

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Кафедра

вычислительной техники

Вычислительная техника

Направление подготовки
Магистерская программа

09.04.01 «Информатика и вычислительная техника»
«Вычислительные машины, комплексы, системы и сети»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА МАГИСТРА

на тему

РАЗРАБОТКА БЕЗОПАСНОГО БИОМЕТРИЧЕСКОГО ДОСТУПА К
КИБЕРФИЗИЧЕСКИМ СИСТЕМАМ

Студент

Сергина Ирина
Геннадьевна

(подпись, дата)

(ф.и.о. полностью)

Научный руководитель

Д.В. Пашенко

(подпись, дата)

(фамилия, инициалы)

Нормоконтролёр

А.В. Кучин

(подпись, дата)

(фамилия, инициалы)

Рецензент

Коч. Оксана А.О. ИИИ. Зубин В.И. 9.06.18
(должность, место работы) (подпись, дата)

Пашенко Д.В.
(фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от 25.05.2018 г. №38-09-12)

Заведующий кафедрой

Д.В. Пашенко

(подпись, дата)

(фамилия, инициалы)

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

Секретарь ГЭК

Е.И. Гурин

(подпись, дата)

(фамилия, инициалы)

Пенза, 2018

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

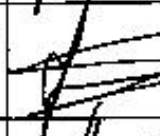
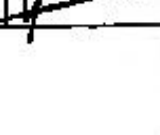
Вычислительная техника

«Утверждаю»
Заведующий кафедрой
Д.В. Пащенко
«24» мая 2018 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
МАГИСТРА

1. Студент гр. 16ВВМ1 факультета ВТ Сергина Ирина Геннадьевна.
2. Тема работы: Разработка безопасного биометрического доступа к киберфизическим системам. Тема утверждена приказом ПГУ № 979 а/о от "3" октября 2016 г.
3. Руководитель работы: Пащенко Д.В.
4. Задание на работу (назначение разработки, исходные данные и т.п.): Обзор систем на основе анализа отпечатка пальца, выбор методов и алгоритмов реализации анализа отпечатка пальца, проектирования и реализации безопасного биометрического доступа, произвести обзор существующих систем идентификации пользователей по биометрической идентификации и классификаторов изображений отпечатков пальца, реализовать один из рассмотренных алгоритмов.
5. Перечень подлежащих разработке вопросов: Аналитический обзор существующих классификаторов и алгоритмов биометрического доступа с помощью отпечатка пальца. Выбор средств программной среды для реализации алгоритма биометрического доступа. Описание разработанного безопасного биометрического доступа. Анализ полученных результатов.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Аналитический обзор существующих классификаторов и алгоритмов биометрического доступа с помощью отпечатка пальца	25%	21.05.2018 – 25.05.2018	
2	Проектирование безопасного биометрического доступа к киберфизическим системам	20%	26.05.2018 – 28.05.2018	
3	Программная реализация системы безопасного доступа к киберфизическим системам	20%	29.05.2018 – 31.05.2018	
4	Анализ полученных результатов.	15%	01.06.2018 – 02.06.2018	
5	Написание пояснительной записки	20%	03.06.2018 – 05.06.2018	

Дата выдачи задания "21" 05 2018 г.

Руководитель магистерской работы
(подпись, дата)

 21.05.18

Задание к исполнению принял студент
(подпись, дата)

 21.05.2018

Работу к защите допустить
Декан факультета
(подпись, дата)

 Л.Р. Рязанова
13.06.2018

РЕФЕРАТ

Пояснительная записка содержит 66 листов, 21 рисунок, 20 формул, 22 использованных источника, 1 приложение.

РАЗРАБОТКА БЕЗОПАСНОГО БИОМЕТРИЧЕСКОГО ДОСТУПА К КИБЕРФИЗИЧЕСКИМ СИСТЕМАМ

ОТПЕЧАТОК ПАЛЬЦА, БИОМЕТРИЧЕСКИЙ ДОСТУП, КИБЕРФИЗИЧЕСКИЕ СИСТЕМЫ, РУТНОН, ХЕШИРОВАНИЕ

Объектом исследования является совокупность параметров и индивидуальных характеристик отпечатка пальца каждого из пользователей.

Предметом исследования является алгоритмы выделения параметров отпечатков пальца, так же методы безопасной идентификации пользователя согласно полученному образу отпечатка пальца.

Цель данной работы состояла в разработке прототипа системы, выполняющей безопасный доступ к киберфизическим системам.

В результате проделанной работы был разработан прототип система, реализующая безопасный биометрический доступ к киберфизическим системам.

Научная новизна состоит в разработке и апробации улучшенных алгоритмов идентификации пользователей на основе образцов отпечатка пальца, а так же оценка точности предложенных алгоритмов по сравнению с существующими методами.

					ВКР 090401.03.18 61 01			
Изм.	Лист	№ докум.	Подпись	Дата	Разработка безопасного биометрического а доступа к киберфизическим системам Пояснительная записка	Лит.	Лист	Листов
Разраб.		Сергина И.Г.		09.06.18				
Провер.		Пашенко Д.В.		09.06.18			3	61
Реценз.								
Н. Контр.		Кучин А.В.		09.06.18		ПГУ, гр.16ВВм1		
Удобр.		Пашенко Д.В.						

Содержание

Введение	6
Глава 1 Отпечаток пальца как средство биометрической идентификации	8
1.1 Понятие киберфизических систем и биометрической идентификации	8
1.2 Параметры биометрического доступа к киберфизическим системам	11
1.3 Особенности система на основе анализа отпечатка пальца	13
1.4 Обзор существующих систем.....	14
1.4.1 Биометрический замок L-9000	14
1.4.2 Сканер отпечатка пальца HTC One Max как пароль	15
1.5 Выводы по главе 1	16
Глава 2 Проектирование безопасного биометрического доступа к киберфизическим системам.....	18
2.1 Обзор существующих классификаторов изображений отпечатков пальцев	18
2.1.1 Метод ближайших соседей	18
2.1.2 Методы на основе нейронных сетей	18
2.1.3 Двухэтапный классификатор	24
2.1.4. Классификатор дерево решений	25
2.2 Обзор существующих алгоритмов безопасного биометрического доступа с помощью отпечатка пальца	26
2.2.1 Корреляционное сравнение	26
2.2.2 Сравнение по узору	27
2.2.3 Сравнение по особым точкам	28
2.2.4 Сопоставление по шаблону	30
2.3 Разработка безопасного метода обработки данных	31
2.3.1 Предварительная обработка данных	32
2.3.2 Алгоритмы идентификации по отпечаткам пальцев	37
2.3.3 Хеширование данных.....	45
2.4 Выводы по главе 2.....	48
Глава 3 Программная реализация системы безопасного биометрического доступа к киберфизическим системам.....	49

3.1 Структура системы безопасного биометрического доступа	49
3.2 Выбор средств программной реализации системы	49
3.3 Описание разработанного метода безопасного биометрического доступа к киберфизическим системам.....	51
3.4 Направления дальнейшего развития системы	55
Заключение	56
Список использованных источников	58
Приложение А. Листинг программы.....	61

Введение

На сегодняшний день актуальной задачей является защита конфиденциальной информации от несанкционированного доступа. Для достижения данной задачи используется комплекс средств, основанных на различных компонентах:

- атрибуты – к ним относятся пропуск, ключ, документы;
- знания – пароли, коды или ответ на вопрос;
- биометрическая информация – совокупность индивидуальных и уникальных физиологических параметров пользователя.

Самые популярные методы биометрической идентификации – методы, которые используют статические характеристики человека. Например, идентификация по радужной оболочке глаза, папиллярному рисунку на пальцах, геометрии лица. Кроме статических характеристик используют и динамические характеристики человека, например, идентификация по голосу, походке, сердечному ритму, динамике рукописного и клавиатурному почерку.

На рисунке 1 представлено примерное значение деления рынка, связанного с биометрической идентификацией.

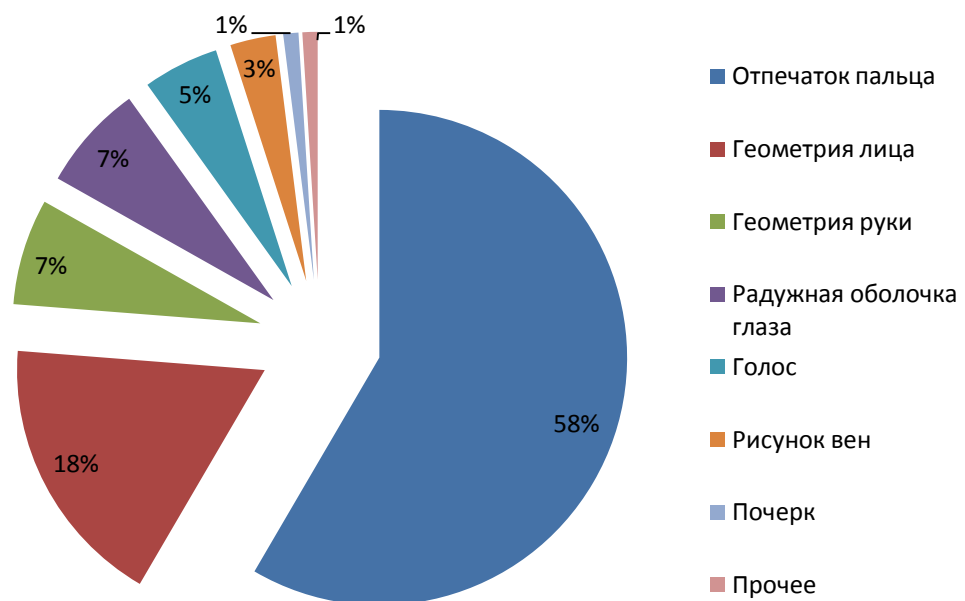


Рисунок 1—примерное значение деления рынка, связанного с биометрической идентификацией

Но системы, обеспечивающие безопасность на основании данных уязвимы и могут легко быть подделаны, украдены, и др. Поэтому методы биометрической идентификации, которые могут определять личность человека по его физиологическим признакам путем распознавания образов, вызывают все больший интерес. Такие способы аутентификации удобны для пользователей тем, что не требуется запоминать многочисленные сложные пароли, так как носители информации находятся при них.

Аутентификация с использованием отпечатка пальца имеет множество преимуществ, например:

- высокую достоверность;
- низкая стоимость устройств, сканирующих отпечаток пальца;
- относительно простая процедура считывания отпечатка.

Существует и недостаток данной аутентификации – присутствует вероятность подделки изображения отпечатка пальца.

Однако на сегодняшний день существуют биометрические устройства, которые с более интеллектуальными сложными устройствами считывания следующего поколения, позволяющие осуществить высоконадежную аутентификацию личности пользователя[1].

В современном мире существуют технические средства идентификации с использованием биометрической идентификации по отпечатку пальца для киберфизических систем (например, мобильный телефон). Но существует вероятность утрата биометрических образов без использования средств защиты, поэтому актуальным является задача создания системы безопасной идентификации с использованием существующих технических решений

Исходя из этого, в рамках данной работы предлагается проектирование прототипа системы безопасного биометрического доступа к киберфизическим системам, а так же разработка комплексного метода идентификации пользователя по характеристикам отпечатка пальца.

Глава 1 Отпечаток пальца как средство биометрической идентификации

1.1 Понятие киберфизических систем и биометрической идентификации

Киберфизические системы (КС) — это системы, состоящие из различных вычислительных ресурсов, позволяющих интеграцию в физические процессы. Компьютеры осуществляют мониторинг и управление физическими процессами с использованием обратной связи, где происходящее в физических системах оказывает влияние на вычислительный процесс и наоборот[2].

Понятие КС было введено для обозначения комплексов, состоящих из природных объектов, искусственных подсистем и контроллеров. Именно с такими системами связана проблема модернизации производства и экономики. Область применения КС достаточно широка, но наиболее востребованными являются в области производства, строительства, транспорта и энергосбережения.

Главным преимуществом КС является преобразование в ценную информацию тех данных, которые для предыдущих поколений были бесполезны и для человека, и для техники.

В КС вычислительные элементы взаимодействуют с датчиками, которые обеспечивают мониторинг киберфизических показателей, и с исполнительными элементами, которые вносят изменения в киберфизическую среду. Очень часто КС ориентированы управление окружающей средой каким-либо образом. Киберфизические системы объединяют информацию от интеллектуальных датчиков, распределенных в физической среде, для лучшего понимания среды и выполнения более точных действий.

В физическом контексте исполнительные элементы на основе получаемых данных вносят изменения в среду обитания пользователей. В виртуальном контексте КС применяются для сбора данных о виртуальных действиях пользователей, например, использование социальных сетей и сайтов

электронной коммерции. КС определенным образом реагируют на такие данные, прогнозируя действия или потребности пользователей в целом. С помощью программных продуктов IBM можно проанализировать данные и события, поступающие от датчиков в реальном времени, и встраивать их в интеллектуальные решения. Архитектура КС представлена на рисунке 1.

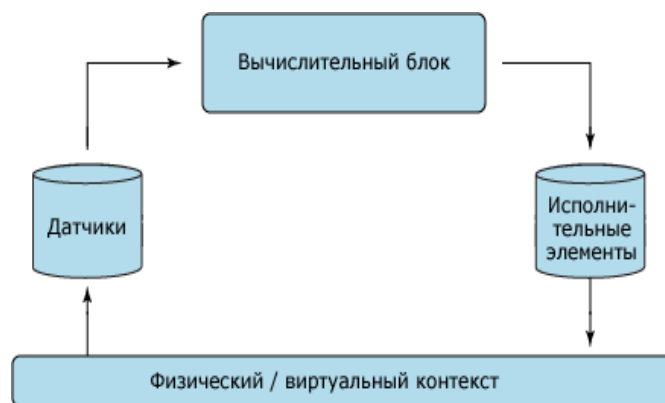


Рисунок 1 – Архитектура КС

Биометрические системы распознают людей с помощью специальных датчиков на основе анатомических особенностей (отпечаток пальцев, образ лица, радужная оболочка, голос) или поведенческих черт (подписи, походки). Поскольку эти черты физически связаны с пользователем, биометрическая идентификация надежна в роли механизма, следящего, чтобы только уполномоченные, могли получить доступ. Биометрические системы дают возможность определить, когда пользователь использует несколько удостоверений (например, паспортами) на разные имена. Таким образом, биометрические системы обеспечивают высокий уровень защищенности.

