

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МОиПЭВМ

Направление подготовки 09.03.02 «Информационные системы и технологии»

Направленность Информационные системы и технологии

БАКАЛАВРСКАЯ РАБОТА

на тему

**Информационная система идентификации
пользователей по биометрическим данным**

Студент

(подпись, дата)

Ионова Дарья Николаевна

(ФИО полностью)

Руководитель

(подпись, дата)

Козлов А.Ю.

(фамилия, инициалы)

Нормоконтролёр

(подпись, дата)

Такташкин Д.В.

(фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой

(подпись)

Козлов А.Ю.

(фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК

(подпись)

Попова Н.А.

(фамилия, инициалы)

Пенза, 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Кафедра

Вычислительной техники

МО и ПЭВМ

«Утверждаю»

Заведующий кафедрой

_____ А.Ю. Козлов

«25» декабря 2023 г

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА

1. Студент _____ Ионова Дарья Николаевна _____ гр. 20ВИ1 факультета ВТ

<Фамилия, имя, отчество полностью>

2. Тема работы _____ Информационная система идентификации пользователя по
_____ биометрическим данным _____

Тема утверждена приказом ПГУ № _____ от " _____ " _____ 20 _____

3. Руководитель работы _____ д.т.н., профессор кафедры «МО и ПЭВМ» Козлов А.Ю.

4. Задание на работу (назначение разработки, исходные данные и т. п.)

Назначение: информационная система предназначена для идентификации
пользователей с использованием биометрии.

Основные функции:

1) Загрузка и предварительная обработка изображения;

2) Сегментация изображения;

3) Выделение ключевых точек и представление их в виде матриц;

4) Сравнение полученной матрицы с уже имеющимися в базе данных;

5) Уведомление пользователя о результате идентификации.

Среда выполнения: операционная система Windows.

Средства разработки: язык Python, Spyder IDE, SQLite

5. Перечень подлежащих разработке вопросов

1) Анализ предметной области и требований к системе

2) Проектирование программных средств

3) Реализация программных средств

4) Тестирование программных средств

5) Оформление пояснительной записки

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и требований к системе	10%	08.04.2024 – 13.04.2024	
2	Проектирование программных средств	20%	15.04.2024 – 25.04.2024	
3	Реализация программных средств	30%	26.04.2024 – 12.05.2024	
4	Тестирование программных средств	20%	13.05.2024 – 22.05.2024	
5	Оформление пояснительной записки	20%	23.05.2024 – 03.06.2024	

Дата выдачи задания «21» декабря 2023г.

Руководитель бакалаврской работы _____

(подпись, дата)

Задание к исполнению принял студент _____

(подпись, дата)

Работу к защите допустить

Декан ФВТ _____ Л.Р. Фионова

(подпись, дата)

Реферат

Пояснительная записка содержит 65 листов, 17 рисунков, 19 использованных источников, 8 таблиц и 3 приложения.

БИОМЕТРИЧЕСКИЕ ДАННЫЕ, РАДУЖНАЯ ОБОЛОЧКА ГЛАЗА, ИДЕНТИФИКАЦИЯ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ.

Объект разработки: информационная система идентификации пользователей по биометрическим данным.

Цель работы: разработка информационной системы идентификации пользователей по радужной оболочке глаза.

Средства разработки: для написания информационной системы использовался язык Python и IDE Spyder. В качестве базы данных для хранения кодов радужных оболочек глаза использовался SQLite.

Результаты работы: В ходе выполнения дипломной работы была разработана информационная система идентификации пользователей по радужной оболочке глаза. Все требования к разработке соблюдены.

					ПГУ 09.03.02 – 08BP201.08 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	«Информационная система идентификации пользователей по биометрическим данным» Пояснительная записка.	Лит.	Лист	Листов
Разраб.		Ионова Д.Н.						
Провер.		Козлов А.Ю.					4	65
Реценз.						группа 20ВИ1		
Н. Контр.		Такташкин Д.В.						
Утверд.		Козлов А.Ю.						

Содержание

Введение.....	7
1 Анализ предметной области	9
1.1 Введение в предметную область.....	9
1.2 Характеристика информационной системы идентификации пользователей по биометрическим данным	10
1.3 Основные биометрические параметры.....	12
1.3.1 Распознавание отпечатков пальцев.....	13
1.3.2 Распознавание по радужной оболочке глаза.....	13
1.3.3 Распознавание по геометрии рук.....	14
1.3.4 Распознавание лица.....	14
1.3.5 Распознавание человека по голосу.....	15
1.3.6 Верификация подписи	16
1.3.7 Сравнительный анализ основных биометрических параметров....	17
1.4 Структура радужной оболочки глаза и ее свойства.....	18
2 Постановка задачи и анализ требований	20
2.1 Общая схема процесса распознавание по радужной оболочке глаза	20
2.2 Методы предобработки изображений	23
2.3 Сегментация	25
2.3.1 Метод на основе активных контуров.....	25
2.3.2 Метод, основанный на преобразовании Хафа	26
2.3.3 Метод, основанный на интегро-дифференциальном операторе Даугмана	27
2.4 Выделение ключевых точек и сравнение признаков	28
2.4.1 Дискретное вейвлет-преобразование с вейвлетом Хаара	28
2.4.2 Фильтр Габора	29
2.4.3 Расстояние Хэмминга	29
2.4.4 Коэффициент Жаккара	30
2.5 Обоснование выбора подхода к идентификации	31
3 Проектирование информационной системы	32
3.1 Анализ используемых технологий разработки системы.....	32

3.1.1	Обоснование выбора языка программирования	32
3.1.2	Обоснование выбора среды разработки	33
3.1.3	Конфигурации компьютера.....	33
3.1.4	Датасет.....	34
3.2	Описание проектируемой информационной системы.....	34
3.3	Проектирование системы объектным методом.....	38
3.3.1	Диаграмма классов.....	38
3.3.2	Диаграмма последовательности	39
3.3.3	Диаграмма деятельности.....	40
4	Реализация системы.....	41
4.1	Реализация процесса подготовки изображения	41
4.2	Реализация процесса выделения ключевых точек	41
4.3	Реализация процесса сравнения радужных оболочек глаза.....	42
5	Тестирование системы и анализ качества кода	43
5.1	Структурное тестирование	43
5.1.1	Метод покрытия операторов.....	45
5.1.2	Метод покрытия условий	46
5.1.3	Метод покрытия маршрутов	47
5.2	Метрики кода	47
5.2.1	SLOC-метрика	48
5.2.2	Метрика Холстеда.....	48
5.2.3	Цикломатическая метрика Мак-Кейба	50
	Заключение	52
	Список использованных источников	53
	Приложение А (справочное) Глоссарий	55
	Приложение Б (рекомендуемое) Листинг программы	57

Введение

Вопросы надежной и точной системы биометрической идентификации и аутентификации пользователей является актуальной темой на сегодняшний день. Традиционные методы, такие как пароли, все чаще оказываются уязвимыми к угрозам безопасности. В данных условиях все более перспективным решением проблемы является биометрическая идентификация, способная значительно увеличить уровень безопасности.

Биометрические данные представляют собой уникальные физические и поведенческие характеристики человека, такие как отпечатки пальцев, радужная оболочка глаза, лицо, геометрия рук, голос. Такие данные трудно подделать или украсть, что делает биометрическую идентификацию одной из самых надежных форм аутентификации.

В истории, первое упоминание биометрических данных можно найти в древних цивилизациях. Отпечатки пальцев, например, использовались Древнем Китае для подписи юридических документов. Однако настоящий технологический прорыв в сфере биометрии произошел в последние несколько десятилетий.

В 1892 году была опубликована книга Френсиса Гальтона «Fingerprint», в которой была доказана уникальность отпечатков пальцев и их неизменность в течении жизни. Именно эта работа стала основополагающей для научного подхода к идентификации по отпечаткам пальцев.

С начала двадцатого века биометрические методы стали активно использоваться в правоохранительных органах. Таким образом, в 1924 году службы безопасности США начали собирать и хранить общую базу данных отпечатков пальцев. А уже через полвека началось активное использование автоматизированных систем распознавания отпечатков. Параллельно с этим развивались и такие биометрические методы как распознавание лиц и голоса. Однако они оставались малоиспользуемыми из-за технических ограничений и недостаточной точности.

С развитием компьютерных технологий в конце двадцатого века, биометрические данные стали активно изучаться и использоваться. Были разработаны алгоритмы распознавания лиц, голоса и других биометрических характеристик. В 1985 году была запатентована система идентификации по радужной оболочке глаза. Исследования в этой статье показали, что радужная оболочка глаза, как биометрический параметр, является уникальной, надежной, а также устойчивой к внешним факторам.

Сегодня биометрические данные активно используются во многих областях, включая государственную безопасность, банки, медицину. Уникальность радужной оболочки глаза, ее стабильность и сложность структуры делают этот биометрический параметр одним из самых надежных и безопасных в мире.

Цель работы – разработка информационной системы для идентификации пользователей по радужной оболочке глаза.

Задачи:

- 1) Изучить предметную область, чтобы выбрать методы для конкретных этапов идентификации пользователей.
- 2) Разработать алгоритм, включающий этапы предварительной обработки, сегментации, выделения ключевых точек и сравнения образца и эталонной радужной оболочки глаза.
- 3) Программно реализовать разработанный алгоритм.
- 4) Протестировать работу информационной системы.

1 Анализ предметной области

1.1 Введение в предметную область

Аутентификация – это процедура, позволяющая одной сущности проверить объявленные свойства другой [1].

Выделяется три традиционных способа аутентификации:

- 1) по собственности – физическим предметам, например, ключи, паспорт;
- 2) по знаниям – данные, неизвестные никому, кроме владельца, например, пароли и пин-коды;
- 3) по биометрическим параметрам – динамическим или статистическим параметрам человека.

Наиболее надежным способом являются биометрические системы контроля, так как такие характеристики не могут быть утеряны или украдены, а также не могут быть переданы третьим лицам или скопированы.

Биометрические параметры исследуются в науке – биометрии. Данная область знаний изучает методы и средства измерения и формализации персональных характеристик человека. Целью биометрических систем является распознавание и аутентификация на основе некоторых физиологических или поведенческих характеристик, которые по своей сути уникальны для каждого человека [2].

Физиологические (статистические) биометрические параметры являются физическими характеристиками и измеряются в определенный момент времени. Такими параметрами являются отпечатки пальцев, радужная оболочка глаза глаза или геометрия руки.

Поведенческие (динамические) биометрические параметры являются последовательностью действий и для их измерения требуется определенный период времени. Например, голос или клавиатурный почерк.

Статистические характеристики достаточно разнообразны и не подвержены влиянию окружающей среды, поэтому зачастую их достаточно для

точной идентификации пользователей. Динамические характеристики, так как поддаются влиянию окружения, обычно не могут быть идентифицированы однозначно.

Биометрические параметры также обладают другими свойствами, которые позволяют применять их на практике. Роджером Кларком в статье «Human identification in information systems: Management challenges and public policy issues» [3] были описаны свойства биометрических параметров:

- 1) всеобщность: каждый человек имеет биометрические характеристики;
- 2) уникальность: для биометрии нет двух людей, обладающих одинаковыми биометрическими характеристиками;
- 3) постоянство: биометрические характеристики должны быть стабильны во времени;
- 4) измеряемость: биометрические характеристики должны быть измеряемы каким-либо физическим считывающим устройством;
- 5) приемлемость: совокупность пользователей и общество в целом не должны (сильно) возражать против измерения/сбора биометрических параметров.

1.2 Характеристика информационной системы идентификации пользователей по биометрическим данным

Информационная система идентификации пользователей по биометрическим данным представляет собой совокупность программных и аппаратных средств, осуществляющих получение, обработку, хранение, сравнение данных, а также, выдачу результатов идентификации. Структура такой системы представлена на рисунке 1.

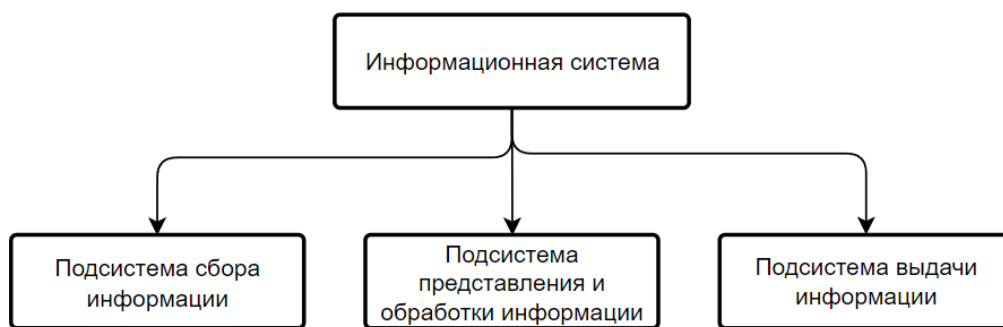


Рисунок 1 – Схема информационной системы

Подсистема сбора информации представляет собой получение и накопление данных для передачи в последующие подсистемы.

Подсистема обработки и представления является ключевым компонентом информационной системы и отвечает за отображение и обработку данных. Она показывает, как разработчики и пользователи системы представляют себе структуру и детали предметной области, для которой и была разработана информационная система.

Подсистема выдачи информации отвечает за предоставление доступа пользователям, которые могут пользоваться системой, а также обеспечивает выполнение задач, для которых информационная система и была создана.

Процессы, происходящие в информационной системе, представлены на рисунке 2.

