происходит сбор биометрических данных пользователя, их обработка и создание шаблона, который сохраняется в базе данных. Этап верификации предполагает повторный сбор биометрических данных и их сравнения с сохраненным шаблоном для подтверждения личностию. При идентификации данные сравниваются со всеми шаблонами в базе данных для определения личности пользователя.

Биометрические системы классифицируются по различным признакам. По типу используемых характеристик выделяют физиологические методы (отпечатки пальцев, геометрия лиц, радужная оболочка глаза, геометрия руки) и поведенческие методы (динамика подписи, походка, ритм работы на клавиатуре). По способу получения данных различают контактные системы, требующие физического взаимодействия с датчиком, и бесконтактные системы, работающие на расстоянии.

Важными характеристиками биометрических систем являются показатели точности распознавания. Коэффициент ложного принятия (False Acceptance Rate) показывает долю случаев, когда система ошибочно принимает неавторизованного пользователя. Коэффициент ложного отклонения (False Rejection Rate) характеризует долю случаев, когда система ошибочно отклоняет авторизованного пользователя. Равная вероятность ошибок (Equal Error Rate) представляет собой значение, при котором коэффициенты ложного принятия и ложного отклонения равны.

Среди различных биометрических технологий распознавание лиц занимает особое место благодаря своим преимуществам. Данный метод является бесконтактным, не требует специального оборудования помимо камеры, позволяет проводить идентификацию на расстоянии и может работать в режиме реального времени. Кроме того, технология распознавания лиц обладает высокой степенью социальной приемлемости, поскольку люди привыкли к визуальной идентификации в повседневной жизни.

Современные системы биометрической идентификации находят широкое применение в различных сферах. В области физической безопасности они используются для контроля доступа в здания и помещения, в банковской сфере - для защиты финансовых операций, в правоохранительной деятельности - для идентификации личности подозреваемых. Особое значение биометрические технологии приобретают в системах информационной безопасности, где они обеспечивают надежную защиту от несанкционированного доступа к конфиденциальной информации.

Внедрение биометрических систем в критически важных объектах регулируется нормативными документами в области информационной безопасности. В Российской Федерации основными документами являются федеральный закон "О персональных данных", стандарты серии ГОСТ Р по защите информации, а также ведомственные документы, устанавливающие требования к системам биометрической идентификации в конкретных сферах применения.

## 1.2. Алгоритмы и методы распознавания лиц

Технология распознавания лиц прошла длительный путь развития от простых геометрических методов до сложных алгоритмов машинного обучения. Современные подходы к решению задачи распознавания лиц можно разделить на несколько основных категорий, каждая из которых имеет свои преимущества и области применения.

Классические методы распознавания лиц основаны на статистических подходах и анализе геометрических характеристик. Метод главных компонент (Principal Component Analysis) является одним из наиболее известных подходов данной группы. Алгоритм создает собственные лица (eigenfaces) путем выделения главных компонент из множества обучающих изображений. Каждое лицо представляется как линейная комбинация собственных лиц, что позволяет значительно сократить размерность данных при сохранении наиболее важной информации.

Линейный дискриминантный анализ (Linear Discriminant Analysis) развивает идеи метода главных компонент, но фокусируется на поиске проекции, которая максимизирует различия между классами и минимизирует различия внутри классов. Данный подход показывает лучшие результаты классификации по сравнению с методом главных компонент, особенно при наличии достаточного количества обучающих данных для каждого класса.

Геометрические методы основаны на анализе расположения и соотношений ключевых точек лица. Алгоритмы данной группы измеряют расстояния между характерными элементами лица: глазами, носом, ртом, бровями. Создается вектор геометрических характеристик, который используется для сравнения и идентификации. Преимуществом геометрических методов является их устойчивость к изменениям освещения, однако они требуют точного определения ключевых точек и могут быть чувствительны к поворотам головы.

Локальные бинарные шаблоны (Local Binary Patterns) представляют собой метод описания текстуры изображения. Алгоритм анализирует локальные области изображения, создавая гистограммы распределения текстурных признаков. Данный подход эффективен для распознавания лиц при изменяющихся условиях освещения и обладает высокой вычислительной эффективностью.

Современные методы распознавания лиц основаны на технологиях машинного обучения и демонстрируют значительно более высокую точность по сравнению с классическими подходами. Метод опорных векторов (Support Vector Machines) создает гиперплоскость в пространстве признаков, которая оптимально разделяет различные классы. Алгоритм показывает хорошие результаты при работе с данными высокой размерности и устойчив к переобучению.

Искусственные нейронные сети революционизировали область распознавания лиц. Многослойные персептроны способны выявлять сложные нелинейные зависимости в данных, что позволяет достигать высокой точности классификации. Обучение нейронных сетей происходит путем настройки весовых коэффициентов на основе обучающей выборки с использованием алгоритма обратного распространения ошибки.

Глубокие нейронные сети представляют собой наиболее передовой подход в современном распознавании лиц. Сверточные нейронные сети (Convolutional Neural Networks) специально разработаны для обработки изображений и способны автоматически выявлять иерархические признаки. Нижние слои сети выделяют простые элементы (края, углы), средние слои комбинируют их в более сложные структуры (глаза, нос), а верхние слои формируют представление лица в целом.

Архитектуры глубоких сетей для распознавания лиц включают различные компоненты. Сверточные слои выполняют операцию свертки с использованием обучаемых фильтров, выявляя локальные признаки изображения. Слои объединения (pooling) снижают размерность данных, сохраняя наиболее важную информацию. Полносвязные слои на выходе сети выполняют окончательную классификацию на основе извлеченных признаков.