Исходя из этого, идентификация пользователя по отпечатку пальца обладает следующими преимуществами:

- способность распознавать и проверять индивидуальные данные человека.
- воссоздать эти данные практически невозможно, поэтому вероятность кражи логина или пароля исключена. Таким образом, биометрическая система более надежна, чем те, в которых используется пароль.

- очевидное преимущество системы для работодателей – это отсутствие вероятности попадания посторонних на территорию предприятия, что было возможно при использовании системы доступа на основе электронных карт.

- пользователям не нужно носить с собой карты или запоминать пароли, ведь при использовании биометрической системы вся необходимая для доступа информация всегда находится при себе.

- несмотря на то, что процесс записи всех данных нового сотрудника довольно трудоемкий, в данном случае отсутствуют постоянные расходы, связанные с выдачей новых карт.

Однако существует ряд недостатков, которые не позволяют использовать данный метод как единственный способ идентификации для систем с повышенными требованиями по безопасности и защиты от НСД:

- биометрическая система может плохо работать на фабриках и заводах, так как сотрудники работают с различными устройствами, после чего их руки часто испачканы маслом, смазочным веществом и т.д. В таком случае система может не справиться со сканированием отпечатков пальцев.

- биометрические устройства с высоким процентом ошибочного разрешения доступа также могут представлять серьезную проблему, предоставляя доступ посторонним лицам. Поэтому лучше, чтобы система имела высокий процент ошибочного отказа в доступе, чем наоборот.

- несмотря на возможность сэкономить на расходах на карты для сотрудников, используя биометрическую систему, все же стоит учитывать, что процесс регистрации шаблонов новых сотрудников в считыватель достаточно утомительный.

1.2 Параметры биометрического доступа к киберфизическим системам

Исходя из предложенного определения исходными данными для проведения анализа являются параметры отпечаток пальца пользователя в процессе авторизации данного пользователя.

Биометрическая система на этапе регистрации данных записывает образец биометрического образа пользователя с помощью специального датчика. Затем из биометрического образца извлекаются индивидуальные черты. Система сохраняет извлеченные черты в качестве эталонного шаблона в базе данных наряду с другими идентификаторами, такими как имя или идентификационный номер. Для аутентификации пользователь предъявляет датчику еще один биометрический образец, черты которого представляют собой запрос. Система сравнивает с эталонным шаблоном заявленной личности с помощью алгоритма сравнения. Он возвращает коэффициент соответствия, отражающий степень схожести между эталоном и запросом. Система принимает положительное решение, только если коэффициент соответствия превышает заранее заданный порог.

Биометрическая система подвержена двум видам ошибок, которые представлены на рисунке 2. Первый – система не распознает легитимного пользователя, происходит отказ в обслуживании. Второй вид ошибки – самозванец неверно идентифицируется в качестве авторизованного пользователя, в этом случае говорится о вторжении. Для таких ситуаций существует много возможных причин. Их можно поделить на естественные ограничения и атаки злоумышленников.

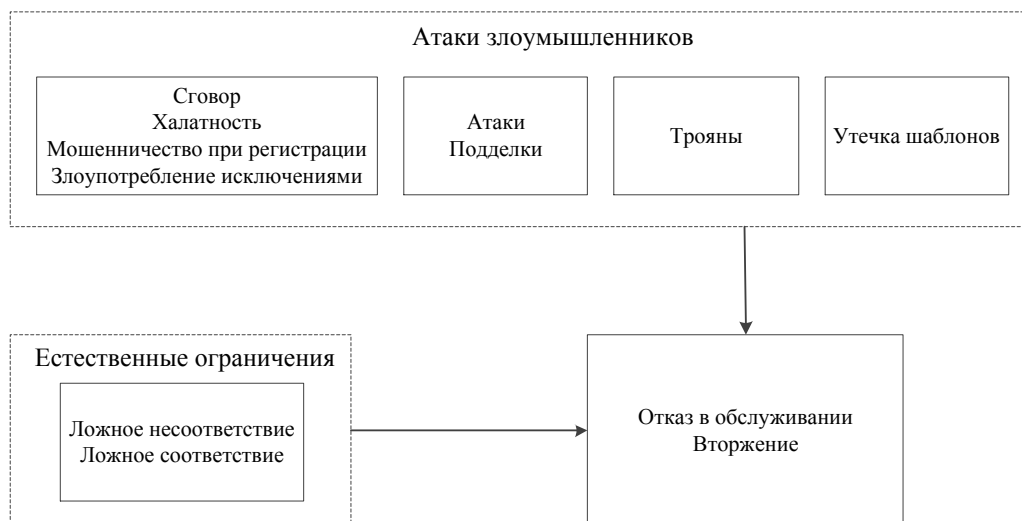


Рисунок 2–Виды ошибок биометрической системы

Если в системах аутентификации по паролю, которые требуют точного соответствия двух строк, то биометрическая система аутентификации полагается на степень схожести двух биометрических образцов. Но из-за того что индивидуальные биометрические образцы, полученные в ходе регистрации и аутентификации, редко идентичны, то, биометрическая система может осуществлять ошибки аутентификации двух видов:

1. Ложное несоответствие.

Два образца от одного и того же пользователя имеют низкую схожесть и система не может их сопоставить. Данный вид ошибки происходит, когда два образца от разных пользователей имеют высокое подобие, поэтому система некорректно объявляет их совпадающими. Это ведет к отказу в обслуживании легитимного пользователя.

2. Ложное соответствие

Такая ошибка обратна ложному несоответствию и может привести к вторжению самозванца.

Биометрическая система также может дать сбой в результате злоумышленных манипуляций. Также злоумышленники могут вызвать сбой в биометрической системе посредством прямых атак на пользовательский интерфейс (датчик), модули экстракции черт или сопоставления, либо на соединения между модулями или базу шаблонов. Атака подделки состоит в

предоставлении поддельного биометрического образа, полученного искусственно: пластилиновый палец, снимок или маска лица, реальный отрезанный палец легитимного пользователя.

Чтобы избежать подделку предлагается хэшировать изображение и хранить не само изображение, а хранить только хэш.

1.3 Особенности системы на основе анализа отпечатка пальца

Все системы биометрической аутентификации оцениваются по следующим характеристикам:

- FRR (False Reject Rate) или ошибка первого рода – вероятность ошибочных отказов авторизованному пользователю (ошибочный отказ «своему»);
- FAR (False Accept Rate) или ошибка второго рода – это вероятность допуска незарегистрированного пользователя (ошибочный пропуск «чужого»);
- EER (Equal Error Rates) – равная вероятность (норма) ошибок первого и второго рода.

Для проверки характеристики FRR системе последовательно предъявляются биометрические образы пользователей, успешно прошедших регистрацию. Затем подсчитывается отношение числа отрицательных результатов проверки (отказ системы в допуске) к общему числу попыток. Итоговое отношение показывает оценку вероятности ошибки FRR. Оценка считается достоверной, если количество образцов превышает $1/\text{FRR}$.

Для оценки характеристики FAR системе последовательно предъявляются биометрические характеристики пользователей, не прошедших регистрацию. Далее подсчитывается отношение количества положительных результатов идентификации (положительное аутентификационное решение) к общему числу попыток. Указанное отношение дает оценку вероятности ошибки FAR, которая считается достоверной, если количество образцов превышает $1/\text{FAR}$.

Таким образом, среди задач разработки безопасного биометрического доступа следует уделить внимание также подбору алгоритмов анализа образцов и методам идентификации, обеспечивающих оптимальное соотношение вероятности ошибок FRR и FAR, исходя из требований безопасности защищаемой системы.

1.4 Обзор существующих систем

Поскольку в настоящее время значительно возрос интерес к построению систем авторизации и безопасности на основе биометрических данных, в том числе и с использованием анализа отпечатка пальца, существует ряд проектов, осуществляющих подобный анализ. Рассмотрим наиболее распространенные из них.

1.4.1 Биометрический замок L-9000

Биометрический замок L-9000 содержит встроенный сканер отпечатков пальцев, а также цифровую клавиатуру для ввода пароля. В память устройства можно внести до 500 отпечатков пользователей, которые могут его открывать. Также замок можно открыть вводом PIN-кода на цифровой клавиатуре. В случае, если источники питания разрядились, замок можно открыть при помощи механического ключа или подсоединить внешний источник питания. Замок имеет массивный металлический корпус и внешне ничем не отличается от обычного механического замка. Сканер отпечатков пальцев и цифровая клавиатура скрыты под слайдером, тем самым оптический сенсор дополнительно защищен от попадания пыли и влаги.

Плюсы биометрических замков – они эффективно распознают тех, кто имеет доступ в дом, так как отпечаток пальца каждого человека уникален. В то же время, такие устройства не являются абсолютной гарантией, что никто чужой не войдет в дом без разрешения. Ведь сама их структура подразумевает несколько обходных путей – магнитную карточку, код, ключ. К тому же, такие

замки не подходят для установки на входных дверях дома, потому что сильно зависят от температуры, влажности и других факторов.



Рисунок 3– Пример биометрического замка

Минусы биометрических замков - работа таких механизмов осуществляется благодаря батарейкам и всегда есть риск, что владелец пропустит момент их замены. В таком случае дверь окажется заблокирована и обойтись без специалистов хозяину будет невозможно[4].

1.4.2 Сканер отпечатка пальца HTC One Max как пароль

Считалось, что дактилоскопические сенсоры в мобильных устройствах должны повысить безопасность персональных данных, но как выяснилось что не везде это так.



Рисунок 4– Дактилоскопический датчик HTCOneмах

Дактилоскопический датчик в HTC One max, изображенный на рисунке 4, находится на тыльной стороне корпуса, под объективом камеры. Существует возможность привязки отпечатков пальцев к разным приложениям. Но большим и очевидным минусом данной модели является то, что данная модель не защищает отпечатки пальцев, а хранит их в виде обычных картинок в памяти устройства, к которым свободно могут получить доступ злоумышленники или приложения [5].

1.5 Выводы по главе 1

На сегодняшний день повышенный интерес к методу идентификации на основе биометрической информации обусловлен в первую очередь простотой эксплуатации и возможностью обеспечения повышенного уровня безопасности и защиты данных от несанкционированного доступа. Современное аппаратное обеспечение, а также современные методы анализа данных позволяют быстро и качественно производить идентификацию пользователя по биометрическим данным.

Подобного рода системы, как правило, устанавливаются на объектах с повышенными требованиями по безопасности для защиты конфиденциальной информации. В силу дороговизны оборудования или сложности настройки и эксплуатации данные технологии недоступны широкому кругу пользователей персональных компьютеров. Одним из вариантов решения данной задачи является методы биометрической идентификации на основании анализа событий со стандартных для персонального компьютера устройств ввода, таких как клавиатура или мышь. В рамках данной работы к рассмотрению предложена безопасная идентификация по отпечатку пальцу, поскольку это очень актуальный вопрос, который затрагивает все сферы современного общества.

Анализ существующих решений показал отсутствие систем, получивших широкое распространение, для безопасной авторизации пользователей по отпечатку пальца для персонального некоммерческого использования.

Исходя из вышесказанного, было принято решение о разработке безопасного биометрического доступа, предназначенной для защиты личных данных пользователей персонального устройства, которая сочетала бы в себе простоту эксплуатации, высокое быстродействие и обеспечивала бы необходимый уровень защиты данных.

Для достижения этих целей предлагается создание системы биометрической аутентификации с шифрованием информации, а также разработка и апробация различных алгоритмов анализа отпечатка пальца пользователя и принятия аутентификационного решения. Разработанная система послужит основанием для проектирования системы получения управления удаленным доступом к автоматизированным комплексам армейского назначения.

Глава 2 Проектирование безопасного биометрического доступа к киберфизическим системам

2.1 Обзор существующих классификаторов изображений отпечатков пальцев

Такой метод как классификация отпечатков пальцев используется для отнесения отпечатка пальца на основании его признаков к одному из нескольких предварительно заданных классов, которые в последствии обеспечат механизм индексации.

2.1.1 Метод ближайших соседей

Основное правило «К» ближайших соседей состоит в том, что вначале находятся «К» ближайших соседей для тестового образца в пространстве признаков. После этого тестовый образец относится к классу, который наиболее часто представлен среди «К» ближайших соседей. Два верхних класса, которые были найдены с помощью классификатора «К» ближайших соседей, должны соответствовать классам, которые имеют самое высокое и второе по величине количество среди «К» ближайших соседей. Обычно рассматриваются 10 ближайших соседей ($K=10$). Точность классификации не всегда увеличивается с увеличением K ; здесь возникает проблема классификации, связанная с определением оптимальной величины K для объема проверочной выборки конечного размера [6].

2.1.2 Методы на основе нейронных сетей

Перспективными методами идентификации являются также методы на основе нейронных сетей. Для решения данной задачи можно предложить два типа сетей: многослойные сети прямого распространения и рекуррентные нейронные сети.

Сети с прямой связью являются универсальным средством аппроксимации функций, что позволяет их использовать в решении задач

классификации. Как правило, нейронные сети оказываются наиболее эффективным способом классификации, потому что генерируют фактически большое число регрессионных моделей (которые используются в решении задач классификации статистическими методами). Данный вариант применим, когда в системе имеется большое количество биометрических образов пользователей, тогда для обучения классификатора в качестве образов «чужих» используются все известные образы, кроме образа текущего пользователя. Такой метод позволит в высокой точностью отличить одного пользователя от другого, однако поведение такой сети для неизвестного образа может быть непредсказуемо. Несмотря на данный недостаток, весомое преимущество данного метода позволяет использовать данный класс нейронных сетей, совместно с другими методами, позволяющими более точно идентифицировать «чужой» образ.

