



ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

**Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал**

Кафедра математики, информационных систем и технологий
Направление подготовки 09.03.02 Информационные системы и технологии
(код, наименование направления подготовки/специальности)
Форма обучения очная

«К ЗАЩИТЕ ДОПУЩЕН(А)»
Заведующий кафедрой

(подпись)
Черняева С. Н.
(ФИО)
20

Выпускная квалификационная работа

Обучающегося Пташника Романа Олеговича
(фамилия, имя, отчество)
Вид работы выпускная квалификационная работа бакалавра
(выпускная квалификационная работа бакалавра, специалиста, магистра)

Пояснительная записка

Тема Разработка приложения учета
рабочего времени на основе распознавания лиц
(на примере ООО «Рубеж безопасности»)
(полное название темы квалификационной работы, в соответствии с приказом об утверждении тематики ВКР)

Руководитель работы к.т.н. доцент Матыцина И.А.
(должность, подпись, фамилия, инициалы, дата)
Консультант _____
(при наличии) (должность, подпись, фамилия, инициалы, дата)
Консультант _____
(должность, подпись, фамилия, инициалы, дата)
Обучающийся Пташник Р.О.
(подпись, фамилия, инициалы, дата)

ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

**Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал**

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии

(код, наименование направления подготовки/специальности)

Форма обучения очная

УТВЕРЖДАЮ
Заведующий кафедрой

(подпись)

Черняева С. Н.

(ФИО)

2024

**Задание
на выпускную квалификационную работу**

Вид работы ВКР бакалавра

(ВКР бакалавра, ВКР специалиста, ВКР магистра)

Обучающемуся Пташнику Роману Олеговичу

(фамилия, имя, отчество)

Тема Разработка приложения учета рабочего времени

на основе распознавания лиц

(на примере ООО «Рубеж безопасности»)

Утверждена приказом ректора университета от _____ 20____, № _____

Срок сдачи законченной работы _____ 20____

Исходные данные (или цель ВКР):

Перечень подлежащих исследованию, разработке, проектированию вопросов (краткое содержание ВКР):

(актуальность темы, цели и задачи ВКР; аналитический обзор литературных источников; постановка задачи исследования, разработки, проектирования; содержание процедуры исследования, разработки, проектирования; обсуждение результатов; дополнительные вопросы, подлежащие разработке; заключение – выводы по работе в целом, оценка степени решения поставленных задач, практические рекомендации; и др.)

- Введение. Актуальность выбранной темы, цель и задачи ВКР
(наименование вопроса, раздела и его краткое содержание)
- Исследовательский раздел. Пример существующих технологий распознавания лиц. Описание выбранных технологий и инструментов . Методы распознавания лиц. Метод Виолы-Джонса. Метод локального бинарного шаблона. Общая характеристика предприятия ООО «Рубеж безопасности»
(наименование вопроса, раздела и его краткое содержание)
- Проектный раздел. Описание стадий процессов и их взаимосвязь. Техническое задание на моделирование. Проектирование системы. Разработка архитектуры системы. Проектирование базы данных. Проектирование модуля распознавания лиц. Разработка основной функции. Пользовательский интерфейс. Процесс тестирования системы. Обзор работоспособности функции определения. Инструкция пользователю по работе с программным обеспечением
(наименование вопроса, раздела и его краткое содержание)
- Заключение
(наименование вопроса, раздела и его краткое содержание)
- Список используемых источников
(наименование вопроса, раздела и его краткое содержание)
- Приложение А
(наименование вопроса, раздела и его краткое содержание)

Перечень графического материала (или презентационного материала):

1. Титульный лист
2. Цель и задачи ВКР
3. Описание предметной области
4. Создание Базы данных
5. Разработка алгоритмов распознавания лиц
6. Разработка системы «Вход-Выход»
7. Система безопасности данных
8. Интерфейс пользователя
9. Результаты ВКР

Консультанты по разделам ВКР (при наличии):

1. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
2. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
3. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