Процесс распознавания лиц с использованием глубоких нейронных сетей включает несколько этапов. На этапе обнаружения алгоритм локализует области изображения, содержащие лица. Этап выравнивания приводит найденные лица к стандартному положению и размеру. На этапе извлечения признаков нейронная сеть создает числовое представление лица в виде вектора высокой размерности. Заключительный этап сравнения вычисляет степень схожести между векторами признаков различных лиц.

Метрики для оценки качества распознавания лиц используют различные подходы к измерению расстояния между векторами признаков. Евклидово расстояние вычисляется как корень из суммы квадратов разностей соответствующих компонентов векторов. Косинусное расстояние основано на угле между векторами в многомерном пространстве. Расстояние Махаланобиса учитывает корреляции между компонентами и может быть более эффективным при наличии статистической информации о распределении данных.

Современные архитектуры нейронных сетей для распознавания лиц достигают точности свыше 99 процентов на стандартных тестовых наборах данных. Однако практическое применение показывает, что качество распознавания может значительно варьироваться в зависимости от условий съемки, качества изображения, освещения и других факторов. Это требует тщательного подбора алгоритмов и параметров для конкретных условий применения системы.

## 1.3. Обзор программных решений и библиотек

Рынок систем распознавания лиц представлен широким спектром коммерческих и открытых решений, каждое из которых имеет свои особенности, преимущества и ограничения. Анализ существующих решений необходим для обоснованного выбора технологической основы разрабатываемой системы.

Коммерческие системы распознавания лиц обычно представляют собой комплексные решения, включающие аппаратную и программную части. Компания NtechLab предлагает систему FindFace, которая демонстрирует высокие показатели точности и может работать с большими базами данных. Система поддерживает режим реального времени и интегрируется с различными типами камер. Однако стоимость лицензирования и технической поддержки делает данное решение доступным преимущественно для крупных организаций.

Российская компания VisionLabs разработала систему LUNA, которая сочетает высокую точность распознавания с возможностью работы в условиях ограниченных вычислительных ресурсов. Система поддерживает различные сценарии использования: от контроля доступа до анализа посещаемости. Преимуществом является соответствие требованиям российского законодательства, однако стоимость решения остается существенным фактором при выборе.

Система Tevian предоставляет облачные и локальные решения для распознавания лиц с возможностью интеграции через программные интерфейсы. Решение ориентировано на бизнес-применения и обеспечивает высокую точность распознавания. Основными ограничениями являются зависимость от интернет-соединения для облачной версии и высокие требования к аппаратному обеспечению для локального развертывания.

Международные платформы, такие как Amazon Rekognition и Microsoft Face API, предоставляют возможности распознавания лиц через облачные сервисы. Данные решения отличаются простотой интеграции и не требуют значительных вычислительных ресурсов на стороне клиента. Основными ограничениями являются необходимость постоянного подключения к интернету, вопросы конфиденциальности данных и ограничения на использование в критически важных системах.

Анализ коммерческих решений показывает, что они обладают высокой точностью и функциональностью, но имеют существенные недостатки для образовательных и исследовательских целей. Высокая стоимость лицензирования, ограничения на модификацию исходного кода и зависимость от внешних сервисов делают их неподходящими для разработки учебных и демонстрационных систем.

Открытые библиотеки для распознавания лиц предоставляют альтернативу коммерческим решениям, обеспечивая доступность исходного кода и возможность модификации под конкретные задачи. Библиотека OpenCV является одной из наиболее популярных и содержит реализации классических алгоритмов распознавания лиц. Преимуществами OpenCV являются широкая функциональность в области компьютерного зрения, активное сообщество разработчиков и обширная документация. Однако точность встроенных алгоритмов распознавания лиц уступает современным решениям на основе глубокого обучения.

Библиотека dlib предоставляет инструменты машинного обучения и компьютерного зрения, включая эффективные алгоритмы обнаружения и распознавания лиц. Преимуществом dlib является высокое качество реализации алгоритмов и оптимизация для производительности. Библиотека включает предварительно обученные модели для обнаружения ключевых точек лица и создания дескрипторов для распознавания.

Фреймворк TensorFlow от компании Google обеспечивает мощные возможности для разработки и обучения нейронных сетей. В экосистеме TensorFlow доступны предварительно обученные модели для распознавания лиц, такие как FaceNet. Преимуществами являются высокая точность современных архитектур нейронных сетей и возможность дообучения моделей на собственных данных. Недостатком может быть сложность освоения для разработчиков без опыта в области машинного обучения.

Библиотека face\_recognition, построенная на основе dlib, предоставляет простой интерфейс для решения задач распознавания лиц. Для обоснования выбора данной библиотеки необходимо рассмотреть альтернативные решения и провести их сравнительный анализ.

InsightFace представляет собой современную библиотеку для распознавания лиц на основе глубоких нейронных сетей. Библиотека обеспечивает высокую точность распознавания и поддерживает различные архитектуры моделей. Однако InsightFace требует значительных вычислительных ресурсов для обучения моделей и имеет более сложный интерфейс программирования, что усложняет интеграцию в образовательные проекты.

FaceNet от Google является одной из наиболее известных архитектур для распознавания лиц, основанной на создании векторных представлений лиц в высокомерном пространстве. Реализации FaceNet доступны в различных фреймворках машинного обучения, но требуют глубоких знаний в области нейронных сетей для корректной настройки и использования.

OpenFace представляет собой открытую реализацию алгоритмов распознавания лиц с использованием глубокого обучения. Библиотека предоставляет предварительно обученные модели и инструменты для их дообучения. Основным недостатком является сложность установки зависимостей и настройки рабочей среды, что может создать барьеры для начинающих разработчиков.