Другой тип нейронных сетей относится к рекуррентным сетям – сетям с обратной связью, которые успешно применяются для распознавания заданного образа, даже при наличии определенной доли шумов. В данном классе сетей можно выделить нейронные сети Хопфилда и Хэминга.

В сети нейронной сети Хопфилда известен набор из K эталонных образов, которые должны храниться в памяти сети и, при необходимости, правильно распознаваться ею. Каждый образ – это вектор, состоящий из M идеальных входных сигналов. После обучения полученная сеть Хопфилда должна уметь из поданного на ее вход зашумленного вектора выделить один из известных эталонных образов или дать заключение, что входные данные не соответствуют ни одному из них.

Нейронная сеть состоит из N искусственных нейронов. Каждый нейрон системы может принимать одно из двух состояний, аксон каждого нейрона связан с дендритами остальных нейронов, образуя обратную связь. Каждый нейрон может находиться в одном из 2-х состояний: -1 или 1 (в некоторых задачах 0 или 1).

Весовые коэффициенты связей между нейронами образуют матрицу взаимодействий $\bar{W}$, которая является симметричной, а диагональные элементы матрицы полагаются равными 0, что исключает эффект передачи сигнала нейрона на самого себя и является необходимым, но не достаточным условием, устойчивости в процессе работы сети.

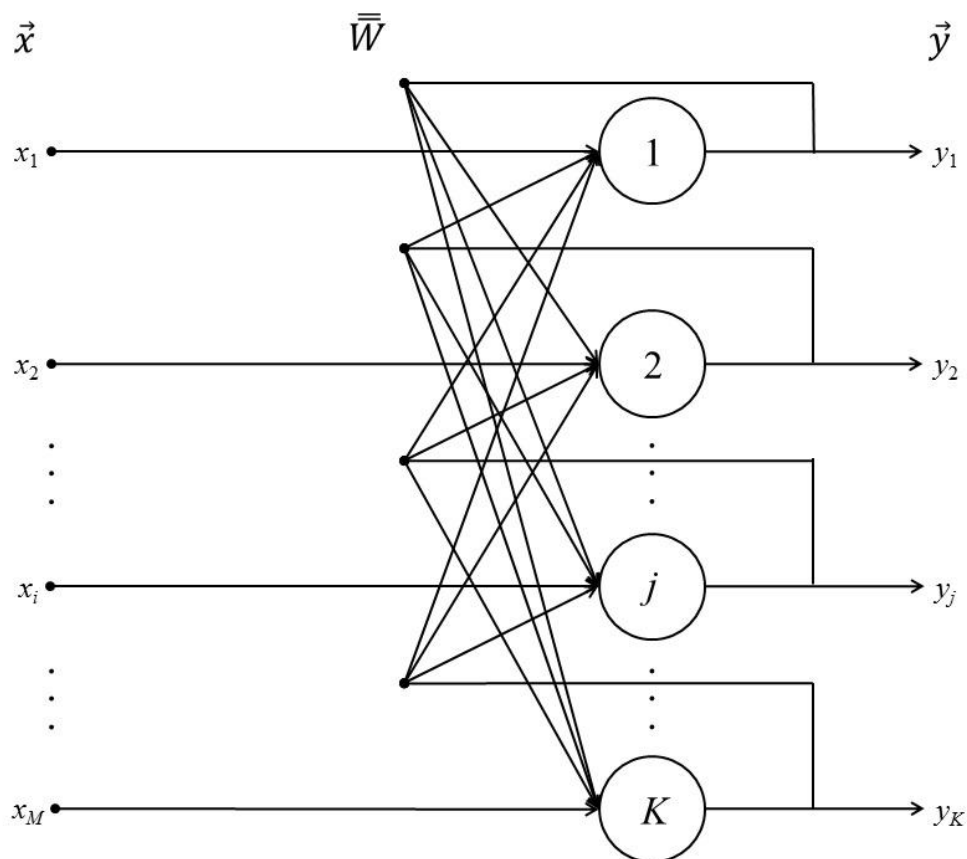


Рисунок 5 – Структура сети Хопфилда

Процесс распознавания образа при помощи обученной сети Хопфилда заключается в итеративном перерасчете элементов матрицы весовых коэффициентов, до тех пор, пока не стабилизируется состояние элементов выходного вектора, или пока выходные вектора не начнут чередоваться. После чего состояние аксонов либо соответствует одному из известных образов, если состояние выходных сигналов стабилизировалось, либо не соответствует ни одному из эталонов, из чего следует, сеть не смогла распознать сигнал[7].

Для решения задачи распознавания эталонных образов помимо сетей Хопфилда также часто применяется другой вид рекуррентных сетей – Сеть Хэминга. В отличие от сети Хопфилда, сеть Хэминга выдает не сам

идентифицированный образ, а номер обученного образца. Кроме того, как правило, этот тип сетей характеризуется меньшими вычислительными затратами.

Структурно нейронная сеть Хэмминга включает два слоя, количество нейронов в которых K равно количеству классов ($K = N$). Число входов M соответствует числу бинарных признаков, по которым различаются образы. Значения входных переменных принадлежат множеству $\{-1; 1\}$. Выходные значения подаются по обратным связям на входы нейронов второго слоя, в том числе на свой собственный нейрон. Обобщенная структура сети Хэмминга представлена на рисунке 6.

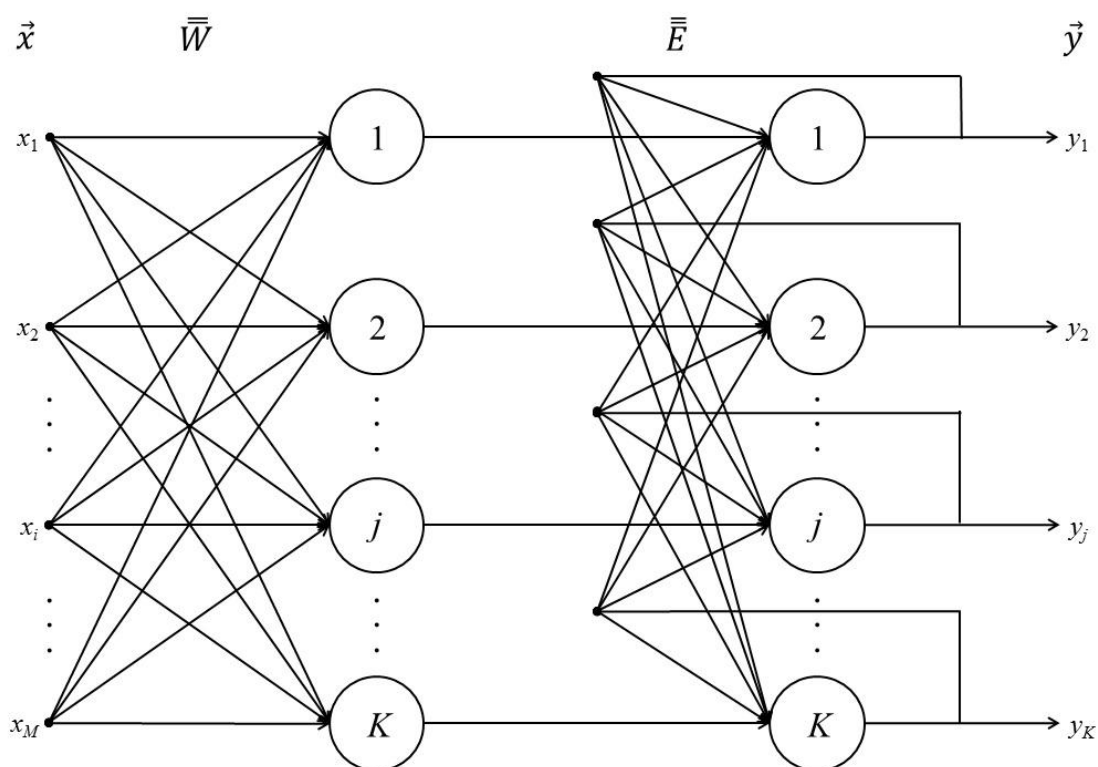


Рисунок 6 – Структура нейронной сети Хэмминга

В основе работы сети лежат процедуры, направленные на выбор в качестве решения задачи классификации одного из эталонных образов, наиболее близкого к поданному на вход сети зашумленному входному образу, и отнесение данного образа к соответствующему классу. Для оценки меры близости образа к одному из классов используется критерий, использующий

расстояние Хэмминга — количество различающихся соответствующих элементов входного и эталонного векторов.

На вход подаётся классифицируемый вектор, и состояние нейронов первого слоя пересчитывается. Выход нейронов первого слоя получается путём применения функции активации к состоянию, и становится начальным значением соответствующих нейронов второго слоя. Далее, состояния нейронов второго слоя получаются из их предыдущего состояния, исходя из матрицы весовых коэффициентов второго слоя, и процедура повторяется итерационно до стабилизации вектора состояния второго слоя— пока норма разницы векторов двух последовательных итераций не станет меньше порогового значения. В случае если в итоге один вектор положительный, а остальные отрицательные, то он указывает на подходящий образец. В случае же, если несколько векторов положительны, и при этом, не один из них не превышает предельного значения, то это значит, что нейронная сеть не может отнести входящий вектор ни к одному из классов, однако положительные выходы указывают на наиболее схожие эталоны[8].

Применения рекуррентных нейронных для идентификации образца отпечатка пальца, характеристики которого представлены в формате временных отметок, возникает необходимость кодирования исходных параметров в двоичном виде.

Представление целочисленных данных в двоичном виде, в котором они размещаются в памяти, является неприемлемым, поскольку могут давать некорректную оценку меры близости по Хеммингу.

Для реализации задачи распознавания входного биометрического образа при помощи рассмотренных рекуррентных сетей входной вектор признаков должен быть представлен в виде бинарного вектора фиксированного размера, одинакового для всех входных образов. Для реализации данного условия предлагается разбить максимальный диапазон входных значений на множество

равных интервалов. После чего имеется два варианта представления величины в виде двоичного вектора:

1) кодировать каждую величину как вектор, в котором разряд, соответствующий интервалу, в который попадает значение, равен “1”, тогда как все остальные принимают значение “-1” или “0”;

2) кодировать каждую величину как вектор, в котором элементы, соответствующие интервалам, не превышающим значения параметра, принимают значение “1”. Всем остальных присваивается значение “-1” или “0”.

Данный способ кодирования позволяет задавать произвольную размерность эталонных образов за счет вариации количества и размера интервалов, что позволяет управлять соотношением размера нейронной сети, от которой зависит скорость обработки, и требуемой точности.

Однако для успешного применения данного типа нейронных сетей на практике с целью распознавания отпечатка пальца пользователя необходимо решить еще одну проблему. В процессе обучения системы на вход может подаваться большое количество образцов отпечатка пальца одного пользователя.

Построение рекуррентной нейронной сети для каждого из входных образцов нерационально. Данный вывод был сделан на основании предъявляемого требования к количеству нейронов входного слоя. В частности, для сети Хопфилда требуется, чтобы количество нейронов как минимум в 6-7 раз превосходило количество эталонных образцов, из чего следует геометрический рост количества узлов сети в зависимости от количества образцов. Исходя из этого, необходимо из всего множества обучающих образцов отпечатка пальца выделять или формировать один или несколько эталонных, которые наиболее полно характеризовали бы биометрический образ пользователя.

2.1.3 Двухэтапный классификатор

Двухэтапный классификатор позволяет сначала отнести отпечаток пальца к одному из подклассов, а после уже в этом подклассе производить сравнение. Целью является выполнение задачи простой классификации, используя классификатор «К» - ближайших соседей и затем используется набор двух-классовых классификаторов нейронных сетей для манипуляции с едва заметными различиями.

Первый этап использует классификатор «К» - ближайших соседей, чтобы выбрать два наиболее вероятных класса для текущего входного образца. В большинстве случаев, класс с максимальной частотой попадания в группу «К» ближайших соседей это правильный класс (класс прошел классификацию) и в оставшихся случаях класс со второй по величине частотой это также правильный класс. Классификатор «К» - ближайших соседей приводит к нахождению двух классов с наибольшей частотой попадания в группу. Этот результат используется для точного отнесения отпечатков пальцев к двум из пяти классов. Каждый отпечаток будет иметь вхождение в два из пяти классов БД и поиск соответствия в БД должен быть произведен только в соответствующих двух классах.

Второй этап использует 10 (C_2^5) различных нейронных сетей для 10 различных парных классификаций. Эти нейронные сети имеют 192 входных нейрона, 20-40 скрытых нейронов находятся в одном скрытом слое, и 2 выходных нейрона. Каждая нейронная сеть обучена использовать образцы только из двух соответствующих классов в обучающей выборке. Схема двухэтапного классификатора представлена на рисунке 7.