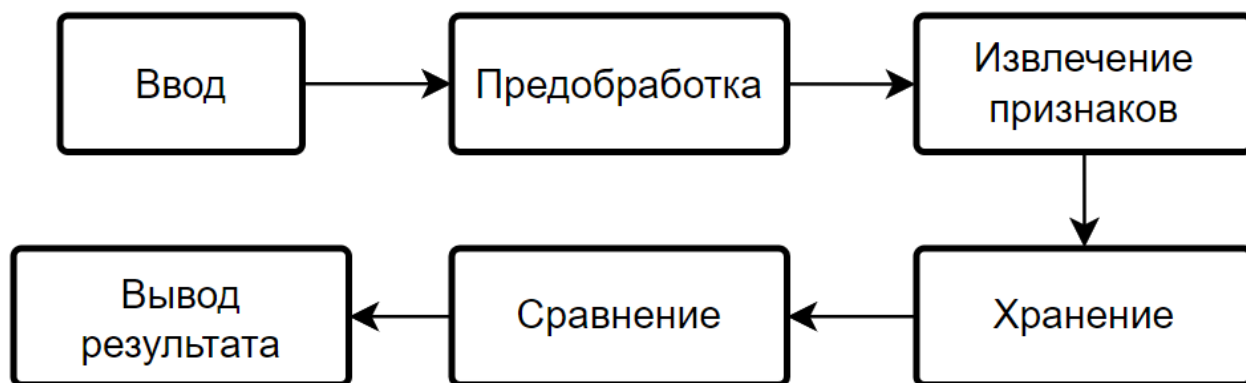


Рисунок 2 – Процессы в информационной системе

Ввод данных может представлять собой загрузку изображения биометрического параметра в систему пользователем или захват изображения

системой. Для данного этапа требуется датчик, сканер или камера высокого разрешения, в зависимости от используемых биометрических данных.

Предварительная обработка заключается в улучшении качества полученного изображения для более качественной дальнейшей обработки. На данном этапе могут использоваться различные методы для уменьшения шумов и искажений.

В качестве методов для извлечения признаков биометрического параметра могут также использоваться как статистические алгоритмы, так и методы машинного обучения или искусственного интеллекта.

Хранение информации может осуществляться любыми средствами с достаточной защищенностью. К таковым можно отнести базы данных, облачное хранилище, хранение на накопителях или устройствах пользователей. Каждый из этих методов имеет свои недостатки и преимущества, поэтому выбирать следует исходя из приоритетов и возможностей системы.

Сравнение также представляет собой некоторые алгоритмы, способные определить коэффициент схожести вводного изображения и хранящегося в базе данных.

Вывод результата может быть в виде графического интерфейса, голосового или даже тактильного. Любой из этих видов должен сообщать пользователю о результате работы предыдущих этапов.

В данной дипломной работе основное внимание было уделено процессу обработки, извлечения признаков, а также сравнения. Для извлечения признаков будут использоваться статистические алгоритмы. Для хранения информации будет выбрана база данных, т. к. является наиболее используемым средством хранения информации и отличается наибольшей безопасностью.

1.3 Основные биометрические параметры

Наиболее часто используемыми, в данный момент, являются шесть биометрических параметров: отпечаток пальца, радужная оболочка глаза, геометрия рук, лицо, голос, подпись.

1.3.1 Распознавание отпечатков пальцев

Дактилоскопия (распознавание отпечатков пальцев) – наука, изучающая строение папиллярных узоров пальцев рук, одной из основных задач которой, является идентификация личности [4].

Уникальность кожных узоров позволяет определить человека, что дало возможность применять данный метод биометрии в различных областях, таких как криминалистика, системы контроля и управления, банковская сфера.

Алгоритмы для распознавания отпечатков пальцев используют характерные точки: окончание линии узора, разветвление линий, одиночные точки. Кроме того, также может учитываться относительное положение замкнутых линий папиллярного узора, «арочных» и спиральных линий. В базу данных заносится закодированные ключевые точки для дальнейшего поиска и сравнения при идентификации человека.

Преимуществами данного метода является высокая достоверность – степень распознавания по отпечаткам пальцев для определения личности, по статистике, несколько лучше, чем степень распознавания по лицу, голосу или подписи. Это обусловлено наибольшей разработанностью биометрического метода в связи с его широким распространением в 20 веке.

Главным недостатком можно назвать чувствительность к мелким повреждениям кожных покровов пальцев. При минимальном физическом или химическом воздействии, сканеры отпечатков пальцев сообщают об отказе сканирования или объявляют пальцы подделкой.

1.3.2 Распознавание по радужной оболочке глаза

Распознавание радужной оболочки глаза представляет собой идентификацию личности на основе уникальных узоров кольцевой области глаза, располагающейся между зрачком (внутренняя часть) и склерой (белок глаза или внешняя часть). Радужная оболочка более чем в два раза точнее

отпечатка пальца благодаря строению и низкой подверженности механическому повреждению, а изменение размера зрачков делает копирование невозможным [5].

Процесс идентификации человека по радужной оболочке глаза можно условно разделить на три этапа: захват изображения и его первичная обработка, выделение ключевых точек и их кодирование, сравнение кодированных ключевых точек с имеющимися кодами в базе данных и команда о допуске, или отказе исполнительному устройству.

Преимуществами такого метода является статистическая надежность алгоритма в сравнении с другими методами биометрии. Получение изображения может происходить с расстояния в несколько сантиметров до нескольких метров, при этом физический контакт с устройством не требуется.

Недостатком можно назвать цену программного комплекса, используемого радужную оболочку глаза, как правило, система, основанная на распознавании отпечатков пальцев или лица, несколько ниже.

1.3.3 Распознавание по геометрии рук

Метод распознавания по геометрии рук представляет собой измерение пяти основных линий, существующих на любой ладони и семнадцать дополнительных геометрических параметров. Изначально определяются пять основных характеристик: ширина ладони, радиус окружности, вписанной в центр кисти, длина пальцев рук и высота кисти. Исходя из полученных параметров и происходит идентификация человека [6].

Однако из-за низкой достоверности и надежности, вследствие сильной подверженности изменениям во времени, данный метод считается устаревшим и не используется для идентификации личности.

1.3.4 Распознавание лица

Распознавание лица является одним из самых обоснованных биометрических параметров, так как человек без проблем может отличить людей

по их лицам. Такие характеристики как рот, нос, глаза, форма головы формируют уникальную характеристику, благодаря которой мы можем определить личность человека. Метод распознавания по лицу включает в себя определение формы лица, соотношения частей лица между собой, формы лба, скулы и подбородка и прочее [7].

Общий процесс распознавания лица выглядит следующим образом (рисунок 3):

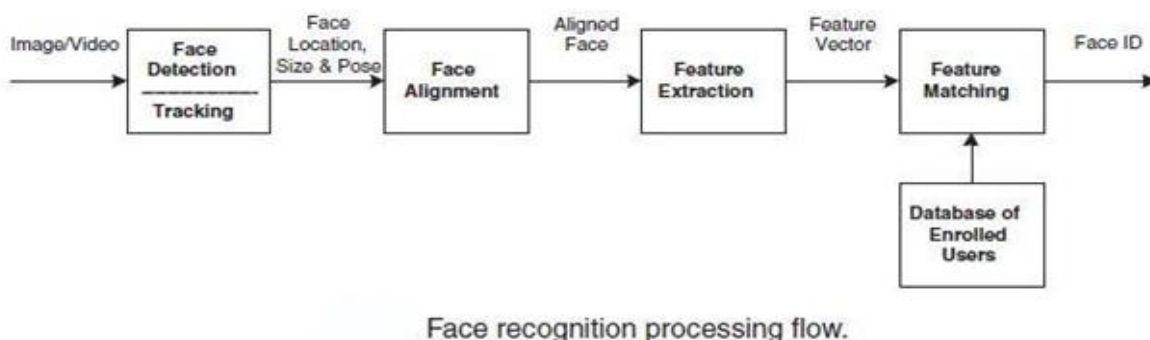


Рисунок 3 – Процесс распознавания лица

После получения снимка производится определение границ лица и предварительная обработка изображения. Далее выделяются признаки, которые кодируются. Конечным этапом является поиск и сравнение закодированных ключевых признаков с эталонными, содержащимися в базе данных [8].

Преимуществом является то, что не требуется физического контакта с устройством.

Главным недостатком можно назвать чувствительность к внешним факторам, из-за чего статистическая достоверность данного метода значительно снижается.

1.3.5 Распознавание человека по голосу

Распознавание человека по голосу позволяет идентифицировать человека исходя из уникальных характеристик речи. Голос является динамическим параметром, что означает, что его характеристики могут меняться в зависимости

от различных факторов: возраст, эмоциональное и физическое состояние, что делает его несколько неточным в определении человека [9].

К преимуществам можно отнести возможность удаленной идентификации человека, что позволяет получать доступ к различным услугам без физического присутствия.

Как уже упоминалось ранее, изменение голоса в зависимости от состояния человека, может привести к ошибочным результатам. Проблемы с шумом или потеря качества при передаче записи, помехи также являются причиной ложного срабатывания или отказа системы. Однако, одним из самых существенных недостатков является возможность имитировать голос, благодаря развитию нейросетей, что существенно снижает надежность и безопасность данного биометрического параметра.

1.3.6 Верификация подписи

Метод верификации подписи использует цифровую рукописную подпись человека, сделанная при помощи соответствующего программного обеспечения (графические планшеты, сенсорные дисплеи). Такая подпись используется для подтверждения подлинности и целостности электронного документа [10].

Алгоритм данного метода представляет собой несколько этапов: регистрация эталона вводом подписи несколько раз для сбора статистики. Определение и анализ уникальных особенностей, их количественное выражение, а также установление стандартных данных и допустимого отклонения от них. Следующим этапом является ввод подписи и выявляются ее особенности. Последним этапом можно назвать сравнение и оценивание особенностей эталона и введенной подписи, насколько они идентичны.

Достоинствами метода можно назвать невысокую стоимость и привычность для человека. Так как такой метод давно используется повсеместно и для пользователей он является обыденным.

Основным недостатком является изменчивость биометрического параметра. При воздействии некоторых факторов, а также изменении почерка со

временем, система может выдавать ошибочные отказы и ложные срабатывания. В случае, если понизить порог для данного метода, то это приведет к уязвимости в виде подделки подписи, что также нежелательно при идентификации пользователей.

1.3.7 Сравнительный анализ основных биометрических параметров

Сравнение методов биометрической идентификации происходит на основании двух основных параметров:

FAR (False Acceptance Rate) – коэффициент ложного пропуска. Этот параметр отображает процент возникновения ситуаций, когда система выдает положительный результат пользователю, который не зарегистрирован в системе.

FRR (False Rejection Rate) – коэффициент ложного отказа. Параметр показывает процент отказов в доступе пользователю, который является зарегистрированным в системе.

С помощью перечисленных выше коэффициентов можно определить какие методы наиболее эффективны. В таблице 1 представлены значения показателей качества [11, 12].

Таблица 1 – Оценка FAR и FRR для биометрических параметров

Биометрический параметр	FAR, %	FRR, %
Отпечаток пальца	0,001	0,6
Радужная оболочка глаза	0,00001	0,016
Геометрия руки	0,1	0,1
Распознавание лица	0,1	2,5
Распознавание по голосу	1	3
Верификация по подписи	1 – 3	3 – 5

Исходя из результатов, приведенных в таблице выше, на данный момент, радужная оболочка глаза является наиболее эффективным методом биометрической идентификации. Именно поэтому этот метод был выбран для данной работы.

1.4 Структура радужной оболочки глаза и ее свойства

Радужная оболочка глаза (РОГ) – почти круглая, слегка эллиптическая, подвижная диафрагма толщиной 0,2 – 0,4 мм и диаметром около 12 мм, разделяющая передний и задний отделы глаза. Радужная оболочка глаза находится позади роговицы, между передней и задней камерами глаза, и расположена перед хрусталиком (рисунок 4). Посередине располагается зрачок, у края которого проходит черная зубчатая линия, образованная переплетением сосудов. Она разделяет радужную оболочку глаза на зрачковый (шириной 1 мм) и ресничный (шириной 3 – 4 мм) края.

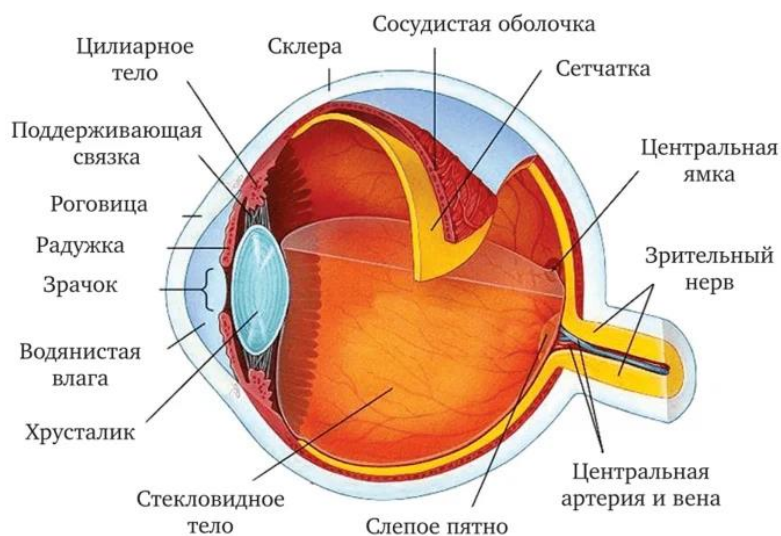


Рисунок 4 – Строение глаза

У самого края зрачка располагается сфинктер радужной оболочки глаза, который осуществляет сокращение зрачка. Между корнем радужной оболочки глаза и сфинктером размещается дилатор – мышца, способствующая расширению зрачка (рисунок 5). Совместная работа этих двух мышц – сфинктера

и дилататора – радужная оболочка глаза регулирует количество света, поступающего на сетчатку глаза.