DeepFace от Facebook предлагает унифицированный интерфейс для работы с различными моделями распознавания лиц, включая VGG-Face, FaceNet и ArcFace. Библиотека упрощает сравнение различных подходов к распознаванию лиц, но требует установки множественных зависимостей и может быть избыточной для простых задач идентификации.

Сравнительный анализ библиотек показывает, что face\_recognition обладает оптимальным балансом между точностью распознавания и простотой использования. Основные преимущества данной библиотеки включают простоту интеграции, хорошую документацию и примеры кода, высокую точность распознавания благодаря использованию современных алгоритмов машинного обучения, и минимальные требования к настройке рабочей среды.

Процесс работы с библиотекой face\_recognition включает несколько простых шагов. Загрузка изображения выполняется стандартными средствами Python. Функция обнаружения лиц автоматически находит все лица на изображении и возвращает их координаты. Извлечение дескриптора создает числовое представление каждого лица в виде вектора из 128 элементов. Сравнение лиц выполняется путем вычисления расстояния между дескрипторами.

Выбор библиотеки face\_recognition для разработки системы обусловлен несколькими факторами. Простота интеграции позволяет сосредоточиться на разработке бизнес-логики системы, не углубляясь в детали реализации алгоритмов машинного обучения. Высокая точность распознавания обеспечивается использованием современных методов глубокого обучения. Активная поддержка и развитие библиотеки гарантируют исправление ошибок и добавление новых возможностей.

Дополнительным преимуществом выбранной библиотеки является её совместимость с другими популярными инструментами Python для разработки приложений. Интеграция с OpenCV обеспечивает возможности захвата и обработки видеопотока. Использование NumPy для работы с массивами данных гарантирует эффективность вычислений. Совместимость с различными графическими библиотеками позволяет создавать удобные пользовательские интерфейсы.

Таким образом, анализ существующих решений показал, что для разработки образовательной системы распознавания лиц оптимальным выбором является использование открытых библиотек, в частности face\_recognition, которая обеспечивает необходимую функциональность при сохранении простоты разработки и возможности модификации системы под конкретные требования.

# Глава 2. Проектирование и разработка системы распознавания лиц

## 2.1. Анализ требований к системе

Разработка автоматизированной системы распознавания лиц требует тщательного анализа функциональных и нефункциональных требований, которые определяют архитектуру будущего программного решения. Система должна обеспечивать надежную биометрическую идентификацию пользователей при соблюдении высоких стандартов информационной безопасности.

Функциональные требования определяют основные возможности системы. Первоочередной задачей является обеспечение распознавания лиц в режиме реального времени с использованием видеопотока от веб-камеры. Система должна автоматически обнаруживать лица на изображении, извлекать биометрические характеристики и сравнивать их с данными зарегистрированных пользователей. При успешном распознавании система обязана отображать информацию о пользователе и фиксировать событие в журнале безопасности.

Управление базой пользователей представляет собой критически важный функциональный блок. Система должна предоставлять возможности регистрации новых пользователей с загрузкой их фотографий, редактирования существующих записей, обновления биометрических данных и удаления учетных записей. Каждая операция управления пользователями должна сопровождаться соответствующими записями в журнале аудита для обеспечения полной прослеживаемости действий.

Подсистема аудита безопасности должна обеспечивать всестороннее протоколирование событий системы. Требуется фиксация всех попыток биометрической идентификации с указанием результата, времени выполнения и степени достоверности распознавания. Система должна регистрировать операции управления пользователями, системные события запуска и остановки компонентов, а также предоставлять возможности формирования отчетов и экспорта данных аудита.

Пользовательский интерфейс должен обеспечивать интуитивное взаимодействие с функциями системы через графическую оболочку. Интерфейс должен быть разделен на логические блоки, соответствующие основным функциональным областям: распознавание лиц, управление пользователями и мониторинг безопасности. Каждый блок должен предоставлять все необходимые элементы управления и обратную связь о состоянии выполняемых операций.

Нефункциональные требования определяют качественные характеристики системы. Производительность системы должна обеспечивать обработку видеопотока с частотой не менее 25 кадров в секунду при разрешении 640×480 пикселей. Время отклика на попытку идентификации не должно превышать 200 миллисекунд с момента обнаружения лица в кадре. Система должна поддерживать одновременную работу с базой данных, содержащей до 1000 зарегистрированных пользователей, без значительного снижения производительности.

Требования к надежности включают обеспечение стабильной работы системы в течение продолжительного времени без критических сбоев. Система должна корректно обрабатывать ошибочные ситуации, такие как отсутствие подключения к камере, повреждение файлов фотографий пользователей или недоступность базы данных. В случае возникновения ошибок система должна предоставлять информативные сообщения пользователю и предпринимать попытки автоматического восстановления работоспособности.

Требования к надежности включают обеспечение стабильной работы системы в течение продолжительного времени без критических сбоев. Система должна корректно обрабатывать ошибочные ситуации, такие как отсутствие подключения к камере, повреждение файлов фотографий пользователей или недоступность базы данных. В случае возникновения ошибок система должна предоставлять информативные сообщения пользователю и предпринимать попытки автоматического восстановления работоспособности.

Совместимость системы предполагает работу на различных версиях операционной системы Windows, начиная с версии 10, а также поддержку основных дистрибутивов Linux. Система должна корректно функционировать с различными моделями веб-камер, поддерживающих стандартные протоколы видеозахвата. Интерфейс должен адаптироваться к различным разрешениям экрана и масштабированию системы.

Требования к сопровождению включают обеспечение модульной архитектуры, позволяющей независимое обновление отдельных компонентов системы. Код должен содержать подробную документацию и комментарии, облегчающие понимание логики работы для последующего сопровождения. Система должна предоставлять диагностическую информацию для выявления и устранения возможных проблем в работе.