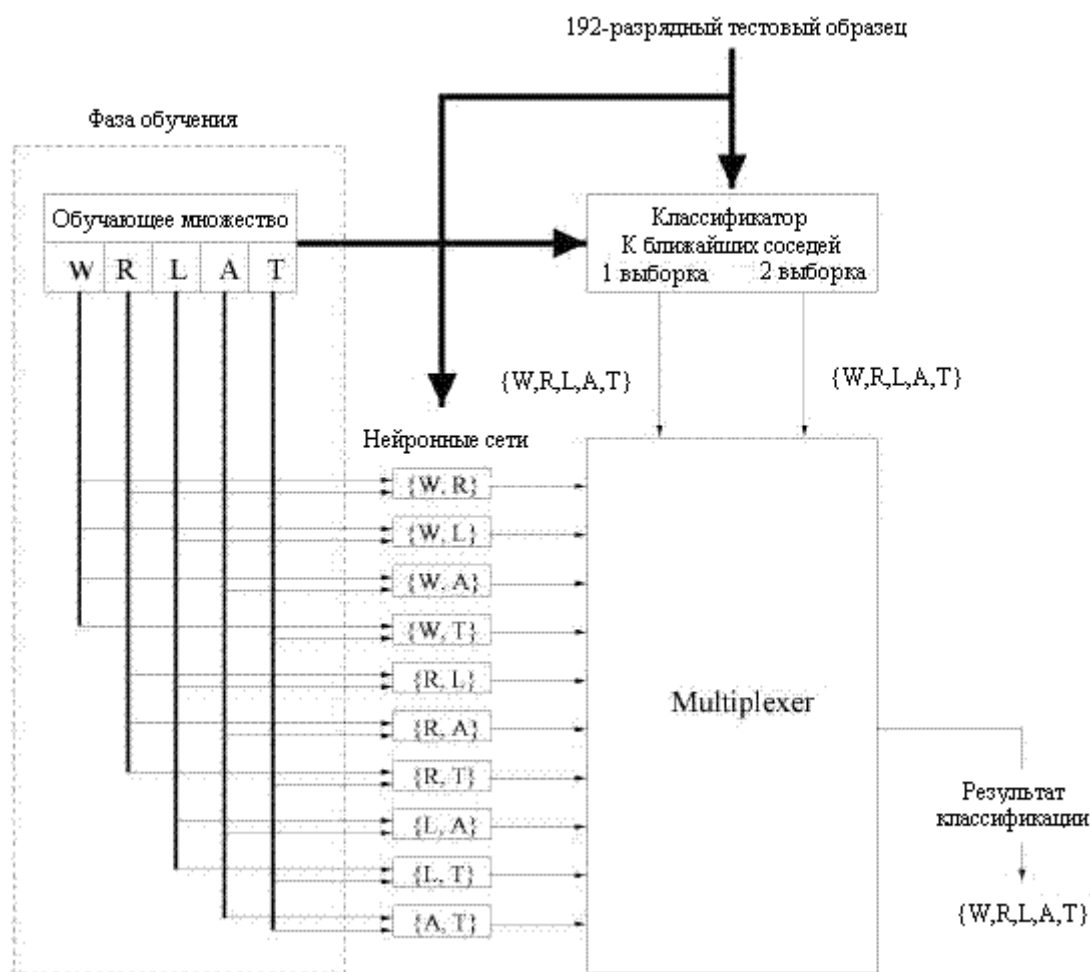


Рисунок 7 – Схема двухэтапного классификатора

Хотя этот классификатор устойчив к шумам и способен корректно классифицировать большинство низкокачественных отпечатков пальцев в БД, он имеет недостатки на некоторых других изображениях отпечатков пальцев, которые имеют очень низкое качество и не содержат информацию о выступах, присутствующую в центральной части отпечатка пальца. На низкокачественных отпечатках пальцев очень трудно правильно обнаружить центральную точку.

2.1.4. Классификатор дерево решений

Процедура, принимаемая здесь для классификации отпечатков пальцев и включенная в начальную фазу выделения признаков, сопровождается построением вопросов, которое заключается в создании информационных вопросов, помогающих при классификации. Этот комплекс вопросов

объединяется по иерархическому принципу, чтобы сформировать деревья решения, которые используются для классификации. Поскольку деревья создаются случайным образом, то деревья, созданные для решения одинаковой проблемы, имеют различное исполнение и затем многочисленные деревья объединяются, чтобы произвести окончательную классификацию [9].

2.2 Обзор существующих алгоритмов безопасного биометрического доступа с помощью отпечатка пальца

В процессе аутентификации решается задача классификации неизвестного пользователя, предъявившего свои биометрические параметры в виде вектора V , на «своего» и «чужого». При этом применяются различные методы классификации:

- корреляционное сравнение;
- сравнение по узору;
- сравнение по особым точкам;
- сопоставление по шаблону;
- методы на основе применения искусственных нейронных сетей (ИНС).

2.2.1 Корреляционное сравнение

Корреляционное сравнение заключается в следующем, два изображения одного отпечатка накладываются один на другой. Затем производится подсчет корреляции между соответствующими пикселями, вычисленной для разных способов выравниваний изображения. Полученный со сканера отпечаток пальца накладывается на каждый эталон из базы данных поочередно, после чего прямо по пикселям изображений осуществляется просчет различий между ними. Но приходится учитывать то, что человек каждый раз прикладывает палец под разными углами и не точно в одно и то же место рабочей области сканера. Из этого следует, что процесс сравнения отпечатка его пальца с эталонами должен

включать в себя множество итераций, на каждой из которых изображение, полученное со сканера, поворачивается под небольшим углом.

Главное достоинство данного метода идентификации заключается в низких требованиях к качеству изображения. Недостаток заключается в большой продолжительности процедуры сравнения эталона с полученным папиллярным узором, что слишком ограничивает сферу применения данного метода. В связи со сложностью и продолжительностью работы рассматриваемого алгоритма, особенно в ходе решения задач идентификации (сравнение "один-к-нескольким") – системы, которые построены с его применением, в настоящее время почти не применяются [10].

2.2.2 Сравнение по узору

В данном алгоритме сравнения используются особенности строения папиллярного узора на поверхности пальцев. Сначала полученное со сканера изображение отпечатка пальца, разбивается на множество мелких ячеек, размер которых зависит от требуемой точности.

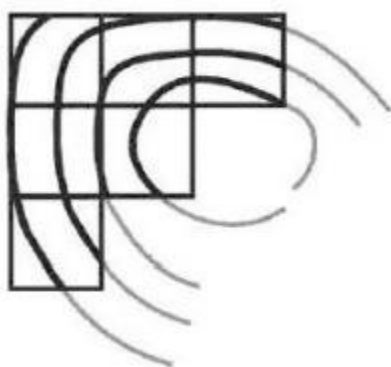


Рисунок 8 – Разбиение папиллярного узора на ячейки

Расположение линий в каждой ячейке описывается параметрами определенной синусоидальной волны, то есть, задается начальный сдвиг фазы (δ), длина волны (λ) и направление ее распространения (θ). Такое волновое представление изображено на рисунке 9.



Рисунок 9 – Волновое представление линий в ячейке

Полученный для сравнения отпечаток выравнивается и приводится к тому же виду, что и шаблон.

Преимуществом рассмотренного алгоритма являются достаточно высокая скорость его работы и низкие требования к качеству получаемого изображения. Но данный метод сравнения по узору пока не получил широкого распространения, ввиду сложности, трудоемкости и требует солидной математической базы [11].

2.2.3 Сравнение по особым точкам

Сравнение по особым точкам заключается в следующем, на полученных со сканера изображениях отпечатков пальцев, имеющих вид двумерной поверхности, находятся особые точки (точки ветвления и конечные точки), использующиеся для идентификации. В процессе идентификации изображения отпечатков пальцев происходит сравнение выделенных на текущих отпечатках и отпечатках-шаблонах точки ветвления и конечные точки. По числам совпадений точек принимается решение об идентификации. Схема алгоритма, реализующего этот метод, представлена на рисунке 10 [12].

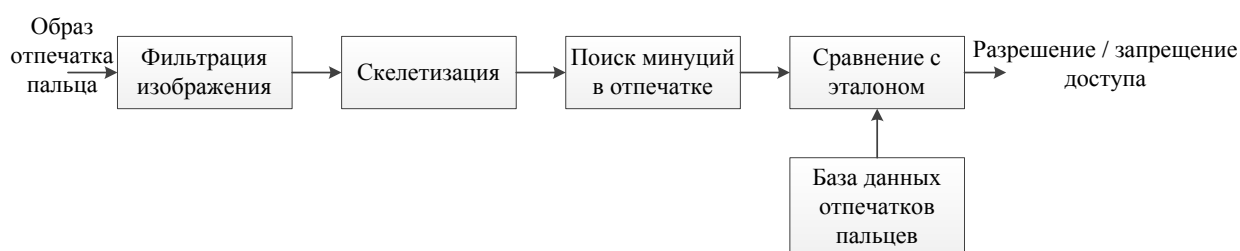


Рисунок 10 – Схема алгоритма сравнения по особым точкам

Часто изображение, полученное с помощью сканера, имеет низкое качество. На качество полученного изображения влияет множество факторов – сила нажима пальца при сканировании, влажность воздуха, чистота пальца и самого сканера. Поэтому для повышения качества исходного изображения используют различные методы фильтрации.

Этап скелетизации подразумевает утончение бинарного изображения образа отпечатка до скелета. Скелет линии – это простая цепь (u,v) , имеющая вершины с u и v -смежностями. Она проходит рядом с геометрическим центром линии. При этом для каждой вершины существует еще две вершины, смежные с ней.

В скелете изображения производится поиск минучий – ключевых точек. В большинстве алгоритмов для распознавания используется только два типа минучий - окончание и разветвление. В скелете изображения они определяются следующим образом: окончанием называется вершина скелета такая, что для неё существует ровно одна смежная с ней вершина. Разветвлением называется такая вершина скелета, что для неё существует три смежные вершины, и, попарно несмежные.

В процессе поиска минучий для каждой ключевой точки определяют ее тип (разветвление или окончание), ориентацию, а также координаты x и y .

На рисунке 11 приведены результаты этапов фильтрации, скелетизации и поиска минучий для образа отпечатка пальца.



Рисунок 11 – Результаты этапов фильтрации

Последний этап идентификации по отпечатку пальца заключается в сравнении принятого на вход системы изображения с эталоном из базы данных. На основе проведения анализа имеющихся данных с использованием решающего правила принимают решение о запрете или разрешении.

Преимуществом метода можно назвать быстроту работы, а недостаток заключается в высоких требованиях к качеству изображения[13].

2.2.4 Сопоставление по шаблону

В данном алгоритме во внимание принимается не только отдельно взятые точки, но и общая характеристика отпечатка пальца, которая может включать определенный процент дополнительных данных, включая толщину полос, их кривизну или плотность.

Алгоритм сопоставления шаблонов во время регистрации определяет наличие различных дополнительных характеристик отпечатка пальца вместо регистрации ключевых точек. Небольшие участки отпечатка пальца и расстояние между ними извлекаются из отпечатка пальца с целью максимально увеличить количество уникальной информации. Наиболее значимы участки вокруг ключевых точек и участки с небольшим радиусом изгиба. Процесс подтверждения начинается с предварительной обработки считанного изображения отпечатка пальца. Зарегистрированное изображение, считанное с

шаблона, сопоставляется с изображением отпечатка, чтобы определить, насколько шаблон совпадает с изображением. Порог, описывающий малейшее допустимое отклонение в последствии используется при определении степени соответствия отпечатка имеющемуся шаблону.

Преимущества данного метода:

- хорошо работает с известными типами сканеров отпечатков пальцев (не зависимо от разрешения сканера);
- не возникают сложности при распознавании пальца с отпечатком худшего качества;
- подходит для работы с недостаточным количеством вычислительных ресурсов (например, смарт-картой);
- любой отпечаток, который можно записать, может быть зарегистрирован, даже если он не имеет или имеет небольшое количество ключевых точек.

Недостатки метода:

- не может использовать базу данных AFIS (однако, может использовать недообработанные изображения)
- не приспособлен для распознавания (для множества поисков в базе данных).

2.3 Разработка безопасного метода обработки данных

Исходя из вышеизложенного, можно сделать вывод, что существует множество алгоритмов идентификации пользователя по образу отпечатка пальца. Но кроме считывания и хранения отпечатка пальца требуется зашифровать отпечаток пальца для безопасной обработки данных. Однако существует общая проблема данного вида идентификации:

Для идентификации человека по отпечаткам пальцев будет использован алгоритм сравнения по особым точкам.

Выбор этого метода определяется его преимуществами — высокой точностью распознавания отпечатков пальцев и быстротой работы. Однако для получения высокой точности распознавания необходимо иметь достаточно высокое качество изображения отпечатков пальцев. Однако это требование не критично, поскольку современные сенсоры позволяют получить достаточно высокое качество изображений отпечатков пальцев.

Система идентификации человека по отпечаткам пальцев с использованием метода особых точек включает модули предварительной обработки изображений отпечатков, выделения особых точек и сопоставления текущих отпечатков с отпечатками-шаблонами из базы данных.

Рассмотрим методы, лежащие в основе этих модулей.

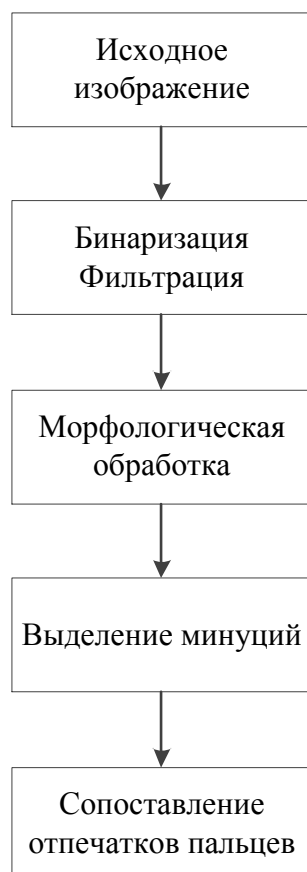


Рисунок 12 – Модули системы идентификации

2.3.1 Предварительная обработка данных

При обработке отпечатков пальцев для выделения особых точек изображение проходит следующие преобразования:

Фильтрация

Применяется к изображениям с целью уменьшения шумов.

Бинаризация

Изображение приводится к черно-белому цвету. Бинаризация начинается после того, как будет улучшено качество изображений и вычислены все направления имеющихся линий [14].

Изображение B определяется как изображение, если каждый пиксель принимает одно из двух возможных значений – нуля единицы. Единица соответствует гребню отпечатка, ноль – впадине:

$$R(i, j) = \begin{cases} 1, & \text{если } G(i, j) > B_0, \\ 0, & \text{иначе} \end{cases}, \quad (1)$$

где B_0 – порог маскирования, а $G(i, j)$ – интенсивность пикселя изображения.