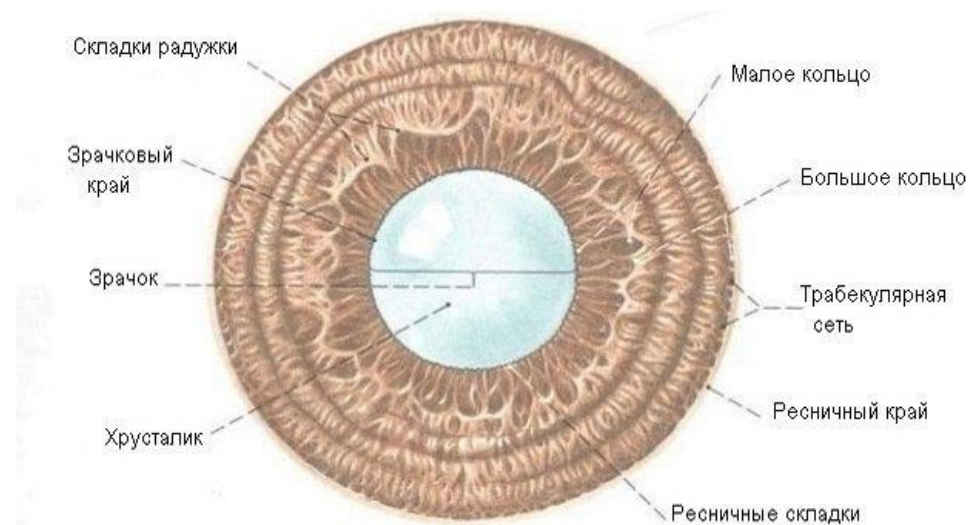


Рисунок 5 – Строение РОГ

Радужная оболочка глаза имеет губчатую структуру, состоящую из тонких радиальных перепонок, называемые трабекулами. Они формируются из плотной соединительной ткани, окружающей кровеносные сосуды радужной оболочки глаза. Между трабекулами располагаются углубления (лакуны или крипты). Параллельно зрачковому краю расположен невысокий зубчатый валик – зрачковый пояс, или малое кольцо радужной оболочки глаза, где радужная оболочка имеет наибольшую толщину, и здесь же встречаются особенно большие крипты.

Уникальная и сложная текстура радужной оболочки делает ее высокоинформативным биометрическим признаком, который можно использовать для надежной идентификации личности.

2 Постановка задачи и анализ требований

Перед разработкой программного продукта важно провести анализ существующих решений для поставленной задачи. В этом разделе будет рассмотрен процесс идентификации. Кроме того, будут проанализированы различные технологические решения и обоснован выбор каждого из них для решения конкретной задачи.

2.1 Общая схема процесса распознавание по радужной оболочке глаза

Общая схема распознавания по радужной оболочке глаза представлена на рисунке 6.

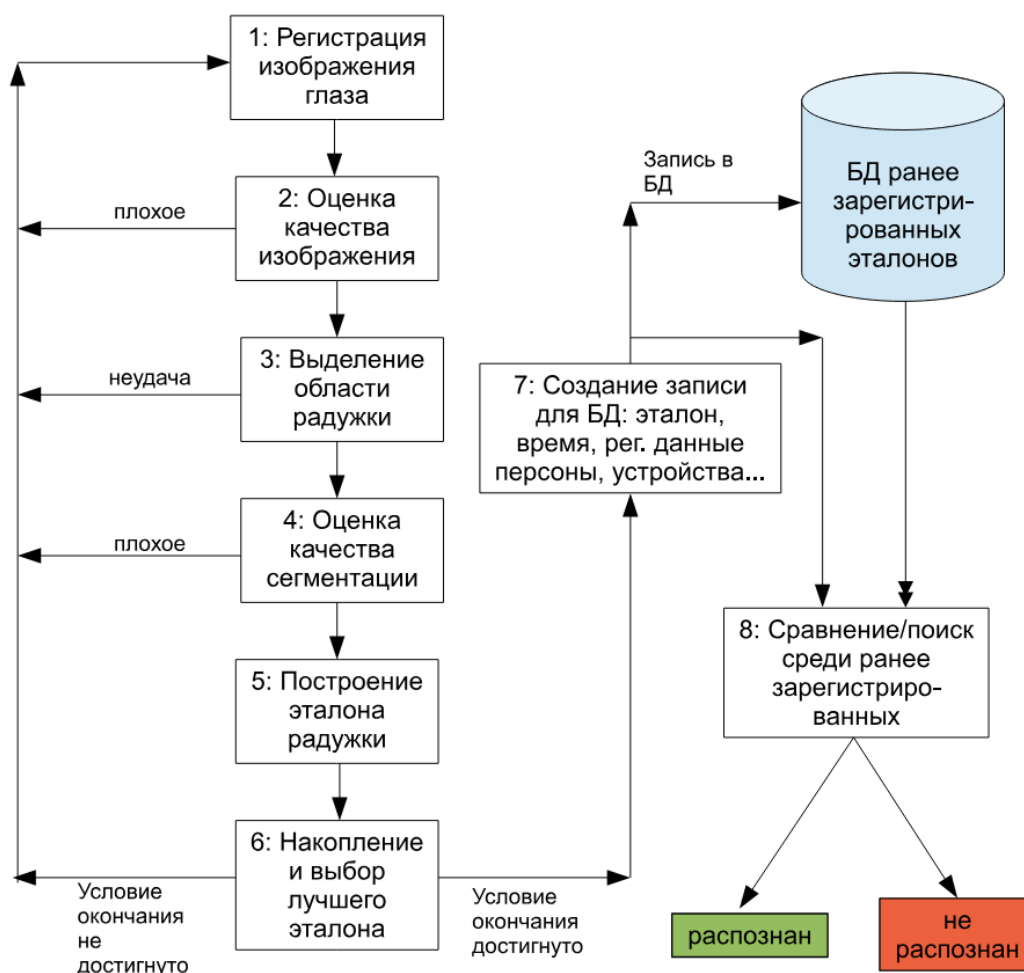


Рисунок 6 – Схема распознавания по РОГ

Блок 1. Регистрация изображения глаза осуществляется при помощи камеры, работающей в ближнем инфракрасном диапазоне (от 850 до 900 нм). Камера оснащена полосовым фильтром, который блокирует свет за пределами определённого диапазона, и инфракрасным осветителем с узким спектральным максимумом в этом диапазоне. Вместе, правильно подобранный фильтр и осветитель обеспечивают преимущество яркости осветителя над посторонними источниками света, такими как солнце, дневное небо и лампы накаливания. Это позволяет подавить блики от посторонних объектов, отражающихся в роговице и мешающих распознаванию рисунка радужной оболочки глаза. При этом уровень мощности инфракрасного излучения остаётся безопасным для здоровья. После захвата изображения проводится его предварительная обработка для удаления артефактов, таких как пространственная и яркостная дисторсия, что описано далее в этой главе.

Блок 2. Оценка качества изображения и Блок 4. Оценка качества сегментации предполагают использование методов определяющих, насколько подходит изображение для дальнейшей обработки, то есть создания эталонной радужной оболочки глаза для определенного пользователя. Применить способы выделения радужной оболочки можно и к некачественному изображению, получив при этом данные. Однако эталон не будет соответствовать реальным биометрическим данным из-за низкого качества фото, это означает, что с большей вероятностью радужная оболочка глаза не будет распознана при сравнении с эталоном, находящимся в базе данных, построенным для того же глаза. Чтобы не допускать подобных критичных ошибок, требуется использовать оценку качества изображения. Возможные методы, применяемые для блока 2, описаны в следующем подразделе.

Блок 3. Выделением области радужной оболочки глаза называется сегментация, заключающаяся в поиске на изображении темного, похожего на круг объекта. В сегментацию также входит и отсеечение еще более темного объекта, меньшего по размерам, чем радужная оболочка глаза – зрачка. Сложность в сегментации представляет определение век, которые могут

закрывать как верхнюю, так и нижнюю части радужной оболочки. В одном случае, применяется способ явного выделения век и отбрасывание ложных данных с закрытого участка. В другом же варианте веки не выделяются как таковые, а выявляются закрытые части путем сравнения нескольких снимков. Отсечение частей радужной оболочки глаза, закрытых ресницами может быть возможна только при анализе ряда изображений, так как текстура, создаваемая ресницами, практически не отличается от структуры радужной оболочки. Задача сегментации радужной оболочки глаза может быть решена многими способами, некоторые из них описаны в подразделе 2.3.

Блок 5. Построение эталона радужной оболочки глаза представляет собой выделение ключевых точек изображения. Зачастую, перед применением методов для выявления уникальных особенностей, сегментированную радужную оболочку глаза могут преобразовать из исходной полярной системы координат в декартову, данный метод описан в работе [13]. Способы вычисления информативных признаков будут описаны далее в подразделе 2.4.

Блок 6. Накопление и выбор лучшего эталона. Этот этап не всегда существует в системах распознавания. При использовании системы с регистрациями, используемыми только одно изображение, нет возможности выбрать лучший эталон из ряда снимков, представляется возможным взять только поданное на вход изображение и проводить действия по нему, либо отвергнуть, что происходит в блоке 2. При использовании систем, использующих несколько снимков, например, видеоряд, есть возможность выбрать наиболее подходящее фото. Такой подход значительно увеличивает точность распознавания. Однако, подход с видеорядом в данной работе не рассматривается.

Блок 7. Создание записи для базы данных. Этот блок отвечает за формирование записей о конкретном пользователе, основным параметром которых является бинарное кодирование ключевых точек.

Блок 8. Сравнение/поиск среди ранее зарегистрированных. Для идентификации пользователя имеет смысл использовать способы сравнения

двух радужек, а точнее, их бинарных представлений. Такие методы вычисляют коэффициент схожести двух изображений и исходя из результата, делают заключение: является пользователь зарегистрированным или нет.

2.2 Методы предобработки изображений

Предварительная обработка изображения играет важную роль в процессе идентификации пользователя по радужной оболочке глаза. Благодаря методам преобразования, убирается шум, увеличивается контрастность, происходит бинаризация для более качественной сегментации и выделения ключевых точек радужной оболочки глаза.

Выделяют несколько основных способов улучшения качества изображения радужной оболочки глаза для ее дальнейшего использования: удаление шумов, увеличение контрастности. Далее эти способы будут рассмотрены конкретнее с приведением реализующих их методов.

Для удаления шумов чаще всего применяются гауссово размытие и медианный фильтр.

Размытие Гаусса является популярным и весьма действенным методом для удаления шумов с изображения. Этот метод основан на свертке снимка с гауссовым ядром, что позволяет эффективно сглаживать текстуру, сохраняя основные контуры и детали. Математическое обоснование этого метода заключается в его способности действовать как фильтр низких частот. Формула функции Гаусса в двухмерном измерении имеет вид:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1)$$

где (x, y) – координаты в пространстве изображения;

σ – среднеквадратичное отклонение нормального распределения.

При применении данной функции, она формирует поверхность, контуры которой представляют собой концентрические окружности с нормальным распределением вокруг центральной точки.

Процесс свертки изображения I с гауссовым ядром G можно описать следующим образом:

$$I'(x, y) = (I * G)(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k I(x + i, y + j) * G(i, j), \quad (2)$$

где $I'(x, y)$ – значение пикселя в размытом изображении;

$I(x + i, y + j)$ – значение пикселя в исходном изображении;

$G(i, j)$ – значение гауссова ядра в точке (i, j) ;

k – размер гауссова ядра (половина его ширины).

Медианный фильтр – один из пространственных методов фильтрации использующий ядро, представляющее собой окрестность пикселя (также называемую окном). С математической точки зрения, окном называется M -окрестностью точки $A(x, y)$, где x и y – координаты пикселя в абсолютной системе координат изображения, которая является центральной для этой окрестности [14]. Медианный фильтр определяется следующим образом:

$$A(x, y) = \text{med} \{M(x, y)\}. \quad (3)$$

Медианная фильтрация эффективно устраняет помехи, воздействующие на отдельные пиксели. В качестве примера можно привести «битые» пиксели при цифровой съемке или «снеговой» шум, когда некоторые пиксели заменяются на пиксели с максимальной интенсивностью. Преимущество медианной фильтрации заключается в том, что «горячий» пиксель на темном фоне будет заменен темным пикселем, а не «размазан» по окрестности.

Увеличение контрастности достигается при помощи методов обрезание верхних значений и алгоритм контрастного ограниченного адаптивного выравнивания гистограммы.

Метод обрезания верхних значений заключается в замене слишком высоких по интенсивности пикселей на более темные, что позволяет сделать изображение более равномерным по яркости и более точно выделить темные области для корректной сегментации четко по границе радужной оболочки глаза и зрачка.

Алгоритм контрастного ограниченного адаптивного выравнивания гистограммы (CLAHE) является модификацией метода выравнивания

гистограммы, и по своему принципу отличается тем, что повышает контрастность локально. Данный метод делит изображение на блоки и контрастность изменяется внутри этого блока, независимо от соседних. Однако данный метод может усиливать мелкие текстуры и шум, что может помешать определению реальных ключевых точек. Также применение этого метода может искажать реальные границы, что также критично для биометрических систем.