Ограничения и допущения определяют рамки функционирования системы. Предполагается использование системы в контролируемых условиях освещения с достаточной освещенностью для качественной работы камеры. Система ориентирована на идентификацию одного пользователя за раз и не предназначена для одновременного распознавания множественных лиц. Предполагается, что пользователи будут располагаться на расстоянии от 0,5 до 2 метров от камеры для обеспечения оптимального качества распознавания.

## 2.2. Архитектура программной системы

Архитектура разрабатываемой системы распознавания лиц основана на принципах модульного программирования и разделения ответственности между компонентами. Система построена по многоуровневой архитектуре, которая обеспечивает четкое разделение бизнес-логики, уровня данных и пользовательского интерфейса.

Общая архитектура системы включает четыре основных уровня. Уровень представления отвечает за взаимодействие с пользователем и реализован в виде графического интерфейса на основе библиотеки tkinter. Уровень бизнес-логики содержит основные алгоритмы распознавания лиц, управления камерой и обработки данных. Уровень данных обеспечивает хранение и извлечение информации о пользователях и событиях безопасности. Уровень аудита реализует функциональность протоколирования всех операций системы.

Модульная структура приложения организована в виде пакетов, каждый из которых отвечает за определенную область функциональности. Файловая организация проекта представлена на рисунке 1.

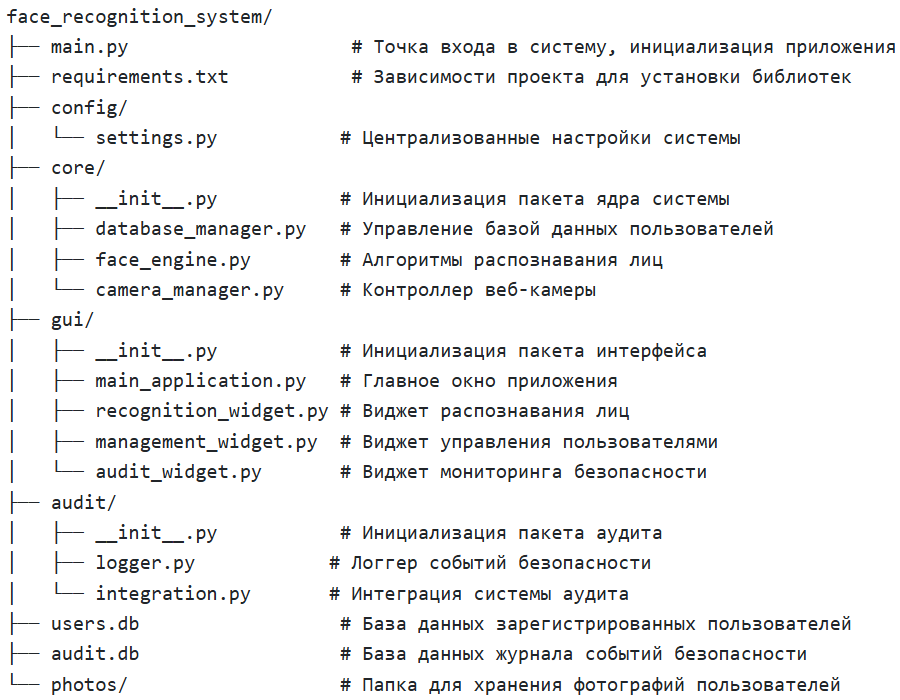


Рис. 1. Файловая организация проекта

Пакет core содержит ядро системы с основными компонентами для работы с данными, камерой и алгоритмами распознавания. Пакет gui включает все компоненты пользовательского интерфейса и логику взаимодействия с пользователем. Пакет audit реализует систему протоколирования событий безопасности. Пакет config содержит централизованные настройки системы.

Компонент DatabaseManager представляет собой основной интерфейс для работы с базой данных пользователей. Данный модуль инкапсулирует всю логику взаимодействия с SQLite, обеспечивая операции создания, чтения, обновления и удаления записей пользователей. Менеджер базы данных также отвечает за сериализацию и десериализацию биометрических отпечатков лиц, преобразуя многомерные массивы признаков в формат, пригодный для хранения в реляционной базе данных.

Листинг 1 – Фрагмент кода класса DatabaseManager и метода add\_user:

class DatabaseManager:

def \_\_init\_\_(self, db\_name=USERS\_DB):

self.db\_name = db\_name

self.initialize\_database\_structure()

def add\_user(self, user\_id, name, photo\_path, face\_encoding=None):

connection = sqlite3.connect(self.db\_name)

cursor = connection.cursor()

encoding\_blob = None

if face\_encoding is not None:

encoding\_blob = pickle.dumps(face\_encoding)

cursor.execute('''

INSERT INTO users (user\_id, name, photo\_path, face\_encoding)

VALUES (?, ?, ?, ?)

''', (user\_id, name, photo\_path, encoding\_blob))

Компонент FaceAnalysisEngine реализует центральную логику анализа и распознавания лиц. Движок анализа использует библиотеку face\_recognition для извлечения биометрических признаков и выполнения сравнений. Компонент поддерживает в памяти базу зарегистрированных пользователей для обеспечения высокой скорости распознавания в режиме реального времени. Алгоритм распознавания включает этапы обнаружения лиц на изображении, извлечения 128-мерных векторов признаков и сравнения с эталонными данными.

Компонент CameraController обеспечивает абстракцию над функциями видеозахвата операционной системы. Контроллер камеры управляет жизненным циклом подключения к устройству, настройкой параметров захвата и получением кадров для обработки. Реализация следует принципу получения ресурса при инициализации для обеспечения корректного освобождения системных ресурсов при завершении работы.