Морфологическая обработка

После бинаризации отфильтрованного изображения выделяют однородные области с проведением дальнейшей морфологической обработки изображения. Она состоит из удаления шума (то есть удаление мелких областей черного и белого цвета).

Изображения со структурной избыточностью, такие как дактилоскопические, состоят из большого количества линий и деталей.

Морфологические операции эффективны для дактилоскопической обработки в случаях, когда эти линии у считанного изображения нечеткие и требуется определить их контур.

Эрозия

Операция эрозии эффективна для удаления малых объектов и шумов, «ужимает» («утончает») двоичное изображение с помощью структурообразующего элемента.

Для множества X и S из пространства Z^2 эрозия X по S ($X \ominus S$) определяется как

$$X \ominus S = \{\xi \mid (S)\xi \subseteq X\}. \quad (2)$$

Другими словами, эрозия – множество всех точек ξ , при сдвиге в которые множество S полностью содержится в X . Здесь S – структурообразующее множество, X – изображение.

При совпадении значения 2 структурообразующего элемента со значением 2 бинарного изображения, выполняется логическое сложение соответствующих пикселей. Все объекты, меньшие, чем структурообразующий элемент, отбрасываются.

Дилатация

Операция дилатации «наращивает» или «утолщает» объекты двоичных изображений. Для множеств X и S из пространства Z^2 дилатация X по S ($X \oplus S$) определяется в соответствии [Гонзалес Р, Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072 с.], как

$$X \oplus S = \{\xi \mid (S)\xi \cap X \neq \emptyset\} \quad (3)$$

Выделение особых точек

Для выделения особых точек или минуций изображение разбивается на блоки 3x3 пикселей и к 8-ми соседним точкам применяется формула

$$cn(p) = \frac{1}{2} \sum_{i=1..8} |val(p_i) - val(p_{i-1})|, \quad (4)$$

где p – значение для каждого пикселя, находящегося рядом с пикселем центрального блока; $val \in \{0,1\}$ – бинарное значение интенсивности пикселя.

После этого подсчитывается число черных (ненулевых) пикселей, находящихся вокруг центра. Пиксель в центре считается минуцией. Если он не нулевой, и соседний не нулевой пиксель один, минуция "окончание" или пикселя три, минуция "бифуркации".

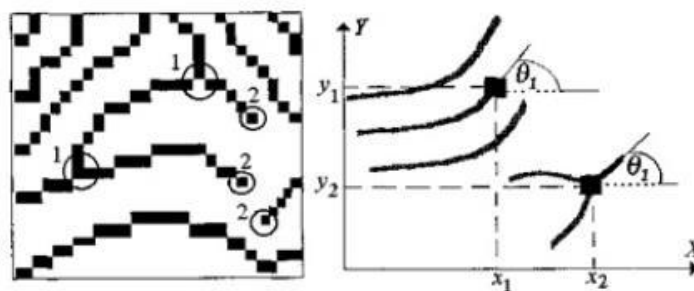


Рисунок 13 – Выделение минуций

Подсчет числа черных пикселей вокруг текущего пиксела основан на анализе 8-связной области смежных пикселей.

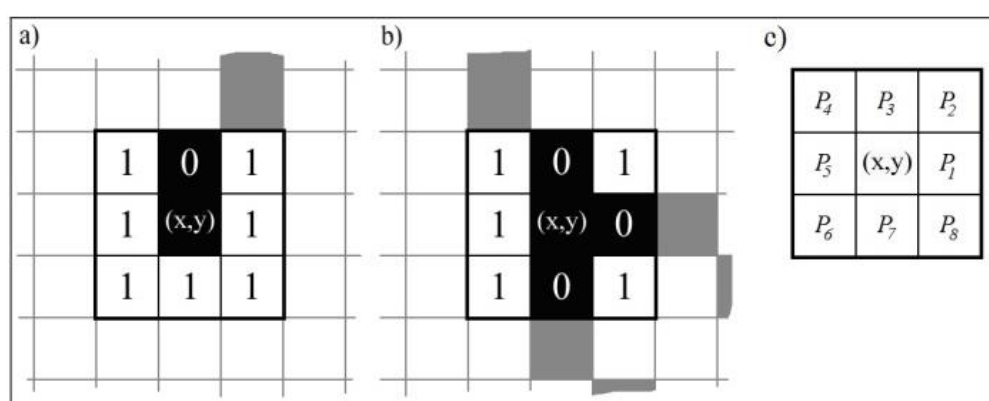


Рисунок 14 – 8-связная область смежных пикселей

Координаты обнаруженных минуций и их углы ориентации записываются в вектор: $W(p)=[(x_1, y_1, t_1), (x_2, y_2, t_2)...(x_p, y_p, t_p)]$ (p – число минуций)[16].

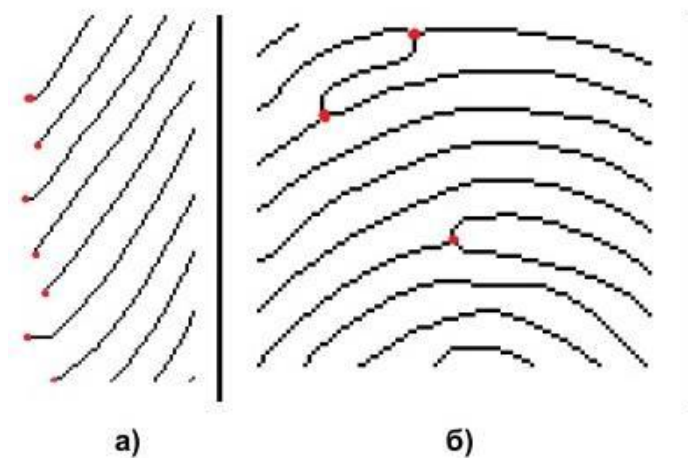


Рисунок 15 – Пример. Выделение минуций: а – окончание, б – бифуркация

Сопоставление отпечатков пальцев

Соответствие отпечатков пальцев вычисляется путем сопоставления двух отпечатков пальцев. Отпечатки одного и того же пальца должны быть максимально соответствующими, и минимально соответствующими, если пальцы разные.

Пусть T и Q есть функция векторов, представляющих контрольные точки, форма шаблона отпечатков пальцев и запроса соответственно. Каждый элемент из этих характеристических векторов является точкой минуций, которые могут быть описаны с помощью различных атрибутов, таких как расположение, ориентация, тип, качество соседства региона и т. д. Наиболее общее представление о минуциях – триплет x, y, θ , где x, y – координаты положения, θ – угол минуции. Пусть количество минуция в T и Q m и n , соответственно, т.е.

$$T = m_1, m_2, \dots, m_m, \quad m_i = x_i, y_i, \theta_i, \quad i = 1 \dots m, \quad (5)$$

$$Q = m'_1, m'_2, \dots, m'_m, \quad m'_j = x'_j, y'_j, \theta'_j, \quad j = 1 \dots m,$$

В минуции m_i в T и m'_j в Q , считаются совпадающими, если следующие условия удовлетворены:

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_0, \quad (6)$$

$$dd(m'_j, m_i) = \min(|\theta'_j - \theta_i|, 360 - |\theta'_j - \theta_i|) \leq \theta_0,$$

где r_0 и θ_0 параметры окна толерантности, которая требуется, чтобы компенсировать ошибки в функции извлечения и искажений из-за пластичности кожи[17].

После того, как найдены совпадающие пары минуций, можно вычислить подобие отпечатков. Показатель подобия должен точно описать, как похожи два отпечатка, с учетом всей информации, полученной от предыдущего шага, такой как количество совпадающих пар минуций и степень схожести пар.

2.3.2 Алгоритмы идентификации по отпечаткам пальцев

В соответствии с методом, выбранным для реализации в работе из описанных выше, общий алгоритм идентификации человека по отпечаткам пальцев представлен схемой на рисунке 16.

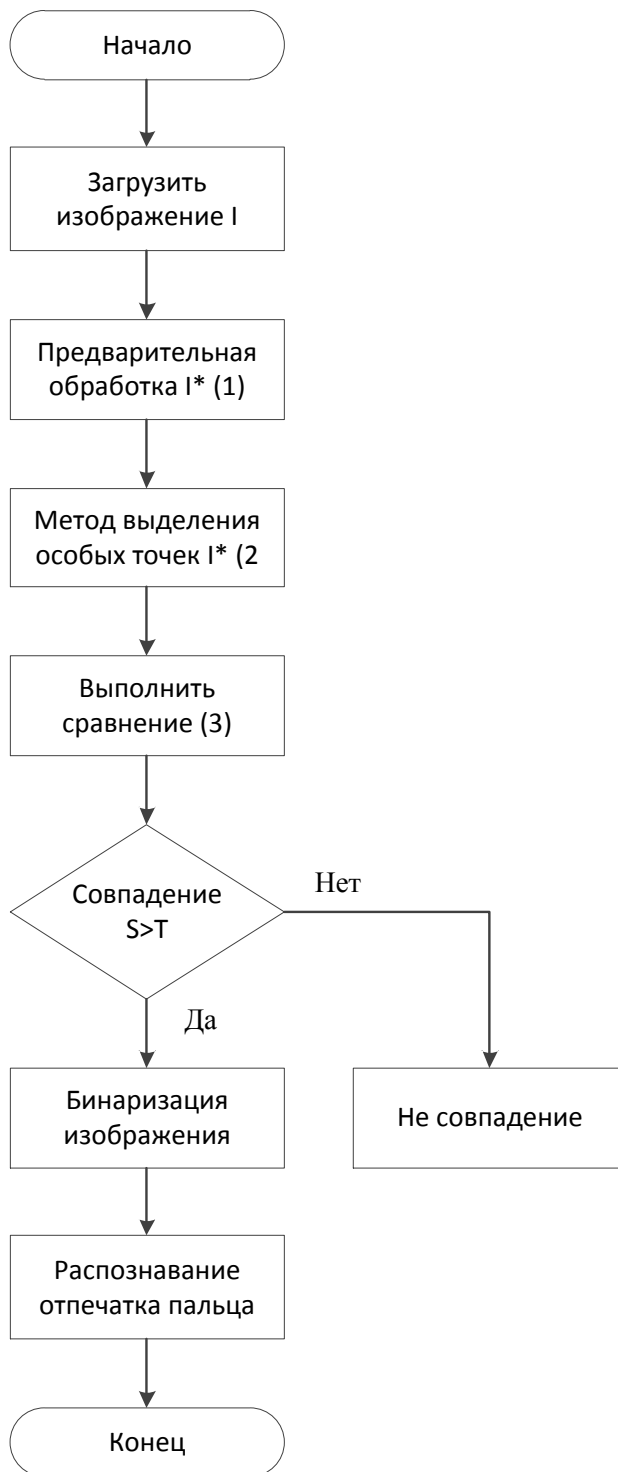


Рисунок 16 – Схема общего алгоритма идентификации по отпечаткам пальцев

Алгоритм включает 3 этапа, каждый этап разбит на отдельные процедуры. На первом шаге алгоритма получается исходное изображение, после этого начинается этап 1 предварительной обработки, который требуется для уменьшения количества шума на исходном изображении I . На этапе 2 выделяются особые точки из изображения I^* , и получается вектор признаков, необходимый для этапа 3 сравнения с шаблоном базы данных, записанных для распознавания отпечатка пальца. Этот этап возвращает индекс S совпадения отпечатков пальцев, который сравнивается с пороговым значением T . В результате, если результат сравнения положителен (Да), то фиксируется совпадение отпечатков и процедура завершается идентификацией человека, в противном случае (Нет) совпадение не находится и определяется, что не существует этого человека в базе данных.

В соответствии с общей схемой алгоритма в следующих разделах описаны процедуры, реализующие блоки схемы.

Алгоритм предварительной обработки

Такая обработка, как было сказано ранее, требуется для уменьшения количества шума на изображении. На вход процедуры предварительной обработки поступает оригинальное изображение I , внутри процедуры производится улучшение входного изображения, для чего применяется бинаризация, ориентационное, Фильтрация и т.д. Подробный алгоритм процесса представлен на рисунке 17.

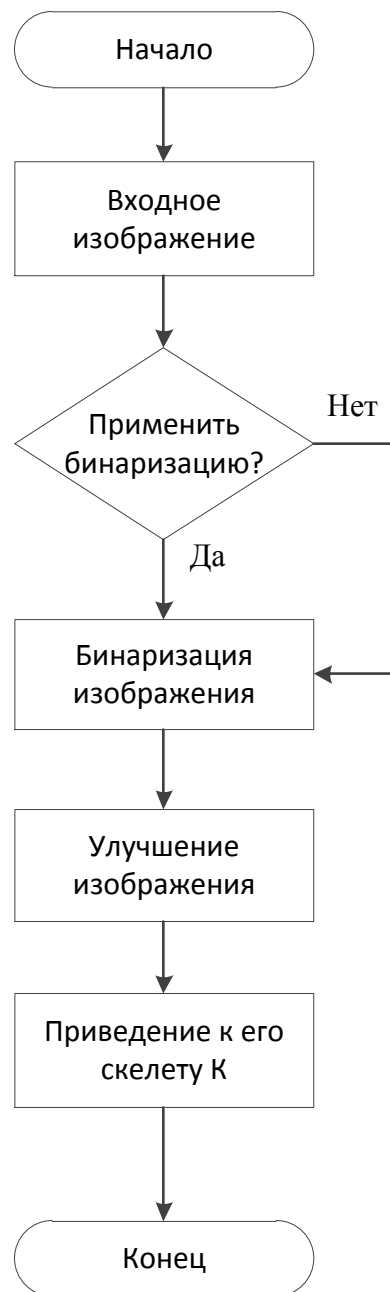


Рисунок 17 – Схема алгоритма этапа предварительной обработки

Бинаризация изображение I выполняет преобразование каждого пикселя со значением интенсивности, соответствующим оттенку серого, в двоичное значение интенсивности – черный (0) или белый (1). Самый простой подход заключается в применении глобального порога, где каждый пиксель сопоставляется.