Дата выдачи задания: _____ 20__

Задание согласовано и принято к исполнению: _____ 20__

Руководитель ВКР: к.т.н. доцент Матыцина Ирина Александровна
(должность, ученая степень, ученое звание, ФИО) (подпись)

Обучающийся: ИТ-4-1 Пташник Роман Олегович
(учебная группа, ФИО) (подпись)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ	7
1.1. Пример существующих технологий распознавания лиц	7
1.2. Описание выбранных технологий и инструментов	11
1.3. Методы распознавания лиц	14
1.3.1. Метод Виолы-Джонса	16
1.3.2. Метод локального бинарного шаблона	17
1.4. Общая характеристика предприятия ООО «Рубеж безопасности»	21
2. ПРОЕКТНЫЙ РАЗДЕЛ	24
2.1. Описание стадий процессов и их взаимосвязь	24
2.2. Техническое задание на моделирование	26
2.3. Проектирование системы	28
2.4. Разработка архитектуры системы	29
2.5. Проектирование базы данных	31
2.6. Проектирование модуля распознавания лиц	34
2.7. Разработка основной функции	34
2.8. Пользовательский интерфейс	37
2.9. Процесс тестирования системы	39
2.10. Обзор работоспособности функции определения	41
2.11. Инструкция пользователю по работе с программным обеспечением	49
ЗАКЛЮЧЕНИЕ	53
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	54
ПРИЛОЖЕНИЕ А	56

ВВЕДЕНИЕ

Сегодня большое количество предприятий интегрируют в организацию системы контроля и доступа. При помощи систем биометрии лица, можно более точно определять и отслеживать рабочий процесс всех сотрудников предприятия.

На данный момент для распознавания личности на предприятии используются электронные турникеты, которым необходимо считывать карты или отпечатки пальцев сотрудника, но технологии биометрии лица кардинально изменяют данную систему, предприятия зачастую переходят на более точные методы идентификации, примером служит приложения учета рабочего времени на основе распознавания лиц.

Отталкиваясь от этого была выявлена цель выпускной квалификационной работы: Разработать приложения для учёта рабочего времени на основе распознавания лиц.

Для выполнения цели требуется:

- Изучить классификацию алгоритмов распознавания;
- Исследовать существующие систем распознавания на рынке;
- Определить основные инструменты для разработки приложения;
- Реализовать систему распознавания.

В данной работе должны быть описаны все этапы реализации программы, а также должны быть рассмотрены и описан все использующиеся методы, необходимые для создания проекта.

Распознавание лиц – это процесс выполнения алгоритмов, которые идентифицируют человека по видеопотоку. Это можно представить следующим образом:

Обнаружение: Программа получает на вход видеопоток, где с помощью алгоритмов поиска на видеокамере определяется лицо человека.

После отсканированное лицо видоизменяется, происходит процесс изменения яркости, пикселей, масштабирования для приведения его к заданному шаблону, данное действие называется трансформацией объекта.

Далее система выполняет функцию сравнения изображения из камеры с изображениями в базе данных и находит максимальное сходство, происходит идентификация объекта.

Для идентификации личности приложение сравнивает идентифицированные лица с известными в базе и определяет лицо, это процесс верификации сотрудника.

Для выполнения данной задачи необходимо:

- изучить алгоритмы распознавания лиц;
- сравнить эффективность алгоритмов;
- определить основные инструментальные средства для разработки;
- спроектировать систему распознавания.

Реализовав выше поставленные цели будет разработана программа для распознавания лиц и контроля времени, которую можно будет использовать в различных предприятиях и организациях.

1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1. Пример существующих технологий распознавания лиц

Технология распознавания лиц вызывает большой интерес, на данный момент создано более 10 систем идентификации лица, которые предназначены для использования в разных сферах. Они работают по яркости изображений (рис. 1.1).