Система аудита безопасности построена как отдельный архитектурный слой, который интегрируется с основными компонентами через механизм внедрения зависимостей. SecurityAuditLogger обеспечивает централизованное протоколирование всех событий системы в отдельную базу данных. Компонент SecurityAuditIntegration реализует паттерн интегратора, обеспечивая бесшовное встраивание функций аудита в существующие модули без нарушения их основной логики.

Пользовательский интерфейс реализован по паттерну составного объекта (Composite), где главное приложение FaceRecognitionSystem выступает в роли контейнера для специализированных виджетов. Каждый виджет отвечает за определенную функциональную область и может быть разработан и тестирован независимо от других компонентов интерфейса.

FaceRecognitionWidget реализует интерфейс для работы с видеопотоком и отображения результатов распознавания. Виджет управляет циклом обработки кадров, визуализацией обнаруженных лиц и отображением информации о распознанных пользователях. Реализация использует неблокирующий подход к обработке видео через механизм планировщика событий tkinter.

UserManagementWidget предоставляет полнофункциональный интерфейс для управления базой пользователей. Виджет включает формы для ввода данных новых пользователей, предварительный просмотр загружаемых фотографий, таблицу существующих пользователей и элементы управления для редактирования и удаления записей. Все операции сопровождаются проверкой входных данных и обратной связью о результатах выполнения.

SecurityAuditWidget реализует интерфейс мониторинга безопасности системы. Виджет отображает статистическую информацию о работе системы в режиме реального времени, подробный журнал событий и предоставляет возможности экспорта данных аудита. Обновление информации происходит автоматически через заданные интервалы времени.

Взаимодействие между компонентами организовано через четко определенные интерфейсы и механизм обратных вызовов. Основные компоненты не имеют прямых зависимостей друг от друга, что обеспечивает слабую связанность и возможность независимого тестирования. Передача данных между уровнями осуществляется через параметры методов и возвращаемые значения, что исключает использование глобальных переменных и повышает предсказуемость поведения системы.

Система конфигурации реализована в виде централизованного модуля settings, который содержит все настраиваемые параметры системы. Такой подход позволяет легко адаптировать поведение системы под различные условия эксплуатации без модификации исходного кода основных компонентов. Настройки включают параметры алгоритмов распознавания, временные интервалы, размеры интерфейса и пути к файлам данных.

## 2.3. Проектирование базы данных

Проектирование структуры базы данных для системы распознавания лиц требует учета специфики хранения биометрических данных и обеспечения высокой производительности операций поиска и сравнения. Система использует реляционную модель данных, реализованную средствами SQLite, что обеспечивает локальное хранение без необходимости развертывания внешнего сервера базы данных.

Концептуальная модель данных основана на разделении информации на две основные предметные области: данные пользователей и события безопасности. Такое разделение позволяет оптимизировать структуру хранения под различные типы операций и обеспечивает возможность независимого масштабирования каждой части системы.

База данных пользователей содержит таблицу users, которая хранит всю информацию о зарегистрированных в системе пользователях. Структура таблицы включает поле id как автоинкрементный первичный ключ, обеспечивающий уникальную идентификацию каждой записи. Поле user\_id содержит пользовательский идентификатор, который должен быть уникальным в рамках системы и используется для связывания событий аудита с конкретными пользователями.

Листинг 2 – Фрагмент создания таблицы users:

CREATE TABLE users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id TEXT UNIQUE NOT NULL,

name TEXT NOT NULL,

photo\_path TEXT,

face\_encoding BLOB

);

Поле name хранит отображаемое имя пользователя, которое используется в интерфейсе системы для представления результатов распознавания. Поле photo\_path содержит путь к файлу фотографии пользователя в файловой системе, что позволяет отображать изображения в интерфейсе управления пользователями. Наиболее критичным является поле face\_encoding, которое содержит сериализованные биометрические характеристики лица пользователя.

Хранение биометрических отпечатков реализовано через сериализацию 128-мерных векторов признаков в двоичный формат с использованием библиотеки pickle. Такой подход обеспечивает точное сохранение числовых данных с плавающей точкой, необходимых для корректного функционирования алгоритмов сравнения. Двоичное представление также обеспечивает компактность хранения по сравнению с текстовыми форматами.

База данных аудита содержит таблицу security\_events, предназначенную для хранения всех событий безопасности системы. Структура таблицы разработана с учетом требований к полноте протоколирования и возможности последующего анализа данных аудита. Первичный ключ id обеспечивает уникальность и упорядоченность записей событий.

Листинг 3 – Фрагмент создания таблицы security\_events:

CREATE TABLE security\_events (

id INTEGER PRIMARY KEY AUTOINCREMENT,

timestamp TEXT NOT NULL,

event\_type TEXT NOT NULL,

user\_id TEXT,

result TEXT NOT NULL,

distance REAL

);

Поле timestamp содержит временную метку события, что обеспечивает стандартизированное представление времени и возможность точной хронологической сортировки событий. Использование текстового формата для хранения времени упрощает экспорт данных и обеспечивает читаемость журналов без дополнительного преобразования.

Поле event\_type определяет категорию события безопасности и принимает значения из предопределенного множества. Основными типами событий являются попытки распознавания лиц, операции управления пользователями и системные события. Стандартизация типов событий обеспечивает возможность автоматического анализа и построения отчетов по категориям активности.

Поле user\_id содержит ссылку на идентификатор пользователя, связанного с событием. Для системных событий, не связанных с конкретным пользователем, данное поле может содержать значение NULL. Такая структура обеспечивает возможность построения отчетов как по общей активности системы, так и по действиям конкретных пользователей.