Выделение особых точек

Бинаризация в сочетании с морфологическим прореживанием обеспечивает простую основу для выделения минуции. Далее на рисунке 18 представлена схема процесса.



Рисунок 18 – Схема алгоритма выделения особых точек

На первом шаге, чтобы найти минуций, необходимо улучшить скелет изображения K , ширина скелета должна быть строго в один пиксель. Однако, это не всегда верно, имеются некоторые места, где скелет имеет ширину в два пикселя, обычно в некоторых ошибочных местоположениях пикселей. Ошибочные пиксели определяются как один с более, чем двумя, подключенными соседями. Наличие ошибочных пикселей может изменять типа минуция.

Поэтому, прежде чем получить минуции, необходимо устранить ошибочные пиксели при сохранении связности скелета в регионах бифуркации.

Для этого проводится сканирование скелета изображения по строке из верхнего левого угла изображения в нижний правый, дальше проверяется, имеет ли пиксель значение 1, и подсчитывается сумма его четырех подключенных соседей. Если сумма окажется больше двух, он помечается как ошибочный пиксель. Далее этот ошибочный пиксель удаляется. Таким образом, после сканирования всего изображения будут удалены все ошибочные пиксели.

После прореживания минуций отпечатков пальцев, делается маркировка точек минуции. Следует заметить, что чем большее количество минуций обнаружено, тем больше вероятность получения точного результата.

Далее для удаления ошибочных минуций применяется евклидово расстояние (7), используемое для сравнения характеристик, полученных на предыдущем шаге. Если значение $d(x,y)$ меньше, чем D двух минуций в том же скелете, они удаляются оба. Здесь D – среднее расстояние между двумя параллельными соседними участками скелета.

$$d_{(x,y)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (7)$$

где $(x_i - x_j)$ и $(y_i - y_j)$ являются наборами характеристик минуций [18].

С другой стороны, для каждой минуции отпечатка пальца, вектор признаков состоит из следующих компонентов:

$$(d_n, t_n, \theta_{1(n)}), \quad (8)$$

где d_n – расстояние от n минуции до точки отсчета; t_n – тип минуции до точки отсчета, которые могут быть типа окончания или типа бифуркации.

Если характеристикой точки отсчета является $(x_{m(n)}, y_{m(n)})$, то характеристикой n минуции будет:

$$d_n = \sqrt{(x_{rp} - x_{m(n)})^2 + (y_{rp} - y_{m(n)})^2} \quad (9)$$

Для каждого окончания имеем значение 1 и бифуркации - значение 3 - и вычисляется $\theta_{1(n)}$:

$$\theta_{1(n)} = \theta_n - \theta_r, \quad (10)$$

где θ_n – угол линии, соединяющей точка отсчета и энной мелочи по отношению к горизонтальной оси; θ_r – угол линии, соединяющей точку отсчета и опорной ориентации относительно горизонтальной оси, оба параметра находятся в диапазоне $[0, 2\pi]$. После получения вектора характеристик минуций, он сохраняется как шаблон обработанного отпечатка пальца.

Сравнение с эталоном

Если даны два набора минуция для двух изображений отпечатков пальцев, алгоритм сравнения определяет по ним соответствуют наборы минуций (один и тоже же палец) или нет (разные пальцы). Примененный в этой работе алгоритм сравнения отпечатков частично вытекает из [19]. Он включает в себя две последовательных процедуры: выравнивания и сопоставления. На рисунке 19 представлен алгоритм сравнения.

Сопоставления для каждого пальца осуществляется на основании извлеченных характеристик. Пусть и два набора минуций задаются уравнениями:

$$I_1 = \{m_1, m_2, \dots, m_m\}, \quad \text{где } m_i = (x_i, y_i, \theta_i) \quad (11)$$

$$I_2 = \{m'_1, m'_2, \dots, m'_m\}, \quad \text{где } m'_i = (x'_i, y'_i, \theta'_i) \quad (12)$$

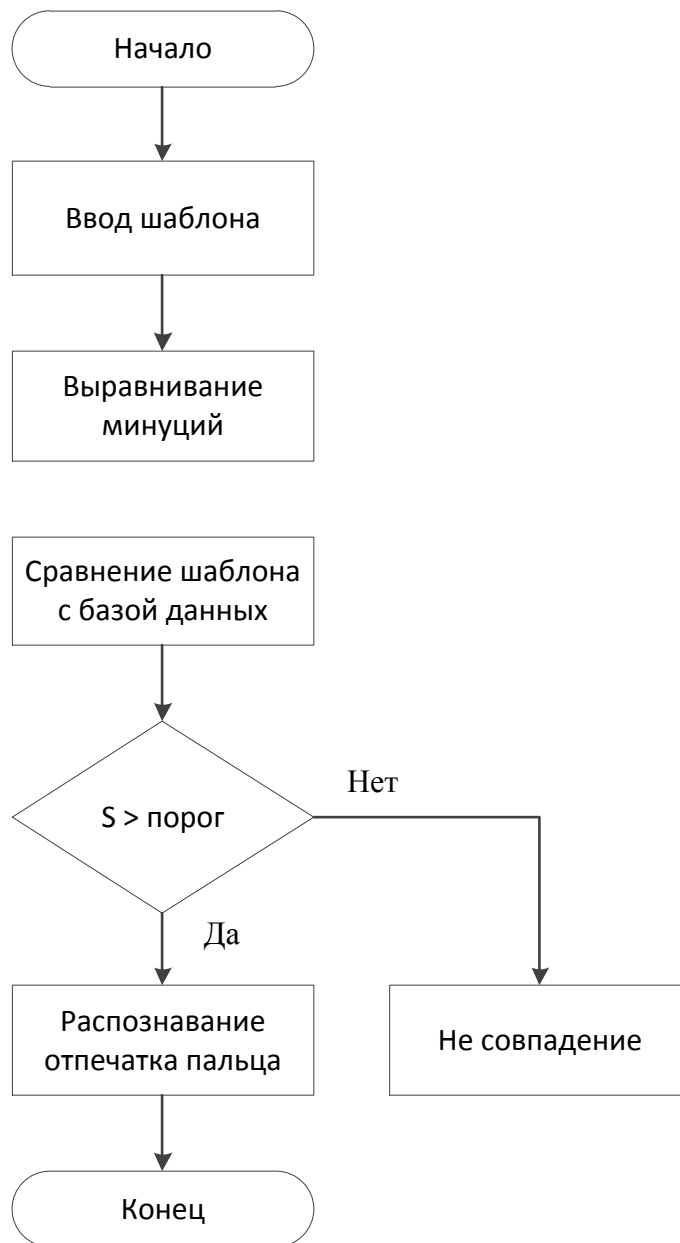


Рисунок 19 – Схема алгоритма сравнения с эталоном

Участок скелета, связанный с каждой минуцией может быть представлен серией x -координат $(x_1, x_2, \dots, x_n)$ точек на нем. Затем вычисляется сходство корреляции двух участков скелета по следующей формулы:

$$S = \frac{\sum_{i=0}^m x_i X_i}{\sqrt{\sum_{i=0}^m x_{i1} X_{i2}}} \quad (13)$$

На данном этапе $(x_i \dots x_n)$ и $(X_i \dots X_n)$ – набор x -координаты для двух минуций, которые были выбраны. Где m – одно из значений n . Далее над каждым преобразованным минуций выполняется аффинное преобразование.

Аффинное преобразование является преобразованием, которое сохраняет линейность и соотношения расстояний. Для этого принимается $M(x, y, \theta)$ в качестве опорной минущии (x_i, y_i, θ_i) отпечатков пальцев относительно $M(x, y, \theta)$. Преобразование выполняется по формуле

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = (TM)x \begin{bmatrix} x_i - x \\ y_i - y \\ \theta_i - \theta \end{bmatrix} \quad (14)$$

$$TM = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

где TM – матрица преобразования. В результате получаем новые координаты каждой минущии M и новую ось, совпадающую с направлением минущии M . Так получатся преобразованные наборы минущий I_1 и I_2 .

После этого выполняется процедура сравнения. Здесь строка (x, y, θ) используется для поиска количества совпавших пар минущий между m_i в I'_1 и минущий m_j в I'_2 считаются “сопоставленными”, если пространственное расстояние (sd) между ними меньше заданного допуска r_0 , и разница направление (dd) между ними меньше, чем угловой допуск θ_0 . При этом пространственное расстояние (sd) и разница направлений (dd) рассчитываются по формулам (15) и (16) соответственно:

$$sd = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r_0 \quad (15)$$

$$dd(|\theta_i - \theta_j|, 360 - |\theta_i - \theta_j|) \leq \theta_0 \quad (16)$$

Пусть $mm(m_i, m_j)$ – является функцией индикатора, который определяется формулой (17) и возвращает 1, когда минущии m_i и m_j сопоставляются в соответствии с приведенными выше уравнениями.

$$mm(m_i, m_j) = \begin{cases} 1, & sd(m_i, m_j) \leq r_0 \text{ и } dd(m_i, m_j) \leq \theta_0 \\ 0, & \text{иначе} \end{cases} \quad (17)$$

Далее вычисляется общее количество подобных пар минущий, используя следующее уравнение

$$\text{num}(\text{matchPointed minutiae}) = \sum mm(m_i, m_j) \quad (18)$$

Нахождение двух одинаковых пар минуций может быть не точным по причине незначительных деформаций, а также не точного квантования. Алгоритм сопоставления для выравнивания узоров минуция должен быть гибкий. Для компенсации упругости минуция удерживается в рамке вокруг каждого шаблона минуции. Если минуции, которые должны быть согласованы, находятся в установленной прямоугольной рамке и несоответствие направлений между ними маленькие, то эти минуции воспринимаются как пара подобных минуций. Так проверяется на совпадение каждая из минуций шаблона и текущего изображения. В результате вычисляется коэффициент соответствия для двух пальцев, используя уравнение

$$S = \frac{nmp}{nmt} \quad (19)$$

где S – результат совпадения; nmp – общее количество подобных пар минуций; nmt – количество минуций шаблона. Если результат совпадения (Да), то есть больше, чем пороговое значение T , которое предварительно задано, то два взятых отпечатка пальцев соответствую одному пальцу, и система распознает отпечаток пальца и отображает данные человека, связанные с шаблоном. В противном случае (Нет), т.е. совпадение не находится и определяется, что не существует этого человека в базе данных.

2.3.3 Хэширование данных

Хэш-функции – это функции, предназначенные для «сжатия» произвольного сообщения или набора данных, записанных в некоторую битовую комбинацию фиксированной длины, называемую сверткой. Хэш-функции имеют разнообразные применения при проведении статистических экспериментов, при тестировании логических устройств, при построении алгоритмов быстрого поиска и проверки целостности записей в базах данных.

Основным требованием к хэш-функциям является равномерность распределения их значений при случайном выборе значений аргумента.

Хэш-функцией называется всякая функция $h: X \rightarrow Y$, легко вычисляемая и такая, что для любого сообщения M значение $h(M) = H$ (свертка) имеет фиксированную битовую длину. X — множество всех сообщений, Y — множество двоичных векторов фиксированной длины.

Для получения значения $h(M)$ сообщение сначала разбивается на блоки длины m . При этом если длина сообщения не кратна m , то последний блок неким специальным образом дополняется до полного блока. Затем к полученным блокам $M_1, M_2, \dots, M_n$ применяют следующую последовательную процедуру вычисления свертки:

$$\begin{aligned} H_0 &= v, \\ H_i &= f(M_i, H_{i-1}) \quad i = 1, \dots, N, \\ h(M) &= H_N, \end{aligned} \quad (20)$$

где v — некоторая константа, часто ее называют инициализирующим вектором.

Выделяют два вида криптографических хэш-функций: ключевые, бесключевые. Ключевые хэш-функции называют кодами аутентификации сообщений. Они дают возможность без дополнительных средств гарантировать как правильность источника данных, так и целостность данных в системах с доверяющими друг другу пользователями.

Бесключевые хэш-функции называются кодами обнаружения ошибок. Они дают возможность с помощью дополнительных средств (например, шифрования) гарантировать целостность данных.

Для криптографических хэш-функций есть дополнительные условия, которые отличают их от всех остальных:

- необратимость: для заданного значения хэш-функции m должно быть вычислительно неосуществимо найти блок данных X , для которого $H(X)=m$.

- стойкость к коллизиям первого рода: для заданного сообщения M должно быть вычислительно неосуществимо подобрать другое сообщение N , для которого $H(N) = H(M)$.
- стойкость к коллизиям второго рода: должно быть вычислительно неосуществимо подобрать пару $\sim(M, M')$, имеющих одинаковый хэш.

Алгоритмы хеш-функций

Алгоритмы CRC16/32 — контрольная сумма (не криптографическое преобразование).

Алгоритм MD5 имел большую популярность, но первые предпосылки взлома появились еще в конце девяностых, и сейчас его популярность стремительно падает.

Алгоритмы линейки SHA сейчас широко распространены. Идет активный переход от SHA-1 к стандартам версии SHA-2. SHA-2 — собирательное название алгоритмов SHA224, SHA256, SHA384 и SHA512. SHA224 и SHA384 являются аналогами SHA256 и SHA512 соответственно, только после расчета свертки часть информации в ней отбрасывается. Использовать их стоит лишь для обеспечения совместимости с оборудованием старых моделей [20].

Для реализации хеш-функций отлично подойдет модуль стандартной библиотеки Python hashlib. Он включает в себя безопасные алгоритмы хеширования FIPS, такие как SHA1, SHA224, SHA256, SHA384, а также SHA512 и MD5. Python также поддерживает функции хеширования adler32 и crc32, но они содержатся в модуле zlib. Одно из самых популярных применений хеширования это хранение хеша пароля, вместо самого пароля. Конечно, хеш должен быть хорошим, в противном случае он может быть расшифрован.