2.3 Сегментация

Не менее существенное значение в идентификации пользователя имеет определение радужной оболочки глаза на изображении. Выделение на изображении области, соответствующей радужной оболочке глаза, относится к задаче сегментации, для которой используются различные методы. Наиболее используемые из них описаны ниже, в подглавах 2.3.1, 2.3.2 и 2.3.3.

2.3.1 Метод на основе активных контуров

Метод активных контуров был описан в 1987 году [15] и представляет собой деформируемые кривые (контур), которые динамически изменяют свою форму и координаты, стремясь минимизировать функционал энергии. Задачей данного метода является анализ изображения и, в частности, сегментация.

Функционал энергии к соответствию локального минимума контуру, выделяющему искомый сегментируемый объект. Энергетическая функция E_{snake} представляет собой сложение внутренней и внешней энергий. Внутренняя энергия E_{int} складывается из непрерывности и плавности контура. Внешняя E_{ext} энергия может включать в себя любые дополнительные силы, влияющие на контур, такие как внешние воздействия или ограничения.

Активный контур представляется параметрической кривой на плоскости изображения x, y , определенную следующим образом:

$$C(s): [0,1] \rightarrow \mathbb{R}^2, \quad s \mapsto C(s) = (x(s), y(s))^T, \quad (4)$$

где s – параметр кривой;

x, y – декартовы координаты на плоскости изображения.

Как уже было сказано ранее, функционал энергии представляет собой сумму внутренней и внешней энергии:

$$E_{snake}(C) = \int_0^1 (E_{int}(C(s)) + E_{ext}(C(s))) ds, \quad (5)$$

где $E_{int}(C(s))$ – внутренняя энергия;

$E_{ext}(C(s))$ – внешняя энергия

Внутренняя энергия определяется как

$$E_{int}(C(s)) = \alpha(s)|C'(s)|^2 + \beta(s)|C''(s)|^2, \quad (6)$$

где $\alpha(s)$ регулирует упругость контура;

$\beta(s)$ регулирует его жесткость.

Внешняя энергия имеет вид:

$$E_{ext}(C(s)) = P(C(s)), \quad (7)$$

где функция $P(C(s))$ – потенциал, определяющий внешнее воздействие на активный контур, основанное на характеристиках изображения $I(x, y)$.

2.3.2 Метод, основанный на преобразовании Хафа

Преобразование Хафа было разработано в 1962 году и использовалось для обнаружения объектов, которые могут быть определены в параметрической форме. Круговое преобразование может быть использовано для обнаружения окружностей на изображении. Уравнение окружности можно записать следующим образом:

$$r^2 = (x - a)^2 + (y - b)^2, \quad (8)$$

где r – радиус окружности;

a, b – координаты центра.

В параметрической форме точки уравнения окружности записываются в следующем виде:

$$\begin{cases} x = a + r \cos \theta \\ y = b + r \sin \theta \end{cases} \quad (9)$$

Вычисление требуемого круга производится с помощью рисования окружностей заданного радиуса в каждой точке краевого изображения.

Координата точки увеличивалась на единицу в местах прохождения периметра окружности. Это производилось для каждого круга, чтобы создать массив накопления. Однако, для применения этого метода требуется знание радиуса окружности, которую мы ищем. Так как радиусы зрачка и радужной оболочки неизвестны, вычисления должны быть использованы для диапазона радиусов. Для каждого радиуса сохраняются координаты и значения максимума. Радиус с наибольшим значением является наиболее вероятным радиусом радужной оболочки глаза.

2.3.3 Метод, основанный на интегро-дифференциальном операторе Даугмана

Интегро-дифференциальный оператор впервые был предложен Даугманом в работе «High Confidence Visual Recognition of Persons by a Test of Statistical Independence» [16]. Оператор предполагает, что зрачок и радужная оболочка глаза имеют круглую форму, и применяется для возможных координат центра радужной оболочки на изображении. Поиск двух координат и радиуса окружности производится при помощи следующего интегро-дифференциального оператора:

$$\max_{(r, x_0, y_0)} \left| G_\sigma(r) * \frac{d}{dr} \int_{(r, x_0, y_0)} \frac{I(x, y)}{2\pi r} ds \right|, \quad (10)$$

где $G_\sigma(r)$ – сглаживающая функция Гаусса;

$I(x, y)$ – значение интенсивности изображения в точке (x, y) ;

s – контур окружности с координатами (x_0, y_0) и радиусом r .

$\int_{(r, x_0, y_0)} \frac{I(x, y)}{2\pi r}$ является интегралом, вычисляемым вдоль окружности с радиусом r и центром (x_0, y_0) . Далее применяется $\frac{d}{dr}$ производная по радиусу, позволяющая найти места с резкими изменениями интенсивности, характерными для границ. Также применяется гауссово размытие для сглаживания значений радиусов. Это помогает в устранении шума, что делает процесс более устойчивым. Для определения значения радиуса r , а также координат центра (x_0, y_0) такие, при которых будет достигаться максимальное значение

выражения внутри модуля, применяется $\max_{(r, x_0, y_0)}$. Применение данной функции позволяет определить параметры окружности, соответствующей границе радужной оболочке глаза или зрачка. Данный метод имеет высокую точность, что и обуславливает его распространенность в система идентификации пользователей по биометрическим данным.

2.4 Выделение ключевых точек и сравнение признаков

Последним этапом в идентификации пользователей по радужной оболочке глаза является выделение ключевых точек биометрического образца и его сравнение, в результате которого определяется степень схожести, используемая для вынесения решения о доступе к системе. Оба этапа часто рассматриваются в едином контексте, поскольку они взаимосвязаны и значительно зависят друг от друга. Далее будут описаны методы как выделения ключевых точек (подглавы 2.4.1, 2.4.2), так и способы сравнения признаков (подглавы 2.4.3 – 2.4.5).

2.4.1 Дискретное вейвлет-преобразование с вейвлетом Хаара

Вейвлет Хаара был предложен в 1909 году и представляет собой набор простых вейвлетов, называемых признаками Хаара. Преобразование вычисляет разности интенсивности между соседними пикселями, что позволяет выделить уникальные текстурные и структурные особенности. Дискретное вейвлет преобразование разделяет входное изображение на четыре компонента: приближенное (уменьшенный в два раза исходный снимок) и три изображения, содержащих детализирующую информацию. Многоуровневое преобразование используется для извлечения данных о ключевых особенностях рисунка радужной оболочки глаза. В результате работы преобразования получается матрица, отображающая информацию о вертикальных, горизонтальных и диагональных контурах.

При использовании дискретного вейвлет преобразования получаются отрицательные и значения с плавающей точкой. Поэтому, чтобы получить бинарные значения, к полученному результату применяется квантование:

$$V(x)' = \begin{cases} 1, V(x) \geq 0, \\ 0, V(x) < 0, \end{cases} \quad (11)$$

где $V(x)$ – значение в исходной матрице.

Перевод значений с плавающей точкой к бинарному виду позволяет сравнивать и хранить данные без потери точности и минимальными затратами памяти.

2.4.2 Фильтр Габора

Одним из самых распространённых методов выделения уникальных особенностей радужной оболочки глаза является фильтр Габора. Данный метод эффективно применяется для анализа текстуры и выделения ключевых точек радужной оболочки глаза. Вейвлет Габора принимает на вход нормализованное, сегментированное изображение и представляет собой функцию, определенную следующим образом:

$$G(x, y) = \exp\left(-\frac{1}{2}\left[\frac{x_\phi^2}{\sigma_x^2} + \frac{y_\phi^2}{\sigma_y^2}\right]\right) \cos(2\pi\theta x_\phi), \quad (12)$$

где $x_\phi = x \cos \phi + y \sin \phi$;

$y_\phi = -x \sin \phi + y \cos \phi$;

θ – частотная модуляция фильтра;

ϕ – ориентация вейвлета относительно главных осей;

σ_x и σ_y – стандартные отклонения гауссова ядра по оси x и y соответственно.

Применение данного способа обеспечивает достаточную точность и надежность. После выделения ключевых точек необходима процедура бинарного кодирования для дальнейшего сравнения и хранения ключевых особенностей.

2.4.3 Расстояние Хэмминга

Вычисление соответствия двух радужек глаза чаще всего производится с помощью расстояния Хэмминга. При сравнении двух битовых шаблонов таким

способом, измеряется количество позиций, в которых соответствующие значения различны. Для двух бинарных матриц A и B одинакового размера, расстояние Хэмминга d_H рассчитывается по следующей формуле:

$$d = W(Z) = W(X \text{ xor } Y), \quad (13)$$

где X и Y – битовые коды.

При рассмотрении двух битовых кодов одинаковой радужной оболочки глаза расстояние Хэмминга должно быть равно нулю. Однако на практике такого не происходит из-за разницы в нормализации, а также наличие шумов, поэтому сравнение двух кодов одной и той же радужной оболочки глаза будет близко, но не равно нулю.

2.4.4 Коэффициент Жаккара

Коэффициент Жаккара является мерой сходства, предложенной в 1901 году. Для вычисления индекса требуется определить количество схожих позиций в двух бинарных матрицах и разделить его на общее количество значений в этих матрицах.

Для двух бинарных матриц A и B с одинаковыми размерами коэффициент Жаккара определяется на соответствующих строках и столбца. Этот индекс выражается следующим образом:

$$J(a_i, b_i) = \frac{\Sigma(a_i \cap b_i)}{\Sigma(a_i \cup b_i)}, \quad (14)$$

где a_i и b_i – i -е строки матриц A и B соответственно.

В базовом представлении результатом расчета индекса будет число от 0 до 1, где 0 – полное несоответствие. Данный метод отлично подходит для идентификации пользователей по радужной оболочке глаза, т. к. этот способ чувствителен к незначительным изменениям, что помогает с большей точностью определить человека, которому эта радужная оболочка глаза принадлежит.

2.5 Обоснование выбора подхода к идентификации

В результате рассмотрения вышеперечисленных методов для каждого этапа идентификации было принято решение относительно используемых способов. Для нормализации изображения будут использованы методы обрезание высоких значений и размытие Гаусса, так как в отличие от медианного фильтра, Гауссово размытие меньше влияет на важные текстуры и границы, но также успешно избавляется от шума на изображении. Для сегментации радужной оболочки глаза будет использоваться интегро-дифференциальный оператор Даугмана, а для определения зрачка – преобразование Хафа. Выделение уникальных особенностей и идентификация будут выполняться при помощи фильтра Габора и коэффициента Жаккара соответственно.

3 Проектирование информационной системы

В данном разделе будет рассмотрена разрабатываемая система, используемые данные. Будут построены диаграммы и модели для данной программы. Так же представлено обоснование выбора технологий, используемых в программном продукте.

3.1 Анализ используемых технологий разработки системы

3.1.1 Обоснование выбора языка программирования

Для написания системы, разработанной в рамках данной работы, использовался высокоуровневый язык программирования Python. Первая версия языка была разработана в конце 1980-х и выпущена в 1991 году Гвидо ван Россумом. Python является широко распространённым языком благодаря множеству преимуществ.

Простота и читаемость кода позволяют ускорить процесс разработки и упрощают поддержку системы. Легкость восприятия достигается за счет использования табуляции. Синтаксис языка также не представляет особой сложности, что минимизирует обращение к документации.

Динамическая типизация позволяет определять и изменять тип переменных во время работы программы, что значительно упрощает работу. Отсутствие привязки переменных к типам повышает гибкость кода, благодаря возможности одной функции работать с любыми типами данных, что в свою очередь, уменьшает дублирование кода.

Комфортной работе на языке python также способствует и большое количество стандартных и сторонних библиотек, позволяющих выполнять задачи, связанные с совершенно различными областями.

Недостатками языка можно назвать низкую скорость работы и высокое потребление памяти.

Исходя из перечисленных выше преимуществ, данный язык и был выбран для разработки системы идентификации по биометрическим данным.

3.1.2 Обоснование выбора среды разработки

В качестве среды разработки была выбрана Spyder – кроссплатформенная IDE, разработанная для научных задач и поставляемая вместе с Anaconda Distribution.

Основные преимущества IDE Spyder:

- 1) поддержка научных библиотек, предоставляющих инструменты для анализа данных и обработки изображений;
- 2) встроенная документация с примерами;
- 3) интерактивные графики для визуализации данных;
- 4) интерактивное выполнение отдельных строк или блоков кода;
- 5) анализ кода в реальном времени для мгновенного отображения ошибок, проблем или предупреждений [17].

Данная среда разработки отлично подходит для выполнения поставленной задачи, благодаря интеграции с научными библиотеками и функциональному интерфейсу.

3.1.3 Конфигурации компьютера

Реализация алгоритмов была выполнена на персональном компьютере, имеющем характеристик, представленные в таблице 2.

Таблица 2 – Характеристики ПК

Характеристика	Значение
Операционная система	Windows 11 Pro 64-bit.
Процессор	AMD Ryzen 7 3800X 8-Core Processor 3.90 GHz
Оперативная память	16,0 ГБ
Накопитель (SSD)	Intel 660p Series [SSDPEKNW010T8X1], чтение – 1800 Мбайт/сек, запись – 1800 Мбайт/сек

Этих характеристик достаточно для обеспечения стабильной работы программы идентификации пользователя по радужной оболочке глаза .