Поле result фиксирует результат выполнения операции и принимает значения "success" или "failed". Это позволяет анализировать эффективность работы системы и выявлять потенциальные проблемы безопасности, такие как многократные неудачные попытки доступа.

Поле distance содержит числовое значение расстояния схожести для операций биометрического распознавания. Данный параметр позволяет анализировать качество распознавания и настраивать пороговые значения для оптимизации баланса между точностью и полнотой идентификации. Для событий, не связанных с распознаванием лиц, данное поле содержит значение NULL.

Стратегия резервного копирования предполагает создание копий файлов базы данных через регулярные интервалы времени. Поскольку SQLite хранит данные в виде файлов, резервное копирование может выполняться стандартными средствами файловой системы. Для обеспечения согласованности данных резервное копирование должно выполняться при отсутствии активных транзакций записи.

Безопасность данных может обеспечиваться средствами файловой системы через ограничение прав доступа к файлам базы данных. Дополнительная защита может быть реализована через шифрование файлов базы данных средствами операционной системы или специализированными версиями SQLite с поддержкой шифрования.

## 2.4. Алгоритм работы системы распознавания

Алгоритм работы системы распознавания лиц представляет собой сложный процесс обработки видеопотока в режиме реального времени с применением методов компьютерного зрения и машинного обучения. Процесс включает несколько последовательных этапов, каждый из которых выполняет специфические функции по преобразованию визуальной информации в результат идентификации.

Инициализация системы начинается с загрузки биометрических отпечатков всех зарегистрированных пользователей из базы данных в оперативную память. Данная операция выполняется однократно при запуске системы и обеспечивает высокую скорость последующих операций сравнения. Биометрические данные представляют собой 128-мерные векторы признаков, извлеченные из фотографий пользователей на этапе регистрации.

Листинг 4 – Метод загрузки биометрических данных в память:

def load\_facial\_encodings(self, encodings, user\_ids):

self.registered\_user\_encodings = encodings

self.registered\_user\_identifiers = user\_ids

print(f"Загружено биометрических отпечатков: {len(self.registered\_user\_encodings)}")

Процесс обработки видеопотока организован в виде циклического алгоритма, который выполняется с частотой около 30 итераций в секунду. На каждой итерации происходит захват текущего кадра с веб-камеры, его обработка алгоритмами компьютерного зрения и принятие решения о наличии и идентичности обнаруженных лиц.

Захват видеокадра осуществляется через интерфейс OpenCV, который обеспечивает стандартизированный доступ к устройствам видеозахвата операционной системы. Полученный кадр представляет собой трехмерный массив пикселей в цветовом пространстве BGR с разрешением, определяемым настройками камеры.

Предварительная обработка кадра включает масштабирование изображения для ускорения последующих вычислений. Коэффициент масштабирования составляет 0.25, что означает уменьшение размеров изображения в четыре раза по каждой оси. Такая оптимизация позволяет сократить время обработки в 16 раз при сохранении достаточной точности обнаружения лиц.

Листинг 5 – Метод обнаружения и распознавания лиц на видеокадре:

def detect\_and\_recognize\_faces(self, frame, scale=0.25):

small\_frame = cv2.resize(frame, (0, 0), fx=scale, fy=scale)

rgb\_small\_frame = cv2.cvtColor(small\_frame, cv2.COLOR\_BGR2RGB)

face\_locations = face\_recognition.face\_locations(rgb\_small\_frame)

face\_encodings = face\_recognition.face\_encodings(rgb\_small\_frame, face\_locations)

Обнаружение лиц выполняется алгоритмом на основе гистограммы направленных градиентов, который эффективно выявляет характерные паттерны лиц на изображении. Алгоритм возвращает координаты прямоугольных областей, содержащих обнаруженные лица, в формате верхняя граница, правая граница, нижняя граница, левая граница.

Извлечение биометрических признаков осуществляется сверточной нейронной сетью, обученной на большом наборе данных лиц. Нейронная сеть преобразует изображение лица в 128-мерный вектор признаков, который служит уникальным биометрическим отпечатком. Данный процесс является наиболее вычислительно затратным этапом алгоритма.

Сравнение биометрических отпечатков выполняется путем вычисления евклидова расстояния между вектором признаков обнаруженного лица и векторами всех зарегистрированных пользователей. Для каждого сравнения получается числовое значение расстояния, где меньшее значение указывает на большую степень схожести лиц.

Листинг 6 – Фрагмент кода, где происходит сравнение биометрических данных:

matches = face\_recognition.compare\_faces(self.registered\_user\_encodings, face\_encoding)

face\_distances = face\_recognition.face\_distance(self.registered\_user\_encodings, face\_encoding)

best\_match\_index = np.argmin(face\_distances)

distance = face\_distances[best\_match\_index]

Принятие решения о распознавании основано на двойном пороговом критерии. Первый критерий использует встроенный алгоритм библиотеки face\_recognition для определения наличия совпадения. Второй критерий сравнивает численное расстояние с настраиваемым порогом, который по умолчанию составляет 0.6. Лицо считается распознанным только при одновременном выполнении обоих условий.

Система защиты от избыточных срабатываний реализована через механизм временных ограничений. После успешного распознавания пользователя система игнорирует последующие распознавания того же пользователя в течение трех секунд. Аналогично, при обнаружении неизвестного лица система воздерживается от повторного протоколирования в течение пяти секунд.

Завершение работы алгоритма происходит при остановке камеры пользователем или закрытии приложения. Система корректно освобождает все задействованные ресурсы, включая подключение к камере и временные буферы обработки изображений. Все незавершенные операции протоколирования завершаются перед окончательным завершением работы системы.