Другой популярный случай, в котором применяется хеширование — это хеширование файла, с последующей отправкой файла и его хеша по отдельности. Получатель файла может запустить хеш в файле, чтобы убедиться в том, что файл соответствует отправленному хешу. Если это так, значит никто не менял файл, когда он был отправлен.

2.4 Выводы по главе 2

В данной главе, посвященной проектированию безопасного биометрического доступа, были рассмотрены классификаторы изображений отпечатка пальца. Были рассмотрены их особенности, достоинства и недостатки. Также были рассмотрены алгоритмы безопасного биометрического доступа с помощью отпечатка пальца.

Кроме того, в данной главе обоснована важность предварительной обработки данных и были рассмотрены основные алгоритмы идентификации, которые позволили бы повысить точность идентификации. Также было рассмотрено хеш-функции и их алгоритмы.

Следующим за этапом проектирования является этап непосредственно разработки безопасного биометрического метода, в котором предполагается на практике оценить целесообразность предложенных методов повышения качества распознавания биометрических образов, целесообразность введенных способов извлечения новых признаков и степень их влияния на результат идентификации.

Глава 3 Программная реализация системы безопасного биометрического доступа к киберфизическим системам

3.1 Структура системы безопасного биометрического доступа

Для реализации безопасного биометрического доступа к киберфизическим системам необходима разработка нескольких компонентов: сбор данных, анализа отпечатка пальца, формирование эталонного биометрического образа пользователя, хеширование полученных данных.

Компонент сбора данных загружает исходное изображение отпечатка пальца, который будет предметом анализа для распознавания человека.

Компонент обработки выполняет предварительную обработку введенного изображения отпечатка пальца, реализует один из алгоритмов отпечатка пальца.

Компонент анализа данных и формирования биометрического образа должен реализовывать один из алгоритмов анализа отпечатка пальца, выполнять предварительную обработку данных, предложенных в главе 2.

Компонент хеширования данных сравнивает изображение с эталонным и подсчитывает расстояние Хэмминга, и сохраняет хэш для дальнейшего использования.

3.2 Выбор средств программной реализации системы

Для достижения целей наиболее распространенным решением является применение средств языка R или Python совместно с набором библиотек анализа данных.

Язык R используется как статистическое программное обеспечение для анализа данных и фактически стал стандартом для статистических программ. Он поддерживает широкий спектр статистических и численных методов и обладает хорошей расширяемостью с помощью пакетов. Пакеты представляют собой библиотеки для работы специфических функций или специальных

областей применения. В базовую поставку R включен основной набор пакетов, а всего по состоянию на 2016 год доступно более 7000 пакетов.

Альтернативным инструментом анализа данных является использование языка Python совместно с библиотеками, предназначенными для решения различных вычислительных задач.

Успех языка Python в качестве платформы для научных расчетов отчасти обусловлен простоте интеграции с кодом на C, C++ и Fortran. Кроме того, имеются средства применения Python-сценариев в программах на языках C#, Java и JavaScript. Помимо этого, свой популярностью в области анализа данных язык обязан широкому набору библиотек для научных и статистических расчетов, среди которых можно выделить самые широко распространенные: NumPy, pandas, matplotlib, SciPy и Keras.

NumPy – основной пакет для математических расчетов, который предоставляет:

- быстрый и эффективный объект многомерного массива;
- широкий набор функций обработки массивов и операций над массивами;
- средства сериализации и десериализации массивов из различных форматов.

Библиотека pandas предоставляет структуры данных и функции для простой и эффективной обработки двумерных таблиц, в которых строки и столбцы имеют метки.

Библиотека matplotlib – предоставляет широкий набор средств для создания графиков и для визуализации данных, как в двухмерном, так и в трехмерном виде.

SciPy – собрание пакетов, предназначенных для решения различных стандартных вычислительных задач, таких как:

- численного интегрирования и решения дифференциальных уравнений;

- задачи линейной алгебры и разложения таблиц;
- алгоритмы оптимизации функций;
- стандартные непрерывные и дискретные распределения вероятностей, различные статистические критерии и дополнительные описательные статистики[21].

Keras – открытая нейросетевая библиотека, написанная на языке Python. Она представляет собой надстройку над фреймворками DeepLearning4j, TensorFlow и Theano [22]. Нацелена на оперативную работу с сетями глубинного обучения, при этом спроектирована так, чтобы быть компактной, модульной и расширяемой. Эта библиотека содержит многочисленные реализации широко применяемых блоков нейронных сетей, таких как:

- слои;
- целевые и передаточные функции;
- оптимизаторы;
- множество инструментов для упрощения работы с изображениями и текстом.

Исходя из вышесказанного, для реализации компонента анализа отпечатка пальца был выбран именно язык Python в сочетании с описанными библиотеками, предназначенными для анализа данных.

3.3 Описание разработанного метода безопасного биометрического доступа к киберфизическим системам

После получения изображения отпечатка пальца будет выполнена бинаризация. Часть кода, связанная с бинаризацией, приведен ниже (Листинг кода представлен в Приложение А).

```
def binary(image):
    bImage=[]
    for i in range(image.size[0]):
        tmp=[]
        for j in range(image.size[1]):
            t=image.getpixel((i,j))
            p=t[0]*0.3+t[1]*0.59+t[2]*0.11
        if p>128:
```

```

        p=1
    else:
        p=0
    tmp.append(p)
    bImage.append(tmp)
    return bImage

```

Следующим действием является скелетизация. Используемые шаблоны соответствуют матрице 3x3, где центральный элемент является текущим пикселем в обходе изображения.

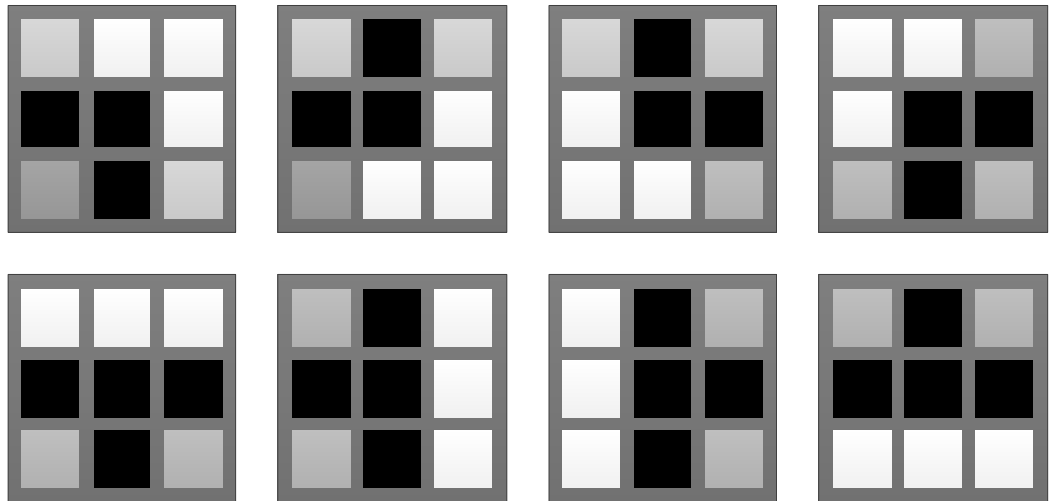


Рисунок 20 – Скелетизация, основная часть

Серым цветом обозначены пиксели, цвет которых не имеет значения. На рисунке 20 представлены шаблоны, которые являются основной частью.

На рисунке 21 представлено устранение «шума». Они могут быть повернуты на 90, 180 и 270 градусов, и ищутся вторым обходом изображения.

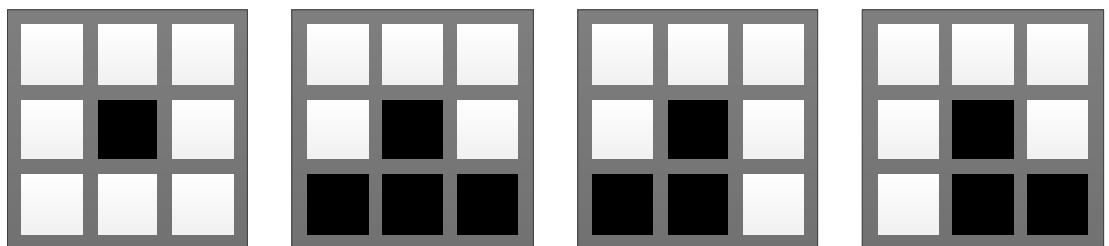


Рисунок 21 – Скелетизация, устранение шума

Если встречается шаблон, то центральный пиксель окрашивается в белый цвет, то есть не принадлежит скелету. Обход продолжается до тех пор, пока остаются возможности удаления.

Далее приведена часть кода, связанная со скелетизацией (Листинг кода представлен в Приложение А). Вызов процедуры скелетизации, где на входе список списков после бинаризации.

```
def tmpDelete(image):
    w=len(image)
    h=len(image[0])
    count=1
    while count!=0:
        count=delete(image,w,h)
        if count:
            delete2(image,w,h)
```

Удаление пикселя по основному набору, возврат количества удаленных пикселей.

```
def delete(image,w,h):
    count=0
    for i in range(1,h-1):
        for j in range(1,w-1):
            if image[j][i]==0:
                if deletable(image,j,i):
                    image[j][i]=1
            count+=1
    return count
```

Удаление пикселя по шумовому набору.

```
def delete2(image,w,h): #удаление пикселя по шумовому набору
    for i in range(1,h-1):
        for j in range(1,w-1):
            if image[j][i]==0:
                if deletable2(image,j,i):
                    image[j][i]=1
```

Следующим шагом идет выделение особых точек. Если в окрестности из 8 точек, есть только одна черная, то это конечная точка. Если их 2, то это точка линии. Если их 3 – точка ветвления.

Далее приведена часть кода (Листинг кода представлен в Приложение А). Подсчет количества черных точек в окрестности

```
def checkThisPoint(image, x, y):
    c=0
    for i in range(x-1,x+2):
        for j in range(y-1,y+2):
            if image[i][j]==0:
                c+=1
    return c-1
```

Формирование списков точек ветвления и конечных точек.

```
def findCheckPoint(image):
    x=len(image)
    y=len(image[0])
    branchPoint=[]
    endPoint=[]
    for i in range(x):
        for j in range(y):
            if image[i][j]==0:
                t=checkThisPoint(image, i, j)
                if t==1:
                    endPoint.append((i,j))
                if t==3:
                    branchPoint.append((i,j))
    return (branchPoint, endPoint)
```

Сделано удаление близко стоящих (10x10) точек ветвления и конечных точек из-за не полного избавления от шумов, что приводило к появлению отростков, которые распознавались как особые точки.

После удаления шума требуется произвести сравнение точек – поиск точки, которая попадает в окрестность 30x30. На вход подаются точки эталонного и проверяемого значения, на выходе отображено число точек, которые совпали, и общее число точек.

```
def matchPointing(r, v):
    allPoint=0
    matchPoint=0
    for i in v[0]:
        x=range(i[0]-15,i[0]+15)
        y=range(i[1]-15,i[1]+15)
        allPoint+=1
        for j in r[0]:
            if j[0] in x and j[1] in y:
                matchPoint+=1
        break
    for i in v[1]:
        x=range(i[0]-15,i[0]+15)
        y=range(i[1]-15,i[1]+15)
        allPoint+=1
        for j in r[1]:
            if j[0] in x and j[1] in y:
                matchPoint+=1
        break
    return (matchPoint,allPoint)
```

Завершающим этапом является хеширование полученных результатов. При повторном анализе одного и тоже же пользователя сравнивается уже не

отпечаток пальца, а хеш-код. Для этого используется стандартный пакет hashlib.

3.4 Направления дальнейшего развития системы

В рамках данной работы в качестве алгоритма анализа образцов отпечатка пальца был использован самый простой сравнение по особым точкам. Результаты исследования показали, что, использование данного простого метода в сочетании с хешированием, позволяет добиться наиболее безопасного метода хранения данных. Применение более совершенных методов в совокупности с предложенными способами позволили бы в дальнейшем значительно улучшить достигнутые результаты.

Так же перспективным направлением дальнейшего исследования является получение управления удаленным доступом к автоматизированным комплексам армейского и специального назначения. В рамках данного направления развития возможно строить предположения использования мобильного устройства, в частности SIM-карту, на которой будет осуществляться обработка отпечатка пальца пользователя для получения средств удаленного доступа.

Однако основной целью проведенного исследования и разработанного метода безопасного биометрического доступа к киберфизическим системам в дальнейшем является разработка системы идентификации пользователя по биометрическому образцу отпечатка пальца для управления удаленным доступом к автоматизированному комплексу специального назначения, такой как, например роботизированные комплексы. Конечная цель разработки в данном направлении – разработка программного комплекса для получения доступа к автоматизированным комплексам армейского и специального назначения.

Заключение

В рамках проведения исследования по разработке безопасного биометрического доступа к киберфизическим системам были поставлены следующие задачи:

- общая постановка задачи идентификации;
- обзор существующих систем на основе анализа отпечатка пальца;
- выбор методов и алгоритмов реализации анализа отпечатка пальца;
- проектирования и реализации безопасного биометрического доступа.

При выполнении данной работы были достигнуты следующие результаты:

- сформулировано понятие киберфизических систем, биометрической идентификация;
- произведен обзор существующих систем идентификации пользователей по биометрической идентификации (в частности по отпечатку пальца), который показал отсутствие в широком доступе безопасных систем идентификации пользователей по отпечатку пальца;
- произведен обзор возможных классификаторов изображений отпечатков пальца, а также обзор существующих алгоритмы безопасного биометрического доступа с помощью отпечатка пальца;
- был реализован алгоритм сравнения по особым точкам;
- на основании проведенных исследований был разработан безопасный биометрический доступ к киберфизическим системам, который позволяет: сбор данных, анализ отпечатка пальца, формирование эталонного биометрического образа пользователя, хеширование полученных данных.