3.1.4 Датасет

Для проверки работоспособности системы был использован набор данных UTIRIS Тегеранского университета [18]. Этот датасет состоит из двух поднаборов, сделанных двумя разными способами: с помощью Canon EOS 10D и ISG Lightwise LW соответственно. Информация о наборе данных представлена в таблице 3.

Таблица 3 – Конфигурация набора данных

Характеристика	RGB	Infrared
Размер изображения (количество пикселей)	2048 × 1360	1000 × 776
Формат сжатия	JPEG	bmp
Цветовое пространство	RGB	Серый
Цветовой профиль	sRGB IEC61966 – 2.1	–
Фокусное расстояние	100	–

Данный набор является первым банком биометрических данных радужной оболочки глаза, зарегистрированный в 2007 году. Состоит из 1540 изображений как с правого, так и с левого глаз.

3.2 Описание проектируемой информационной системы

На вход системе подается изображение радужной оболочки глаза размером от 1000 × 776 до 2048 × 1360 пикселей. Пример входной радужной оболочки глаза изображен на рисунке 7.

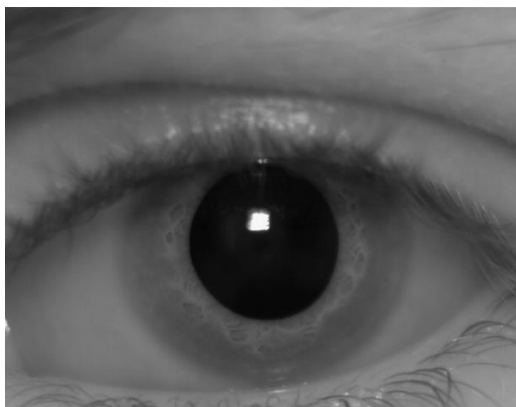


Рисунок 7 – Изображение радужной оболочки глаза

Далее система начинает поэтапно преобразовывать изображение для выполнения задачи.

Первым этапом происходит предварительная обработка изображения. Система применяет размытие гаусса и обрезание высоких значений, для получения подготовленного изображения и дальнейшей сегментации. Результат предварительной обработки представлен на рисунке 8. Важно отметить, что обработка изображения проводится только для сегментации радужной оболочки глаза и получения более четких границ. В то время как для выделения ключевых точек будет использоваться изначальное изображение с некоторой другой обработкой.



Рисунок 8 – Изображение радужной оболочки глаза после предобработки

Следующий этап – сегментация радужной оболочки глаза. В данном случае применяется интегро-дифференциальный оператор Даугмана для обнаружения радужной оболочки глаза и преобразование Хафа для определения зрачка. Результат сегментации представлен на рисунке 9.

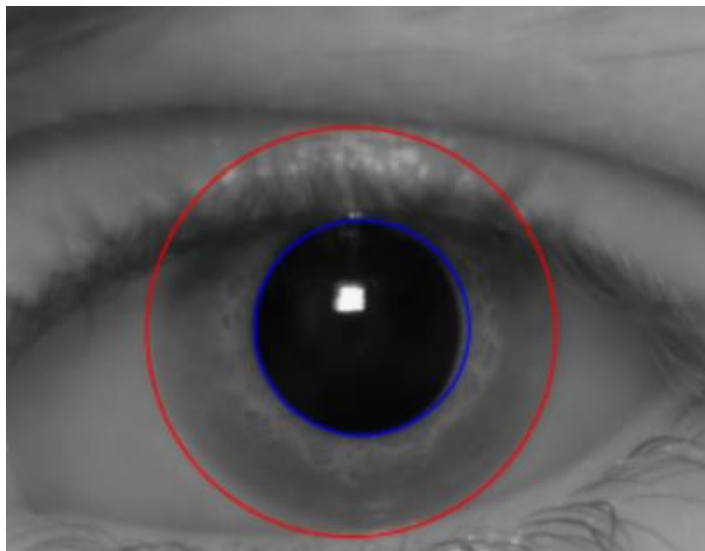


Рисунок 9 – Сегментация изображения

Перед определением ключевых точек выполняется преобразование, при котором сегментированное изображение разворачивается в выпрямленное изображение размером 352×245 . Результат представлен на рисунке 10.



Рисунок 10 – Выпрямленное изображение сегментированной радужной оболочки глаза

Для выделения ключевых особенностей применяется фильтр Габора, который возвращает код изображения и код маски. Маской называют бинарную матрицу, в которой предварительно пиксели с интенсивностью выше

определенного значения назначаются «True», а для остальных присваивается значение «False». После чего, с помощью вейвлетов Габора, вычисляются уникальные признаки и формируются код изображения и маска. Графическое представление кода изображено на рисунке 11.



Рисунок 11 – Графическое представление кода изображения

Графическое представление маски изображено на рисунке 12.



Рисунок 12 – Графическое представление маски

Заключительным этапом в идентификации по радужной оболочке глаза является сравнение эталона с образцом. Для этого применяется индекс Жаккара. Подготовленные в рамках предыдущих этапов код и маска образца передаются как параметр коэффициенту Жаккара, где второй параметр (также код и маска) поочередно сравнивается из базы данных, с уже имеющимися параметрами. В случае совпадения (если индекс Жаккара $> 0,42$), система отображает имя пользователя, имеющееся в базе данных, и выдает сообщение, что пользователь

зарегистрирован, в ином случае результатом является сообщение, что пользователь не идентифицирован.

3.3 Проектирование системы объектным методом

3.3.1 Диаграмма классов

Диаграмма классов – тип статической структурной диаграммы, описывающая структуру системы, отображая классы, атрибуты, методы и отношения между этими классами. Диаграмма классов представлена на рисунке 13.

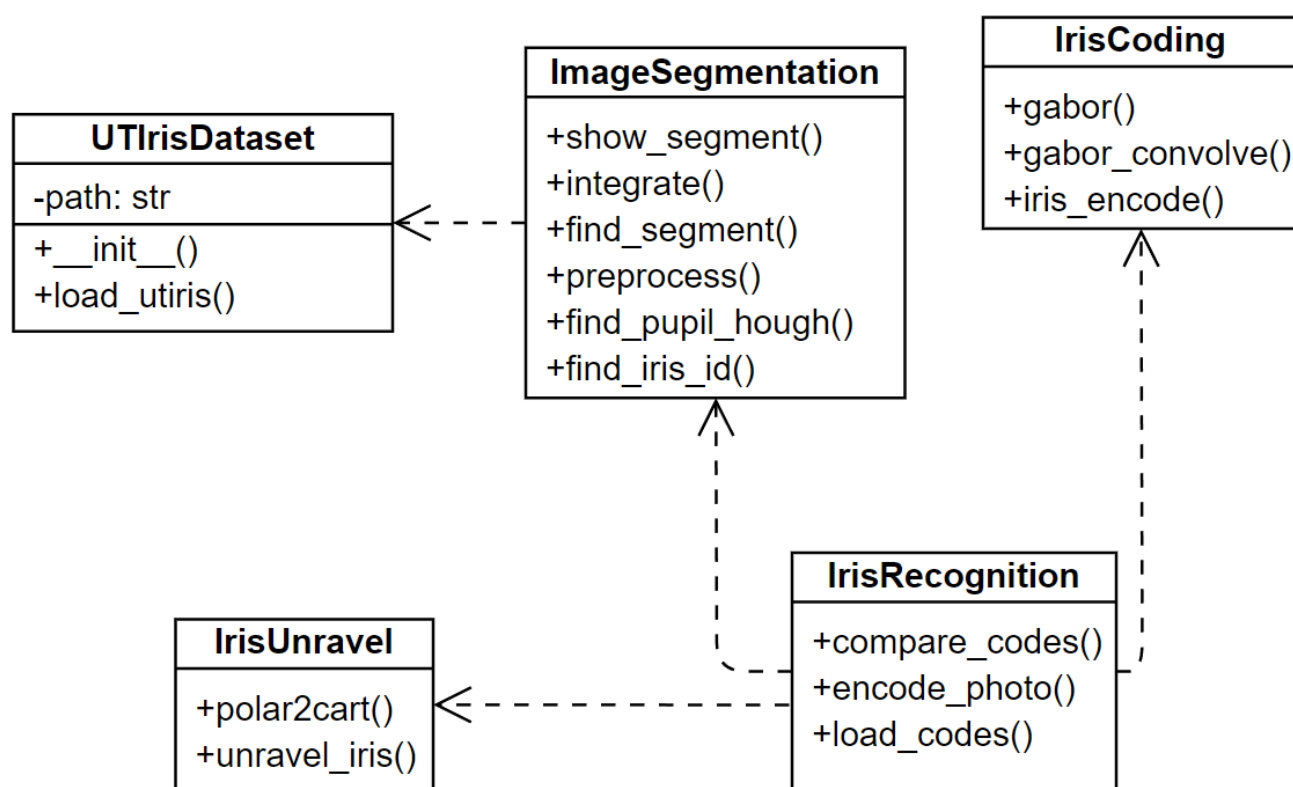


Рисунок 13 – Диаграмма классов

Из данной диаграммы видно, что все классы используют связь «Зависимость», которая отображает что один класс, может использовать методы другого класса. Например, класс **IrisRecognition** использует методы класса **IrisCoding**.

3.3.2 Диаграмма последовательности

Диаграмма последовательности используется для моделирования взаимодействия различных объектов системы для выполнения функции. Диаграмма визуально отображает порядок взаимодействия во времени, используя вертикальную ось.

Для полного отображения жизненного цикла, основных модулей и их взаимодействия была построена диаграмма последовательности, представленная на рисунке 14.

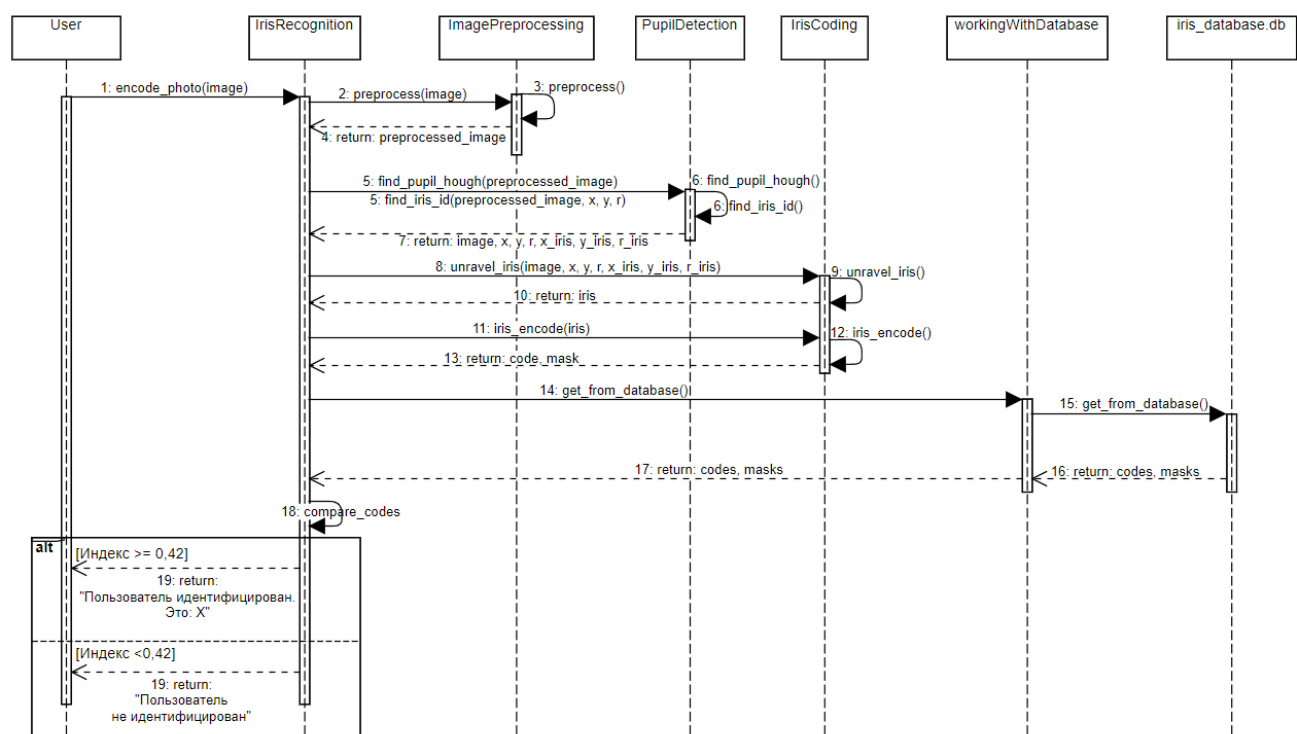


Рисунок 14 – Диаграмма последовательности

На данной диаграмме в качестве актеров представлены пользователь и основные модули системы, выполняющие обработку, сегментацию и сравнение изображения, задаваемого пользователем.

Таким образом, диаграмма отображает взаимодействие пользователя с системой, а также показывает внутреннее устройство системы.

3.3.3 Диаграмма деятельности

Диаграмма деятельности является UML-диаграммой, отображающей рабочие процессы с поддержкой выбора, итерации и параллелизма.

Благодаря диаграмме деятельности можно увидеть взаимодействия объектов в системе и описать последовательность действия. Диаграмма (рисунок 15) показывает вариант использования «Идентификация пользователя».