## 2.5. Разработка системы аудита безопасности

Система аудита безопасности представляет собой критически важный компонент, обеспечивающий полное протоколирование всех операций системы распознавания. Разработка подсистемы аудита основана на принципах неизменяемости журнала событий, полноты протоколирования и обеспечения возможности последующего анализа.

Архитектура подсистемы аудита построена по модульному принципу с четким разделением ответственности между компонентами. Основой системы является модуль SecurityAuditLogger, который обеспечивает централизованное протоколирование всех типов событий. Модуль SecurityAuditIntegration реализует механизм внедрения функций аудита в существующие компоненты системы без нарушения их основной архитектуры.

Модуль SecurityAuditLogger инкапсулирует всю логику работы с базой данных аудита и предоставляет унифицированные методы для протоколирования различных типов событий. Инициализация модуля включает создание структуры базы данных аудита и настройку индексов для обеспечения высокой производительности операций поиска и анализа данных.

Листинг 6 – Фрагмент кода класса логгера событий и метода инициализации структуры базы данных аудита безопасности:

class SecurityAuditLogger:

def \_\_init\_\_(self, db\_name=AUDIT\_DB):

self.db\_name = db\_name

self.initialize\_audit\_database()

def initialize\_audit\_database(self):

connection = sqlite3.connect(self.db\_name)

cursor = connection.cursor()

cursor.execute('''

CREATE TABLE IF NOT EXISTS security\_events (

id INTEGER PRIMARY KEY AUTOINCREMENT,

timestamp TEXT NOT NULL,

event\_type TEXT NOT NULL,

user\_id TEXT,

result TEXT NOT NULL,

distance REAL

)

''')

Алгоритм логирования событий основан на принципе атомарности операций записи. Каждое событие фиксируется в отдельной транзакции, что обеспечивает целостность данных даже в случае неожиданного завершения работы системы. Временные метки событий генерируются в формате ISO 8601 для обеспечения стандартизированного представления времени и возможности точного хронологического анализа.

Категоризация событий безопасности включает несколько основных типов. События распознавания лиц фиксируют все попытки биометрической идентификации с указанием результата и степени достоверности. События управления пользователями протоколируют операции добавления, удаления и модификации учетных записей. Системные события отражают изменения состояния компонентов системы, такие как запуск и остановка камеры или загрузка биометрических данных.

Листинг 7 – Фрагмент кода метода логирования попытки биометрической идентификации и метода логирования операций управления пользователями:

def log\_face\_recognition\_attempt(self, user\_id=None, success=False, distance=1.0):

result = "success" if success else "failed"

self.\_write\_security\_event("recognition\_attempt", user\_id, result, distance)

def log\_user\_management\_action(self, action, user\_id, success=True):

result = "success" if success else "failed"

self.\_write\_security\_event(f"user\_{action}", user\_id, result, None)

Структура журнала безопасности спроектирована для обеспечения максимальной информативности записей при сохранении эффективности хранения. Каждая запись содержит уникальный идентификатор, временную метку, тип события, идентификатор связанного пользователя, результат операции и дополнительные параметры, специфичные для типа события.

Структура журнала безопасности спроектирована для обеспечения максимальной информативности записей при сохранении эффективности хранения. Каждая запись содержит уникальный идентификатор, временную метку, тип события, идентификатор связанного пользователя, результат операции и дополнительные параметры, специфичные для типа события.

Листинг 8 – Фрагмент кода метода генерации статистических данных безопасности:

def generate\_security\_statistics(self, days=7):

connection = sqlite3.connect(self.db\_name)

cursor = connection.cursor()

start\_date = datetime.datetime.now() - datetime.timedelta(days=days)

cursor.execute('''

SELECT event\_type, result, COUNT(\*), AVG(distance)

FROM security\_events

WHERE timestamp >= ?

GROUP BY event\_type, result

ORDER BY COUNT(\*) DESC

''', (start\_date.isoformat(),))

Функции экспорта данных обеспечивают возможность создания отчетов в формате CSV для последующего анализа внешними инструментами или представления контролирующим органам. Экспортируемые данные включают локализованные описания типов событий и результатов операций для повышения читаемости отчетов.

Интеграция системы аудита с основными компонентами реализована через паттерн внедрения зависимостей. Каждый компонент, требующий функций аудита, получает ссылку на экземпляр логгера через специальный метод установки зависимости. Такой подход обеспечивает слабую связанность компонентов и возможность независимого тестирования.

Листинг 8 – Фрагмент кода класса интеграции системы аудита безопасности с основным приложением:

class SecurityAuditIntegration:

@staticmethod

def integrate\_comprehensive\_audit\_system(app\_instance):

app\_instance.audit = SecurityAuditLogger()

app\_instance.audit\_widget = SecurityAuditWidget(

app\_instance.notebook,

app\_instance.audit

)

if hasattr(app\_instance, 'recognition\_widget'):

app\_instance.recognition\_widget.set\_audit\_logger(lambda: app\_instance.audit)

Обеспечение производительности системы аудита достигается через оптимизацию операций записи и использование индексов базы данных. Запись событий выполняется асинхронно относительно основных операций системы, что минимизирует влияние на производительность распознавания лиц. Индексы по временным меткам и типам событий ускоряют операции построения отчетов и фильтрации данных.

Защита целостности журнала аудита обеспечивается средствами файловой системы и организационными мерами. База данных аудита размещается в защищенной директории с ограниченными правами доступа. Рекомендуется регулярное создание резервных копий журнала аудита на отдельных носителях информации.

Автоматическое обновление данных аудита в пользовательском интерфейсе реализовано через механизм периодических запросов к базе данных. Интерфейс мониторинга обновляется каждые пять секунд, обеспечивая актуальность отображаемой информации без чрезмерной нагрузки на систему.