В качестве направлений дальнейшего развития были выделены:

- реализация системы управления удаленным доступом к автоматизированным комплексам на основе более сложных алгоритмов;
- анализ открытых средств связи для получения средств удаленного доступа;
- разработка программного комплекса получения управления удаленным доступом к автоматизированным комплексам армейского и специального назначения.

Список использованных источников

1. Ахметов Б.С., Иванов А.И., Фунтиков В.А., Безяев А.В., Малыгина Е.А. Технология использования больших нейронных сетей для преобразования нечетких биометрических данных в код ключа доступа. – Алматы: ТОО «Издательство LEM» КазНТУ, 2014. – 11 с.
2. [Электронный ресурс]. URL: http://www.tadviser.ru/index.php/Cyber-Physical_System_CPS (Дата обращения: 14.01.2018 г.).
3. [Электронный ресурс]. URL: <https://www.ibm.com/developerworks/ru/library/ba-cyber-physical-systems-and-smart-cities-iot/> (Дата обращения: 14.01.2018 г.).
4. [Электронный ресурс]. URL: <https://www.samara.damofons.info/biometric-hesky-zamok/l-9000.php> (Дата обращения: 14.01.2018 г.).
5. [Электронный ресурс]. URL: https://www.hi-tech.mail.ru/review/HTC_One_Max-rev/ (Дата обращения: 14.01.2018 г.).
6. A.K. Jain, S. Prabhakar, L. Hong, "A Multichannel Approach to Fingerprint Classification"/ IEEE TRANSACTIONS ON PATTERN ANALYSIS and MACHINE INTELLIGENCE, 1999, p.348-359.
7. Портал искусственного интеллекта. Автоассоциативная память [Электронный ресурс]. URL: <http://neuronus.com/theory/963-avtoassotsiativnaya-pamyat.html> (Дата обращения: 03.02.2018 г.).
8. Портал искусственного интеллекта Нейронные сети Хэмминга [Электронный ресурс]. URL: <http://neuronus.com/theory/971-nejronnye-seti-khemminga.html> (Дата обращения: 03.02.2018 г.).
9. A. Senior, "A Combination Fingerprint Classifier"/ IEEE TRANSACTIONS ON PATTERN ANALYSIS and MACHINE INTELLIGENCE, 2001, p. 1165-1174.
10. Анализ методов распознавания отпечатков пальца. [Электронный ресурс]. URL: http://www.nbuu.gov.ua/old_jrn/natural/SOI/2010_6/Rykanov.pdf (дата обращения 15.02.2018).

11. Патрик Э. Основы теории распознавания образов. – М.: Советское радио, 1980.
12. Фам З. Т. Некоторые вопросы применения методов сравнения отпечатков пальцев для биометрических систем идентификации личности. Режим доступа <http://cyberleninka.ru/article/n/nekotorye-voprosy-primeneniya-metodov-sravneniya-otpechatkov-paltsev-dlya-biometricheskih-sistem-identifikatsii-ichnosti>. (дата обращения 05.03.2018).
13. D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, "Handbook of Fingerprint Recognition"/ Springer, New York, 2003.
14. Анализ методов распознавания отпечатков пальца. [Электронный ресурс]. URL: http://www.nbuu.gov.ua/old_jrn/natural/SOI/2010_6/Rykanov.pdf (дата обращения 17.03.2018).
15. Гонзалес Р, Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072 с.].
16. Hong L., Wan Y., and Jain A. Fingerprint Image Enhancement. [Электронный ресурс]. URL:<http://www.math.tau.ac.il/~turkel/imagepapers/fingerprint.pdf>. (дата обращения 17.03.2018).
17. Kumar, D. Ashok, and T. Ummal Sariba Begum. "A Comparative Study on Fingerprint MatchPointing Algorithms for EVM." Journal of Computer Sciences and Applications 1, no. 4 (2013): p.57 Available at: <http://pubs.sciepub.com/jcsa/1/4/1/>.
18. Sangita K. C. An algorithm for fingerprint enhancement &matchPointing. International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622. (VNCET3 Mar 12). [Электронный ресурс]. URL:http://www.ijera.com/special_issue/VNCET_Mar_2012/60.pdf, (дата обращения 17.03.2018).

19. Анализ методов распознавания отпечатков пальца. Режим доступа: http://www.nbuv.gov.ua/old_jrn/natural/SOI/2010_6/Rykanov.pdf pdf (дата обращения 19.03.2018. Российский стандарт — ГОСТ 34.11-94).

21. Маккинни У. Python и анализ данных. —М.:ДМК Пресс, 2015. — 482 с.

22. This Is What Keras Different, According To Its Author. Режим доступа: www.forbes.com/sites/quora/2016/08/25/this-is-what-makes-keras-different-according-to-its-author/#54d878ae66cf (дата обращения 20.03.2018).

Приложение А. Листинг программы

Main.pyw

```
from Tkinter import *
import tkFileDialog
import correlate
import points

def start():
    '''if F.get()=='C':
window=Toplevel(root)
window.title("Fingerprint Check")
        r=correlate.checkFinger(inPut.get(),outPut.get())
Label(window, text="MatchPoint: "+str(r*100)+"%").pack()
    if F.get()=='P':'''
    points.checkFinger(inPut.get(),outPut.get())

def openRef(event):
    options={}
    options['defaultextension'] = ''
    options['filetypes'] = [('text files', '.txt')]
    options['initialdir'] = 'C:\\'
    options['parent'] = root
    options['title'] = 'Open Input File'

    input=tkFileDialog.askopenfilename()
    if input:
        inPut.set(input)

def openVer(event):
    options={}
    options['defaultextension'] = ''
    options['filetypes'] = [('text files', '.txt')]
    options['initialdir'] = 'C:\\'
    options['parent'] = root
    options['title'] = 'Open Output File'

    output=tkFileDialog.askopenfilename()
    if output:
        outPut.set(output)

global root
root=Tk()
```

```

root.title("Defender")

global inPut
global outPut
global autoGraph
inPut=StringVar()
inPut.set('')
outPut=StringVar()
outPut.set('')
autoGraph=StringVar()
autoGraph.set('')

inEntry=Entry(root, textvariable=inPut)
outEntry=Entry(root, textvariable=outPut)

getInFile=Button(root, text='Open Reference')
getOutFile=Button(root, text='Open Verificable')

getInFile.bind('<1>',openRef)
getOutFile.bind('<1>',openVer)

'''
global F
F=StringVar()
F.set('C')
Cor=Radiobutton(root,text='Correlate', variable=F, value='C')
Pnt=Radiobutton(root,text='Control      Points',      variable=F,
value='P')
'''
sign=Button(root, text='Start', command=start)

inEntry.grid(row=0,column=0)
outEntry.grid(row=1,column=0)

getInFile.grid(row=0,column=1)
getOutFile.grid(row=1,column=1)

'''Cor.grid(row=2,column=0)
Pnt.grid(row=2,column=1)'''
sign.grid(row=3,column=1)

root.mainloop()

```

Points.py

```
from Tkinter import *
import Image
import templateSkeletize2 as sk

def binary(image):
    bImage=[]
    for i in range(image.size[0]):
        tmp=[]
        for j in range(image.size[1]):
            t=image.getpixel((i,j))
            p=t[0]*0.3+t[1]*0.59+t[2]*0.11
        if p>128:
            p=1
        else:
            p=0
        tmp.append(p)
    bImage.append(tmp)
    return bImage

def __removeDouble(x,y):
    z=[]
    for i in x:
        c=True
    for j in y:
        if i==j:
            c=False
    if c:
        z.append(i)
    for i in y:
        c=True
    for j in x:
        if i==j:
            c=False
    if c:
        z.append(i)
    return z

def delNoisePoint(r):
    tmp=[]
    tmp2=[]
    for i in r[1]:
        x=range(i[0]-5,i[0]+5)
```

```

        y=range(i[1]-5,i[1]+5)
    for j in r[0]:
        if j[0] in x and j[1] in y:
            tmp.append(i)
            tmp2.append(j)
    return (__removeDouble(r[0],tmp2),__removeDouble(r[1],tmp))

defmatchPointing(r, v):
    allPoint=0
    matchPoint=0
    for i in v[0]:
        x=range(i[0]-15,i[0]+15)
        y=range(i[1]-15,i[1]+15)
    allPoint+=1
    for j in r[0]:
        if j[0] in x and j[1] in y:
            matchPoint+=1
    break
    for i in v[1]:
        x=range(i[0]-15,i[0]+15)
        y=range(i[1]-15,i[1]+15)
    allPoint+=1
    for j in r[1]:
        if j[0] in x and j[1] in y:
            matchPoint+=1
    break

    return (matchPoint,allPoint)

def checkThisPoint(image, x, y):
    c=0
    for i in range(x-1,x+2):
        for j in range(y-1,y+2):
            if image[i][j]==0:
                c+=1
    return c-1

def findCheckPoint(image):
    x=len(image)
    y=len(image[0])
    branchPoint=[]
    endPoint=[]
    for i in range(x):

```

```

for j in range(y):
    if image[i][j]==0:
        t=checkThisPoint(image, i, j)
    if t==1:
        endPoint.append((i,j))
    if t==3:
        branchPoint.append((i,j))
    return (branchPoint, endPoint)

def checkFinger(r, v):
    reference=Image.open(r)

    ref=binary(reference)

    sk.tmpDelete(ref)
    rp=findCheckPoint(ref)
    rp=delNoisePoint(rp)

    verf=Image.open(v)

    ver=binary(verf)

    sk.tmpDelete(ver)
    vp=findCheckPoint(ver)
    vp=delNoisePoint(vp)

    res=matchPointingPoint(rp, vp)
    r=(res[0]/(res[1]*1.))*100

    root=Tk()
    w=len(ver)
    h=len(ver[0])
    C=Canvas(root, width=w*2, height=h)

    for i in range(w):
        for j in range(h):
            if ref[i][j]==0:
                C.create_line([(i,j), (i+1,j+1)])
            if ver[i][j]==0:
                C.create_line([(i+w+1,j+1), (i+w,j)])
    for i in rp[0]:

```

```

        C.create_oval([(i[0]-3,i[1]-
3),(i[0]+3,i[1]+3)],outline="#ff0000")
        for i in rp[1]:
            C.create_rectangle([(i[0]-3,i[1]-
3),(i[0]+3,i[1]+3)],outline="#0000ff")
            for i in vp[0]:
                C.create_oval([(i[0]-3+w,i[1]-
3),(i[0]+3+w,i[1]+3)],outline="#ff0000")
                for i in vp[1]:
                    C.create_rectangle([(i[0]-3+w,i[1]-
3),(i[0]+3+w,i[1]+3)],outline="#0000ff")
                    C.create_text((w, h*0.95), fill="#009900",
text=str(r)+"%",font='Arial,72')
            C.pack()
    root.mainloop()

```

Skeletize.py

```

def tmpDelete(image):
    w=len(image)
    h=len(image[0])
    count=1
    while count!=0:
        count=delete(image,w,h)
        if count:
            delete2(image,w,h)

def delete(image,w,h):
    count=0
    for i in range(1,h-1):
        for j in range(1,w-1):
            if image[j][i]==0:
                if deletable(image,j,i):
                    image[j][i]=1
                count+=1
    return count

def delete2(image,w,h):
    for i in range(1,h-1):
        for j in range(1,w-1):
            if image[j][i]==0:
                if deletable2(image,j,i):
                    image[j][i]=1

```

```

def fringe(a):
    t=[[1,1,1,1,0,1,1,1,1],

        [1,1,1,1,0,1,1,0,0],
        [1,1,1,0,0,1,0,1,1],
        [0,0,1,1,0,1,1,1,1],
        [1,1,0,1,0,0,1,1,1],

        [1,1,1,1,0,1,0,0,1],
        [0,1,1,0,0,1,1,1,1],
        [1,0,0,1,0,1,1,1,1],
        [1,1,1,1,0,0,1,1,0],

        [1,1,1,1,0,1,0,0,0],
        [0,1,1,0,0,1,0,1,1],
        [0,0,0,1,0,1,1,1,1],
        [1,1,0,1,0,0,1,1,0]]
    for i in t:
        if a==i:
            return True

def check(a):
    t123457=[1,1,0,0,1,0]
    t013457=[1,1,1,0,0,0]
    t134567=[0,1,0,0,1,1]
    t134578=[0,0,0,1,1,1]
    t0123457=[1,1,1,0,0,0,0]
    t0134567=[1,0,1,0,0,1,0]
    t1345678=[0,0,0,0,1,1,1]
    t1234578=[0,1,0,0,1,0,1]

    t=[a[1],a[2],a[3],a[4],a[5],a[7]]
    if t == t123457:
        return True
    t=[a[0],a[1],a[3],a[4],a[5],a[7]]
    if t == t013457:
        return True
    t=[a[1],a[3],a[4],a[5],a[6],a[7]]
    if t == t134567:
        return True
    t=[a[1],a[3],a[4],a[5],a[7],a[8]]
    if t == t134578:
        return True

```

```

        t=[a[0],a[1],a[2],a[3],a[4],a[5],a[7]]
    if t == t0123457:
    return True
        t=[a[1],a[3],a[4],a[5],a[6],a[7],a[8]]
    if t == t1345678:
    return True
        t=[a[0],a[1],a[3],a[4],a[5],a[6],a[7]]
    if t == t0134567:
    return True
        t=[a[1],a[2],a[3],a[4],a[5],a[7],a[8]]
    if t == t1234578:
    return True

def deletable(image,x,y):
    a=[]
    for i in range(y-1,y+2):
    for j in range(x-1,x+2):
    a.append(image[j][i])
    return check(a)

def deletable2(image,x,y):
    a=[]
    for i in range(y-1,y+2):
    for j in range(x-1,x+2):
    a.append(image[j][i])
    return fringe(a)

```