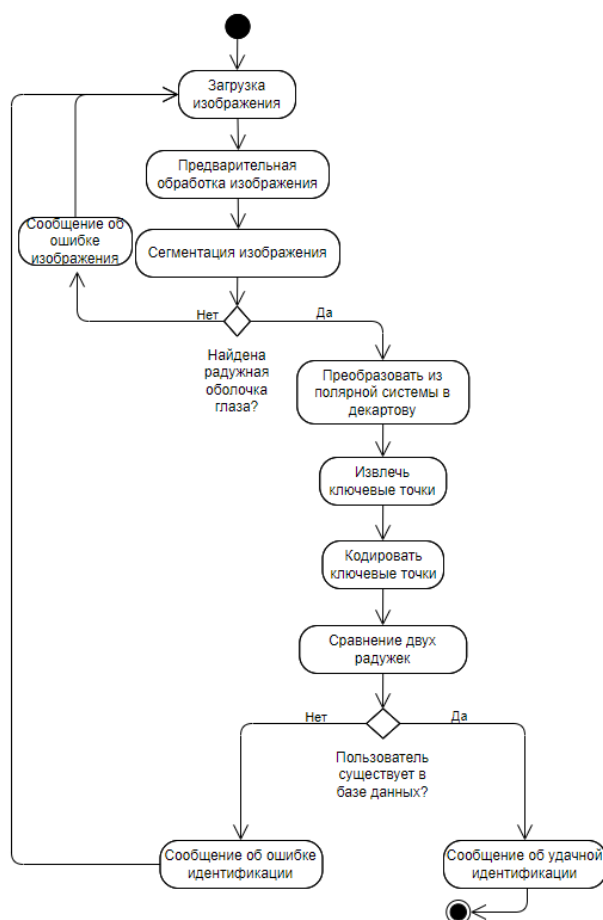


Рисунок 15 – Диаграмма деятельности

Первым действием происходит загрузка изображения. Далее происходит предварительная обработка и сегментация изображения. Если после этих действий, радужная оболочка глаза была найдена, то производятся последующие действия с изображением. Следующим шагом определяется существует ли пользователь в базе данных, если да, то происходит идентификация пользователя, в ином случае система выдает отказ.

4 Реализация системы

При создании модулей системы: для предварительной обработки и сегментации изображения, выделения ключевых точек, идентификации пользователя, использовалась ранее созданная схема, которая представлена на рисунке 13.

4.1 Реализация процесса подготовки изображения

Для реализации процесса подготовки изображения, как уже сказано ранее, были выбраны метод отсечения высоких значений и размытие Гаусса. Для сегментации используется интегро-дифференциальный оператор Даугмана.

Данные алгоритмы были описаны в модуле `segmentation`, в котором был определен класс `ImageSegmentation`. В этом классе основным методом обработки изображения является метод `preprocess`, возвращающий обработанное изображение.

Далее это изображение используется в методах `find_pupil_hough` и `find_iris_id`, которые определяют границы зрачка и радужной оболочки глаза соответственно, а также возвращают координаты и радиус этих объектов. Методы `find_segment` и `integrate` являются вспомогательными, и используются в методе `find_iris_id` для вычисления интегро-дифференциального оператора.

Программная реализация этого модуля представлена в приложении Б.

4.2 Реализация процесса выделения ключевых точек

Реализация процесса выделения ключевых точек производилась методом вейвлетов Габора в модуле `coding`. В данном модуле было реализовано два класса: `IrisUnravel` и `IrisCoding`.

Класс `IrisUnravel` отвечает за развертку изображения в прямоугольный вид, для более корректного выделения ключевых точек. Основным методом класса является `unravel_iris`, который возвращает, преобразованное в прямоугольный вид, сегментированное изображение. `Polar2cart` – вспомогательный метод, реализующий формулы перевода из полярной системы координат в декартову.

Класс `IrisCoding` реализует выделение ключевых точек фильтром Габора и содержит следующие методы: `iris_encode`, `gabor_convolve`, `gabor`. Метод `iris_encode` нужен для кодирования выпрямленного представления радужной оболочки с помощью вейвлетов Габора. Данный метод возвращает матрицу кода и маски изображения. Для извлечения уникальных особенностей радужной оболочки применяется метод `gabor_convolve`, вызывающий, в свою очередь, метод `gabor`, который вычисляет вейвлеты. Метод `gabor_convolve` возвращает два массива (реальная и мнимая часть) типа `ndarray`, которые содержат информацию о горизонтальных, вертикальных или диагональных структурах, а также, о фазовом сдвиге и прочих характеристиках.

Реализация описанных выше классов приведена в приложении Б.

4.3 Реализация процесса сравнения радужных оболочек глаза

Сравнение радужных оболочек глаза вычисляется с помощью Индекса Жаккара, который был реализован в классе `IrisRecognition` модуля `recognition`. Данный класс содержит методы `encode_photo`, `compare_codes`, `load_codes`. Все описанные выше методы для предварительной обработки, сегментации и выделения ключевых точек изображения вызываются в `encode_photo`. В качестве входных параметров передается кортеж матриц кода и маски, с использованием которых, будет вычисляться коэффициент схожести по индексу Жаккара, реализованном в методе `compare_codes`. Данный метод принимает на вход матрицы кода и маски радужной оболочки, которую требуется идентифицировать, а также матрицы кода и маски эталона, с которым происходит сравнение. Возвращает этот метод коэффициент сравнения в промежутке от 0.0 до 1.0, где чем выше коэффициент, тем более вероятно, что радужная оболочка глаза этого пользователя имеется в базе данных. В разработанной системе опытным путем было выявлено, что порогом является число 0.42.

Реализация данного класса также приведена в приложении Б.

5 Тестирование системы и анализ качества кода

Тестирование программного обеспечения – это процесс оценки качества разрабатываемого программного продукта, направленный на проверку его функционала, соответствия ожидаемым результатам. Тестированием является исследование программного обеспечения, цель которого – контроль соответствия фактического поведения системы ее ожидаемому поведению на основе набора тестов, выбранных определенным образом [19].

Для проверки качества разработанной системы было применено структурное тестирование. Проведен анализ качества кода с помощью определения различных метрик.

5.1 Структурное тестирование

Структурное тестирование (также известное как тестирование по принципу «белого ящика») – методология тестирования, проверяющая работу внутренних структур приложения. При таком подходе тестировщик имеет доступ к исходному коду программы и может анализировать его структуру, что позволяет понять, как различные компоненты программы взаимодействуют между собой. Данный метод предполагает проверку входных данных, путей прохождения кода и соответствие ожидаемых выходных данных.

Преимуществами такого подхода являются высокая точность, глубокий анализ, а также эффективное выявление дефектов за счет проверки отдельных компонентов системы.

Среди недостатков можно выделить трудоемкость из-за более тщательной проверки, а также необходимость знаний о внутренней структуре.

Тестирование по принципу белого ящика включает в себя следующие методы: покрытия операторов, покрытия условий, покрытия маршрутов.

Блок-схема алгоритма, на котором будет проводиться тестирование, представлена на рисунке 16.

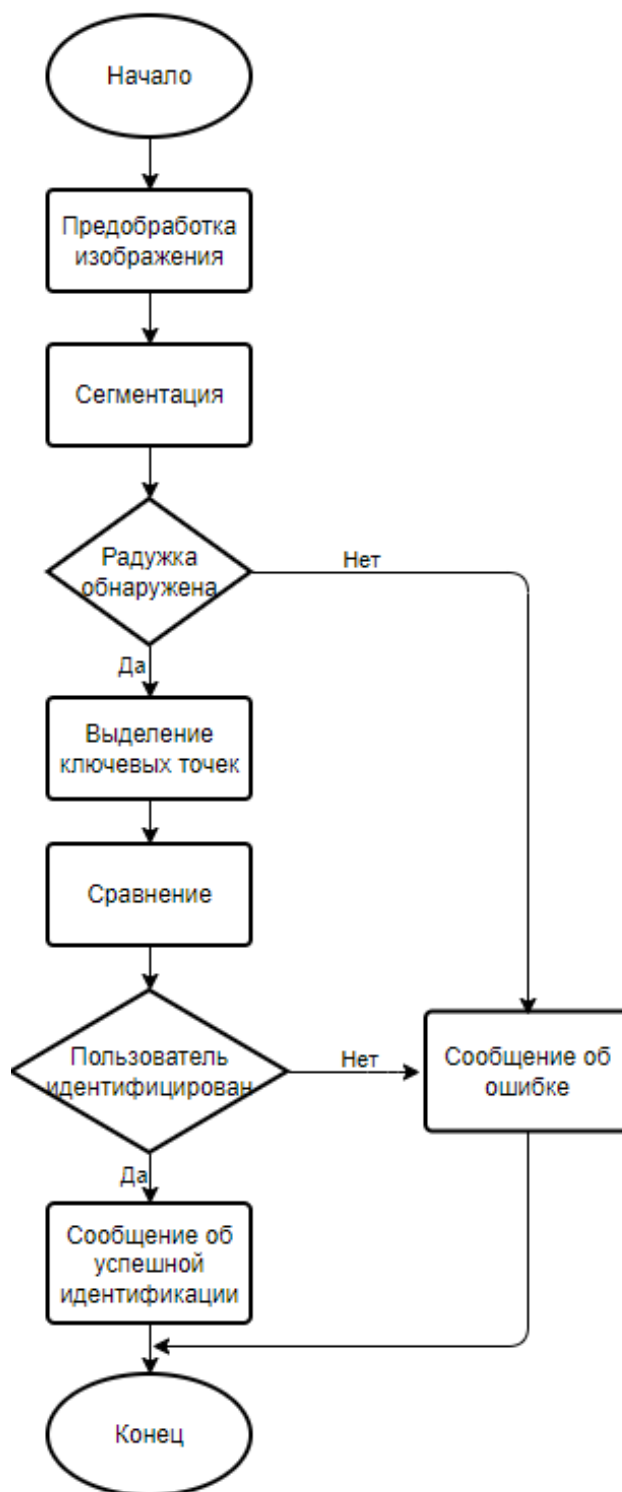


Рисунок 16 – Блок-схема

Последовательность операторов отображается с помощью управляющего графа программы, представленного на рисунке 17.

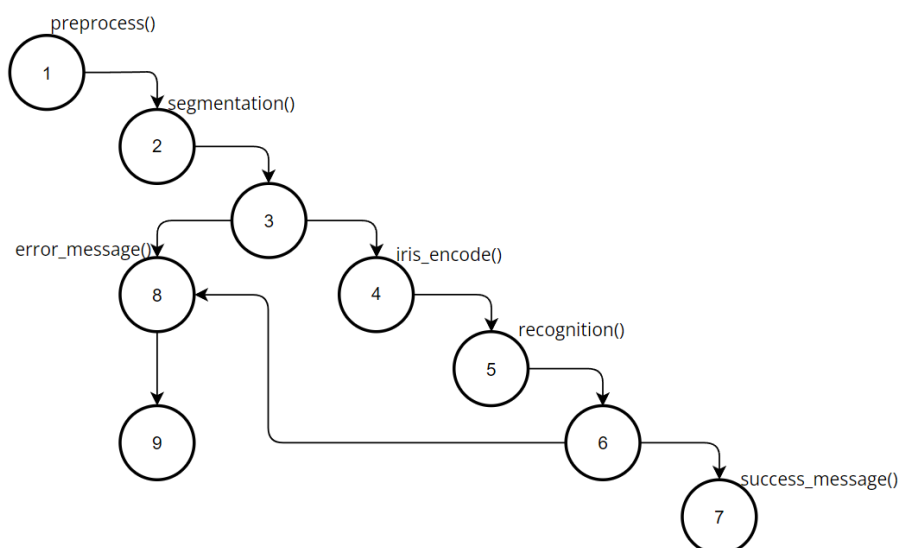


Рисунок 17 – Управляющий граф

Управляющий граф представлен в виде орграфа и показывает множество возможных путей выполнения программы.

5.1.1 Метод покрытия операторов

Метод покрытия операторов представляет собой выполнение каждого оператора в исходном коде как минимум один раз.

Для проверки выполнения всех операторов было составлено 2 теста, результаты которых, приведены в таблице 4.

Таблица 4 – Результаты тестирования методом покрытия операторов

Операторы	Тест	Фактический результат	Результат тестирования
1, 2, 3, 8, 9	Радужная оболочка глаза не найдена на изображении	Сообщение об ошибке.	Успешно.
1, 2, 3, 4, 5, 6, 7	Радужная оболочка глаза найдена на изображении. Пользователь идентифицирован.	Сообщение об успешной идентификации.	Успешно.

Все тесты были успешно пройдены. Полученные результаты соответствуют ожидаемым.

5.1.2 Метод покрытия условий

Метод покрытия условий заключается в проверке всех случаев, когда выполняется каждое из условий хотя бы один раз. Для проверки каждого условия было составлено 3 теста. Результаты этих тестов приведены в таблице 5.

Таблица 5 – Результаты тестирования методом покрытия условий

Условия	Тест	Фактический результат	Результат тестирования
1 – нет	Радужная оболочка глаза не найдена на изображении	Сообщение об ошибке.	Успешно.
1 – да, 2 – нет	Радужная оболочка глаза найдена на изображении. Пользователь не идентифицирован	Сообщение об ошибке.	Успешно.
1 – да, 2 – да	Радужная оболочка глаза найдена на изображении. Пользователь идентифицирован	Сообщение об успешной идентификации.	Успешно.

Все тесты пройдены успешно. Каждый оператор выполнен хотя бы раз. Если радужная оболочка глаза не найдена на изображении или пользователь не идентифицирован в системе, то выводится сообщение об ошибке, иначе – доступ разрешен.