Система фиксирует все попытки доступа к данным пользователей, изменения в составе зарегистрированных лиц и системные события, влияющие на безопасность информации.

## 2.6. Реализация пользовательского интерфейса

Интерфейс построен на основе библиотеки tkinter, которая обеспечивает кроссплатформенную совместимость и не требует установки дополнительных зависимостей в большинстве дистрибутивов Python.

Проектирование графического интерфейса основано на принципе разделения функциональных областей через систему вкладок. Главное окно приложения содержит три основные вкладки, каждая из которых предоставляет доступ к определенной группе функций системы. Такая организация обеспечивает логическое разделение задач и упрощает навигацию для пользователей с различным уровнем технической подготовки.

Первая вкладка "Распознавание лиц" реализует основной функционал работы с видеопотоком. Интерфейс разделен на две области: левая панель содержит видеоизображение с камеры и элементы управления, правая панель отображает информацию о распознанном пользователе. Область видеоизображения имеет фиксированные размеры 480×360 пикселей, что обеспечивает стабильность макета интерфейса.

Элементы управления камерой включают кнопки запуска и остановки видеозахвата. Кнопки имеют взаимоисключающее состояние активности: при работающей камере активна только кнопка остановки, при выключенной камере доступна только кнопка запуска. Такое решение предотвращает некорректные операции и обеспечивает понятную обратную связь о состоянии системы.

Листинг 9 – Фрагмент кода метода запуска видеозахвата и начала обработки кадров:

def start\_camera(self):

if self.camera\_manager.start\_camera():

self.start\_button.config(state="disabled")

self.stop\_button.config(state="normal")

self.process\_video\_frame()

else:

messagebox.showerror("Ошибка", "Не удалось подключиться к камере!")

Информационная панель распознанного пользователя содержит статус распознавания, текстовые поля с идентификатором и именем пользователя, а также область отображения фотографии. Статус распознавания использует цветовое кодирование и символьные индикаторы для быстрого визуального восприятия результата. Фотография пользователя масштабируется до стандартного размера 170×170 пикселей с сохранением пропорций.

Вторая вкладка "Управление пользователями" предоставляет полнофункциональный интерфейс для администрирования базы пользователей. Интерфейс организован по принципу двухпанельного размещения: левая панель содержит форму добавления нового пользователя, правая панель отображает список существующих пользователей и элементы управления ими.

Форма добавления пользователя включает текстовые поля для ввода идентификатора и имени пользователя, секцию загрузки фотографии с предварительным просмотром и кнопку выполнения операции. Предварительный просмотр фотографии реализован в виде миниатюры размером 110×110 пикселей, что позволяет пользователю убедиться в корректности выбранного изображения перед добавлением в систему.

Листинг 10 – Фрагмент кода метода обработки выбора фотографии пользователя:

def handle\_photo\_selection(self):

file\_path = filedialog.askopenfilename(

title="Выберите фотографию пользователя",

filetypes=[("Изображения", "\*.jpg \*.jpeg \*.png \*.bmp")]

)

if file\_path:

self.selected\_photo\_file\_path = file\_path

pil\_image = Image.open(file\_path)

pil\_image = pil\_image.resize(PHOTO\_PREVIEW\_SIZE, Image.Resampling.LANCZOS)

photo\_preview = ImageTk.PhotoImage(pil\_image)

self.photo\_preview.config(image=photo\_preview, text="")

Список пользователей представлен в виде таблицы с колонками для идентификатора, имени и состояния фотографии. Таблица поддерживает прокрутку для работы с большим количеством записей и позволяет выбирать отдельные строки для выполнения операций редактирования или удаления. Элементы управления включают кнопки обновления фотографии, удаления пользователя и обновления списка.

Третья вкладка "Журнал безопасности" реализует интерфейс мониторинга системы аудита. Интерфейс разделен на верхнюю область со статистическими показателями и нижнюю область с детальным журналом событий. Статистическая панель отображает ключевые показатели эффективности системы в режиме реального времени.

Статистические карточки содержат общее количество попыток распознавания за текущий день, количество успешных операций, процент эффективности системы и время последней активности. Информация автоматически обновляется каждые пять секунд через механизм планировщика событий tkinter.

Листинг 11 – Фрагмент кода метода автоматического обновления данных аудита:

def schedule\_automatic\_refresh(self):

self.reload\_audit\_data()

self.frame.after(AUDIT\_DATA\_REFRESH\_INTERVAL, self.schedule\_automatic\_refresh)

Листинг 12 – Фрагмент кода метода обновления статистических показателей безопасности:

def refresh\_security\_metrics(self, stats):

total\_attempts = 0

successful = 0

for stat in stats['general\_stats']:

event\_type, result, count, avg\_distance = stat

if event\_type == 'recognition\_attempt':

total\_attempts += count

if result == 'success':

successful += count

success\_rate = (successful / total\_attempts \* 100) if total\_attempts > 0 else 0

self.success\_rate\_label.config(text=f"{success\_rate:.1f}%")

Журнал событий представлен в виде таблицы с колонками для времени, типа события, идентификатора пользователя, результата операции и степени схожести. Таблица поддерживает автоматическое обновление содержимого и цветовое кодирование строк в зависимости от типа и результата события. Успешные операции распознавания отображаются с зеленым фоном, неудачные попытки — с красным фоном, системные события — с нейтральным серым фоном.

Функция экспорта отчетов интегрирована в интерфейс журнала безопасности через специальную кнопку в заголовке таблицы. При активации функции пользователю предлагается выбрать местоположение для сохранения файла отчета в формате CSV. Экспортируемые данные включают локализованные описания событий и отформатированные временные метки для удобства анализа.

# Заключение

# Список литературы