5.1.3 Метод покрытия маршрутов

Суть данного метода заключается в том, чтобы составить такие тесты, при прохождении которых, будут проверены все возможные маршруты. Для выполнения всех маршрутов было составлено 3 теста. Результаты тестирования приведены в таблице 6.

Таблица 6 – Результаты тестирования методом покрытия маршрутов

Маршрут	Тест	Результат тестирования
1, 2, 3, 8, 9	Радужная оболочка глаза не найдена на изображении	Успешно.
1, 2, 3, 4, 5, 6, 8, 9	Радужная оболочка глаза найдена на изображении. Пользователь не идентифицирован	Успешно.
1, 2, 3, 4, 5, 6, 7	Радужная оболочка глаза найдена на изображении. Пользователь идентифицирован	Успешно.

Все тесты были пройдены успешно. Все маршруты были пройдены хотя бы один раз. Результаты тестирования соответствуют ожидаемым.

5.2 Метрики кода

Метрики программного обеспечения предоставляют важную информацию о различных аспектах кода и помогают принимать обоснованные решения для улучшения качества системы.

Для оценки качества разрабатываемой программы выделены два типа метрик:

- 1) метрика размера программ;
- 2) метрика сложности потока управления программ.

Вычисление метрик для конкретных частей кода приведены в следующих подглавах.

5.2.1 SLOC-метрика

SLOC-метрика – метрика программного кода, применяемая для измерения его объема с помощью подсчета количества строк. Данная метрика применяется для прогноза трудозатрат написания конкретного кода, либо для расчета производительности труда в уже имеющейся программе.

SLOC-метрика будет применена для модуля предварительной обработки и сегментации изображения.

Результаты метрики представлены в таблице 7.

Таблица 7 – Результаты SLOC-метрики

Число физических строк	133
Число логических строк	112
Число пустых строк	9
Число строк-комментариев	42

Важными понятиями в этой метрике являются физические и логические строки кода.

Физическими строками кода называют все строки, включая пустые и строки комментарии. Логическими строками называют количество команд в коде.

5.2.2 Метрика Холстеда

Метрика Холстеда вычисляет размер программы за счет подсчета операторов и операндов, используемых в программе.

Для вычисления данной метрики требуется четыре показателя кода:

- 1) n_1 – количество уникальных операторов программы;
- 2) n_2 – количество уникальных операндов программы;
- 3) N_1 – общее число операторов в программе;
- 4) N_2 – общее число операндом в программе.

Расчет каждой из этих характеристик для модуля *recognition* приведен в таблице 8.

Таблица 8 – Операнды и операторы программы

Операторы	Число вхождений	Операнды	Число вхождений
def	3	IrisRecognition	1
return	4	compare_codes	1
if	1	a	4
for	1	b	3
or	1	mask_a	3
in	2	mask_b	3
class	1	rotation	4
=	18	mask_c	2
-	5	img	2
*	3	ImageSegmentation	2
!=	1	image	3
+	2	x	2
<	1	y	2
-	5	r	2
()	22	x_iris	1
[]	4	y_iris	1
{}	1	r_iris	1
:	6	IrisUnravel	1
,	17	IrisCoding	1
.	9	encode_photo	1
		codes	4
		masks	2
		targets	2
$n_1 = 23$	$N_1 = 107$	$n_2 = 27$	$N_2 = 51$

Исходя из полученных характеристик, рассчитываются следующие оценки.

Длина программы рассчитывается по формуле:

$$N = N_1 + N_2.$$

Словарь программы рассчитывается по формуле:

$$n = n_1 + n_2.$$

Вычислительная сложность рассчитывается по формуле:

$$NV = N * \log_2 n.$$

Объем программы по Холстеду определяют объем кода алгоритма. Значение этой метрики для одной функции должно находиться в диапазоне от 20 до 1000.

В результате вычисления оценок метрики Холстеда были получены следующие значения:

- 1) $N = 158$;
- 2) $n = 50$;
- 3) $NV = 891,752$

В случае рассматриваемого программного кода объём по Холстеду находится в пределах нормы.

5.2.3 Цикломатическая метрика Мак-Кейба

Цикломатическая метрика Мак-Кейба используется для измерения сложности программы. Код представляется в виде управляющего ориентированного графа.

Цикломатическая метрика определяется основной формулой цикломатической сложности, помогающей оценить сложность тестирования и поддержания кода. Для вычисления этой метрики применяется формула:

$$Z(G) = e - v + 2p,$$

где e – число дуг ориентированного графа G ;

v – количество вершин;

p – число компонентов связности графа.

Цикломатическое число Мак-Кейба определяет минимальное количество тестов, необходимых для проверки по критерию «работает каждая ветвь».

Подсчет цикломатической сложности был проведен на основе управляющего графа, представленного на рисунке 17. Показатели для данного графа:

- 1) $e = 9$;
- 2) $v = 9$;
- 3) $p = 1$.

В результате вычисления цикломатической сложности Мак-Кейба на основе представленных показателей, было получено значение 2, что не превышает значение 10, а значит, программа имеет низкую сложность и легко тестируется.

Для вычисления индекса сопровождаемости, который рассчитывается без учета количества комментариев и принимает значения от 0 до 100, применяется формула:

$$MI = \max(0, (171 - 5.2 * \ln(HV) - 0.23 * CC - 16.2 * \ln(LoC)) * 100 / 171),$$

где HV (Halsted Volume) – вычислительная сложность по Холстеду;

CC (Cyclomatic Complexity) – структурная сложность кода, т. е. количество различных ветвей в коде;

LoC (Lines of Code) – количество строк кода.

Получены результаты:

$$CC = 9;$$

$$LOC = 82;$$

$$HV = 891,752.$$

В результате вычисления, индекс сопровождаемости равен 15. Таким образом, программа имеет приемлемый индекс сопровождаемости.

Заключение

В результате выполнения выпускной квалификационной работы была разработана информационная система для идентификации пользователей по радужной оболочке глаза.

В процессе выполнения дипломного проекта были проанализированы существующие биометрические данные, проведено их сравнение. Также был рассмотрен основной алгоритм, применяемый при распознавании пользователей по радужной оболочке глаза. Для каждого этапа идентификации выделены и сравнены алгоритмы.

Программная реализация выполнена на языке программирования Python, в среде выполнения Spyder.

Проектирование программных средств было проведено с использованием унифицированного языка моделирования UML.

Полученная система была протестирована с использованием метода белого ящика и показала, что функционирует корректно.

Разработанная информационная система достаточно гибкая и имеет перспективу для дальнейшего развития. Кроме этого, данная разработка может быть внедрена в работу компаний, для которых важным фактором является безопасность.

Поставленная цель и задачи были успешно выполнены.

Список использованных источников

1. Мао, В. Современная криптография (теория и практика) / В. Мао – М.: Издательский дом «Вильямс», 2005. – 768 с.
2. Joseph, N. Biometric Recognition: Challenges and Opportunities / N. Joseph, I. Lynette. – National Academies Press, 2010. – 165 с.
3. Clarke, R. Human identification in information systems: Management challenges and public policy issues / R. Clarke // Information Technology & People. – 1994. – №7 (4). – С.6 – 37.
4. Попова, Л. В. Биометрические идентификации личности человека / Л. В. Попова, Т. С. Потопольская // Актуальные вопросы охраны общественного порядка и административной деятельности полиции: материалы межведомственной научно-практической конференции, Волгоград, 16 октября 2019 года. – М.: ООО «Издательство «Спутник+», 2020. – С. 41 – 44.
5. Желтов, Г. И. Идентификация по радужной оболочке глаза: проблемы фоторегистрации / Г. И. Желтов // Вестник Московского государственного технического университета им. Н.Э. Баумана. – 2012. – №3 (3). – С. 6.
6. Шаханова, М. В. Биометрические средства идентификации личности / М. В. Шаханова, Д. С. Шаханова // Инновации. Наука. Образование. – 2021. – №25. – С. 563 – 573.
7. Прудник, А. М. Биометрические методы защиты информации / А. М. Прудник, Г. А. Власова, Я. В. Рощупкин. – Минск : БГУИР, 2014. – 123 с.
8. Анализ существующих подходов к распознаванию лиц [Электронный ресурс]. – URL: <https://habr.com/ru/companies/synesis/articles/238129> (дата обращения 18.05.2024).
9. Матвеев, Ю. Н. Технологии биометрической идентификации личности по голосу и другим модальностям / Ю. Н. Матвеев // Вестник МГТУ им. Н. Э. Баумана. Сер. «Приборостроение». – 2012. – №3 (3). – С. 315 – 319.

10. Вдовкина, С. А Библиотека для онлайн-верификации рукописной подписи / С. А. Вдовкина, С. В. Шибанов // Вестник Пензенского государственного университета. – 2022. – №2 (38). – С. 76 – 84
11. Вихман, В. В. Биометрические системы контроля и управления доступом в задачах защиты информации: учебно-метод. пособие / В.В. Вихман, А.А. Якименко. – Новосибирск: Изд-во НГТУ, 2016. – 54 с.
12. Лебеденко, Ю. И. Биометрические системы безопасности / Ю. И. Лебеденко. – Тула: Изд-во ТулГУ, 2012. – 160 с.
13. Daugman, J. How iris recognition works / J. Daugman // IEEE Trans. Circuits Syst. Video Technol. – 2004. – №14 (1). – С. 21 – 30.
14. Андрущенко, А. А. Метод фильтрации изображений на основе медианного фильтра, реализованного с использованием гистограммы цветов / А. А. Андрущенко, К. С. Макаров // Интеллектуальные информационные системы: Теория и практика : Сборник научных статей по материалам I Всероссийской конференции, Курск, 20–22 апреля 2020 года. Часть 1. – Курск: Курский государственный университет, 2020. – С. 6 – 14.
15. Kass, M. Snakes: Active contour models. / M. Kass, A. Witkin, D. Terzopoulos // International Journal of Computer Vision. – 1988. – №1 (4). – С. 321 – 331.
16. Daugman, J. High Confidence Visual Recognition of Persons by a Test of Statistical Independence / J. Daugman // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1993. – №15 (11). – С. 1148 – 1161.
17. Spyder's Documentation [Электронный ресурс]. – URL: <https://docs.spyder-ide.org/current/panes/index.html> (дата обращения 29.05.2024).
18. UTIRIS [Электронный ресурс]. – URL: <https://utiris.wordpress.com/> (дата обращения 29.05.2024).
19. Майерс, Г. Искусство тестирования программ. / Г. Майерс, Т. Баджетт, К. Сандлер. – Диалектика, 2012. – 271 с.

Приложение А
(справочное)
Глоссарий

Биометрическая аутентификация – процесс подтверждения пользователя путем проверки биометрического образа посредством применения методов и протоколов идентификации.

Информационная система (ИС) – совокупность программно-аппаратных средств, предназначенных для хранения, обработки и выдачи информации в соответствии с требованиями предметной области, которую моделирует ИС.

UML (Unified Modeling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

FAR (False Acceptance Rate) – коэффициент ложного пропуска, вероятность ложной идентификации, то есть вероятность того, что система по ошибке признает подлинность пользователя, не зарегистрированного в системе.

FRR (False Rejection Rate) – коэффициент ложного отказа, вероятность того, что система по ошибке откажет пользователю в идентификации, зарегистрированном в системе.

Приложение Б
(рекомендуемое)
Листинг программы

Листинг Б.1 – Код модуля «segmentation.py»

```

from datasets import load_utiris
import matplotlib.pyplot as plt
import cv2
import math
import numpy as np
import random
from scipy.ndimage.filters import gaussian_filter

def show_segment(img, x, y, r, x2=None, y2=None, r2=None):
    """Shows an image with pupil and iris marked with circles.
    :param img: Image of an eye
    :param x: x coordinate of a segment
    :param y: y coordinate of a segment
    :param r: radius of a segment
    :param x2: x coordinate of another segment (optional, can be None)
    :param y2: y coordinate of another segment (optional, can be None)
    :param r2: r coordinate of another segment (optional, can be None)
    """
    ax = plt.subplot()
    ax.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    segment = plt.Circle((x, y), r, color='b', fill=False)
    ax.add_artist(segment)
    if r2 is not None:
        segment2 = plt.Circle((x2, y2), r2, color='r', fill=False)
        ax.add_artist(segment2)
    plt.show()

def integrate(img, x0, y0, r, arc_start=0, arc_end=1, n=8):
    """Calculates line integral in the image.
    :param img: Image of an eye
    :param x0: x coordinate of the centre of the segment
    :param y0: y coordinate of the centre of the segment
    :param r: radius of the segment
    :param arc_start: From which point on the arc should the calculation
    start
    :param arc_end: At which point on the arc should the calculation end
    :param n: Number of points at which intergral is calculated along the
    line (the more points, the more accurate the
    result is)
    :return: Line integral value
    """
    theta = 2 * math.pi / n
    integral = 0
    for step in np.arange(arc_start * n, arc_end * n, arc_end -
    arc_start):
        x = int(x0 + r * math.cos(step * theta))
        y = int(y0 + r * math.sin(step * theta))
        integral += img[x, y]
    return integral / n

"""def integrate(img, x0, y0, r, arc_start=0, arc_end=1,
num_samples=1000):
    integral = 0
    for _ in range(num_samples):
        theta = random.uniform(arc_start * 2 * math.pi, arc_end * 2 *
math.pi)
        x = int(x0 + r * math.cos(theta))

```

```

        y = int(y0 + r * math.sin(theta))
        if 0 <= x < img.shape[0] and 0 <= y < img.shape[1]:
            integral += img[y,x]
        return integral/num_samples"""
def find_segment(img, x0, y0, minr=0, maxr=500, step=1, sigma=5.,
center_margin=30, segment_type='iris', jump=1):
    """Finds the segment (pupil or iris) in the image.

    :param img: Image of an eye
    :param x0: Starting x coordinate
    :param y0: Starting y coordinate
    :param minr: Minimal radius
    :param maxr: Maximal radius
    :param step: The difference between two consecutive radii in the
search space
    :param sigma: The amount of blur of integral values before selecting
the optimal radius
    :param center_margin: The maximum distance from x0, y0 reached to
find the optimal segment centre
    :param segment_type: Either 'iris' or 'pupil' used to optimize the
search
    :param jump: The difference between two consecutive segment centres
in the search space
    :return: x, y of the segment centre, radius of the segment and
integral value matching the optimal result
    """
    max_o = 0
    max_l = []
    if img.ndim > 2:
        img = img[:, :, 0]
    margin_img = np.pad(img, maxr, 'edge')
    x0 += maxr
    y0 += maxr
    for x in range(x0 - center_margin, x0 + center_margin + 1, jump):
        for y in range(y0 - center_margin, y0 + center_margin + 1, jump):
            if segment_type == 'pupil':
                l = np.array([integrate(margin_img, y, x, r) for r in range(minr, maxr,
step)])
            else:
                l = np.array([integrate(margin_img, y, x, r, 1 / 8, 3 / 8, n=8)
integrate(margin_img, y, x, r, 5 / 8, 7 / 8, n=8)
for r in range(minr + abs(x0 - x) + abs(y0 - y), maxr, step)])
                l = (l[2:] - l[:-2]) / 2
                l = gaussian_filter(l, sigma)
                l = np.abs(l)
                max_c = np.max(l)
                if max_c > max_o:
                    max_o = max_c
                    max_l = l
                    max_x, max_y = x, y
                    r = np.argmax(l) * step + minr + abs(x0 - x) + abs(y0 -
y)

    return max_x - maxr, max_y - maxr, r, max_l

def preprocess(image):
    """Preprocesses the image to enhance the process of finding the iris.
Crops high values of the image and blurs it.

```

```

:param image: Image of an eye
:return: Preprocessed image
"""
img = image[:, :, 0].copy()
img[img > 225] = 30
return cv2.medianBlur(img, 21)

def find_pupil_hough(img):
    """Finds the pupil using Hugh transform.

    :param img: Image of an eye
    :return: x, y coordinates of the centre of the pupil and its radius
    """
    circles = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 1, 20,
                                param1=50, param2=30, minRadius=10,
maxRadius=200)
    if circles is not None and len(circles) > 0:
        circles = np.uint16(np.around(circles))
        return circles[0, 0][0], circles[0, 0][1], circles[0, 0][2]
    return None

def find_iris_id(img, x, y, r):
    """Finds the iris in the image usind integro-differential operator.

    :param img: Image of an eye
    :param x: Starting x coordinate
    :param y: Starting y coordinate
    :param r: Starting radius
    :return: x, y coordinates of the centre of the iris and its radius
    """
    x, y, r, l = find_segment(img, x, y, minr=max(int(1.25 * r), 100),
                                sigma=5, center_margin=30, jump=5)
    x, y, r, l = find_segment(img, x, y, minr=r - 10, maxr=r + 10,
                                sigma=2, center_margin=5, jump=1)
    return x, y, r

```

Листинг Б.2 – Код модуля «recognition.py»

```

import numpy as np
from segmentation import *
from coding import *
from workingWithDatabase import *
import os

class IrisRecognition:
    @staticmethod
    def compare_codes(a, b, mask_a, mask_b, rotation=False):
        """Compares two codes and calculates Jaccard index.

        :param a: Code of the first iris
        :param b: Code of the second iris
        :param mask_a: Mask of the first iris
        :param mask_b: Mask of the second iris

```

```

        :param rotation: Maximum cyclic rotation of the code. If this
        argument is greater than zero, the function will
            return minimal distance of all code rotations. If this
        argument is False, no rotations are calculated.

        :return: Distance between two codes.
        """
        if rotation:
            d = []
            for i in range(-rotation, rotation + 1):
                c = np.roll(b, i, axis=1)
                mask_c = np.roll(mask_b, i, axis=1)
                d.append(np.sum((a != c) * mask_a * mask_c) / np.sum(mask_a
* mask_c))
            return np.min(d)
        return np.sum((a != b) * mask_a * mask_b) / np.sum(mask_a *
mask_b)
    @staticmethod
    def encode_photo(image):
        """Finds the pupil and iris of the eye, and then encodes the
        unravelled iris.

        :param image: Image of an eye
        :return: Encoded iris (code, mask)
        :rtype: tuple (ndarray, ndarray)
        """
        img = ImageSegmentation.preprocess(image)
        if ImageSegmentation.find_pupil_hough(img) != None:
            x, y, r = ImageSegmentation.find_pupil_hough(img)
            x_iris, y_iris, r_iris = ImageSegmentation.find_iris_id(img,
x, y, r)
            iris = IrisUnravel.unravel_iris(image, x, y, r, x_iris,
y_iris, r_iris)
            return IrisCoding.iris_encode(iris)
        else:
            return error(False)

    @staticmethod
    def load_codes():
        """Loads codes saved by save_codes function.

        :return: Codes, masks, and targets of saved images
        :rtype: tuple (ndarray, ndarray, ndarray)"""

        codes = []
        masks = []
        targets = []
        i = 0
        while os.path.isfile('codes\\code{}.npz'.format(i)):
            code = np.load('codes\\code{}.npz'.format(i))
            if code.shape[0] != 1:
                codes.append(code)
                masks.append(np.load('codes\\mask{}.npz'.format(i)))
                targets.append(np.load('codes\\target{}.npz'.format(i)))
            i += 1
        return np.array(codes), np.array(masks), np.array(targets)

    def error(flag):

```

```

        if flag:
            print("Ошибка!", "Радужная оболочка глаза найдена.
Пользователь не идентифицирован", sep="\n")
        else:
            print("Ошибка!", "Радужная оболочка глаза не найдена.",
sep="\n")
if __name__ == '__main__':
    dataset = workingWithDatabase()
    for i in range (workingWithDatabase.lenght()):
        flag = False
        data = dataset.get_from_database()
        image = cv2.imread(data[1])
        code, mask = IrisRecognition.encode_photo(image)
        code2, mask2 = data

        if IrisRecognition.compare_codes(code, code2, mask, mask2,
rotation=10) < 0.42:
            print("Пользователь идентифицирован")
            flag = True
            break
    if flag == False:
        error(True)

```

Листинг Б.3 – Код модуля «coding.py»

```

import numpy as np
from skimage.util import view_as_blocks

class IrisUnravel:
    @staticmethod
    def polar2cart(r, x0, y0, theta):
        x = r * np.cos(theta) + x0
        y = r * np.sin(theta) + y0
        return x, y

    @staticmethod
    def unravel_iris(img, xp, yp, rp, xi, yi, ri, phase_width=300,
iris_width=150):
        """Unravels the iris from the image and transforms it to a
straightened representation.

        :param img: Image of an eye
        :param xp: x coordinate of the pupil centre
        :param yp: y coordinate of the pupil centre
        :param rp: Radius of the pupil
        :param xi: x coordinate of the iris centre
        :param yi: y coordinate of the iris centre
        :param ri: Radius of the iris
        :param phase_width: Length of the transformed iris
        :param iris_width: Width of the transformed iris
        :return: Straightened image of the iris
        :rtype: ndarray
        """
        if img.ndim > 2:
            img = img[:, :, 0].copy()
            iris = np.zeros((iris_width, phase_width))
            theta = np.linspace(0, 2 * np.pi, phase_width)

```

```

        for i in range(phase_width):
            begin = IrisUnravel.polar2cart(rp, xp, yp, theta[i])
            end = IrisUnravel.polar2cart(ri, xi, yi, theta[i])
            xspace = np.linspace(begin[0], end[0], iris_width)
            yspace = np.linspace(begin[1], end[1], iris_width)
            iris[:, i] = [255 - img[int(y), int(x)]
                          if 0 <= int(x) < img.shape[1] and 0 <= int(y) <
img.shape[0]
                          else 0
                          for x, y in zip(xspace, yspace)]

    return iris
class IrisCoding:
    @staticmethod
    def gabor(rho, phi, w, theta0, r0, alpha, beta):
        """Calculates gabor wavelet.

        :param rho: Radius of the input coordinates
        :param phi: Angle of the input coordinates
        :param w: Gabor wavelet parameter (see the formula)
        :param theta0: Gabor wavelet parameter (see the formula)
        :param r0: Gabor wavelet parameter (see the formula)
        :param alpha: Gabor wavelet parameter (see the formula)
        :param beta: Gabor wavelet parameter (see the formula)
        :return: Gabor wavelet value at (rho, phi)

        return np.exp(-w * 1j * (theta0 - phi)) * np.exp(-(rho - r0) ** 2
/ alpha ** 2) * \
            np.exp(-(phi - theta0) ** 2 / beta ** 2)
        """
        theta_diff = np.mod(phi - theta0 + np.pi, 2 * np.pi) - np.pi #
Обеспечение правильного углового расстояния
        return np.exp(-w * 1j * theta_diff) * np.exp(-(rho - r0) ** 2 /
alpha ** 2) * \
            np.exp(-theta_diff ** 2 / beta ** 2)

    @staticmethod
    def gabor_convolve(img, w, alpha, beta):
        """Uses gabor wavelets to extract iris features.

        :param img: Image of an iris
        :param w: w parameter of Gabor wavelets
        :param alpha: alpha parameter of Gabor wavelets
        :param beta: beta parameter of Gabor wavelets
        :return: Transformed image of the iris (real and imaginary)
        :rtype: tuple (ndarray, ndarray)
        """
        rho = np.array([np.linspace(0, 1, img.shape[0]) for i in
range(img.shape[1])]).T
        x = np.linspace(0, 1, img.shape[0])
        y = np.linspace(-np.pi, np.pi, img.shape[1])
        xx, yy = np.meshgrid(x, y)
        return rho * img * np.real(IrisCoding.gabor(xx, yy, w, 0, 0.5,
alpha, beta).T), \
            rho * img * np.imag(IrisCoding.gabor(xx, yy, w, 0, 0.5,
alpha, beta).T)

    @staticmethod
    def iris_encode(img, dr=15, dtheta=15, alpha=0.4):

```

```

        """Encodes the straightened representation of an iris with gabor
        wavelets.

        :param img: Image of an iris
        :param dr: Width of image patches producing one feature
        :param dtheta: Length of image patches producing one feature
        :param alpha: Gabor wavelets modifier (beta parameter of Gabor
        wavelets becomes inverse of this number)
        :return: Iris code and its mask
        :rtype: tuple (ndarray, ndarray)
        """
        # mean = np.mean(img)
        # std = img.std()
        mask = view_as_blocks(np.logical_and(100 < img, img < 230), (dr,
        dtheta))
        norm_iris = (img - img.mean()) / img.std()
        patches = view_as_blocks(norm_iris, (dr, dtheta))
        code = np.zeros((patches.shape[0] * 3, patches.shape[1] * 2))
        code_mask = np.zeros((patches.shape[0] * 3, patches.shape[1] *
        2))
        for i, row in enumerate(patches):
            for j, p in enumerate(row):
                for k, w in enumerate([8, 16, 32]):
                    wavelet = IrisCoding.gabor_convolve(p, w, alpha, 1 /
        alpha)
                    code[3 * i + k, 2 * j] = np.sum(wavelet[0])
                    code[3 * i + k, 2 * j + 1] = np.sum(wavelet[1])
                    code_mask[3 * i + k, 2 * j] = code_mask[3 * i + k, 2
        * j + 1] = \ 1 if mask[i, j].sum() > dr * dtheta * 3 / 4 else 0
                    code[code >= 0] = 1
                    code[code < 0] = 0
        return code, code_mask

```

Листинг Б.4 – Код модуля «workingWithDatabase.py»

```

import sqlite3 as sq
import pickle
import numpy as np

class IrisDatabase:
    def __init__(self, db_name='iris_database.db'):
        self.conn = sq.connect(db_name)
        self.cursor = self.conn.cursor()
        self.cursor.execute('''
        CREATE TABLE IF NOT EXISTS iris_codes(
            user_id INTEGER PRIMARY KEY,
            iris_code BLOB,
            iris_mask BLOB
        )
        ''')

    def save_to_database(self, code, mask):
        self.cursor.execute('DELETE FROM iris_codes WHERE user_id>0')
        query = 'INSERT INTO iris_codes (iris_code, iris_mask) VALUES
        (?,?)'
        db_code = pickle.dumps(code)
        db_mask = pickle.dumps(mask)
        self.cursor.execute(query, (db_code, db_mask))

```

```

        self.conn.commit()

    def get_from_database(self):
        self.cursor.execute('SELECT iris_code, iris_mask FROM
iris_codes')
        row = self.cursor.fetchone()
        if row:
            codes = pickle.loads(row[0])
            masks = pickle.loads(row[1])
            return codes, masks
        else:
            return None, None

    def close(self):
        self.cursor.close()
        self.conn.close()

    def lenght(self):
        count=self.cursor.execute('SELECT COUNT(*) FROM iris_codes;')
        return count

```