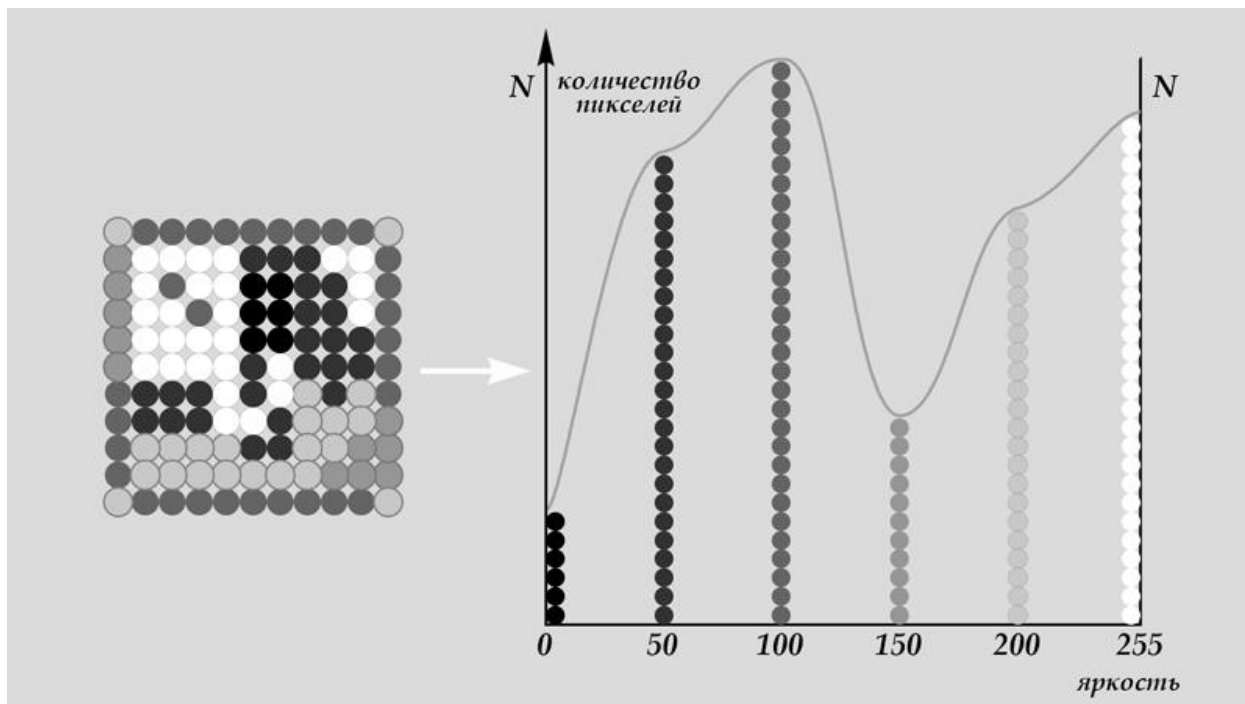


Рисунок 1.1 - Гистограмма яркости изображения

Из компаний, которые занимаются подобной разработкой стоит отметить следующие:

Амазон (Amazon) - Компания "Amazon" предоставляет продукт под названием «Amazon rekognition», он является сервисом анализа изображений и видео, который добавляет в свои приложения обширные возможности искусственного интеллекта, а в частности компьютерного зрения. Данный продукт реализован на технологии глубокого обучения и не требует конкретных познаний в машинном обучении, а также имеет легкий в использовании API, что позволяет пользователю моментально

проанализировать изображения и видеофрагменты, которые хранятся в хранилище.

Вы можете добавить в свое приложение функциональность с помощью Rekognition API для обнаружения объектов, текста и небезопасного контента, анализа изображений и видео и сравнения лиц. Используя API распознавания лиц Amazon Rekognition, вы можете обнаруживать, анализировать и сравнивать лица для различных случаев использования, включая аутентификацию пользователей, каталогизацию, подсчет людей и общественную безопасность.

Сервис основан на проверенной, хорошо масштабируемой технологии глубокого обучения, разработанной учеными Amazon в области компьютерного зрения, и может анализировать миллиарды изображений и видео каждый день. Rekognition регулярно изучает новые данные и часто добавляет в Сервис новые теги и функции [1].

Apple (США) – Face id – Эта технология распознавания лица. Ранее программа применялась в самых разных сферах. Например, для розыска преступников (рис. 1.2).

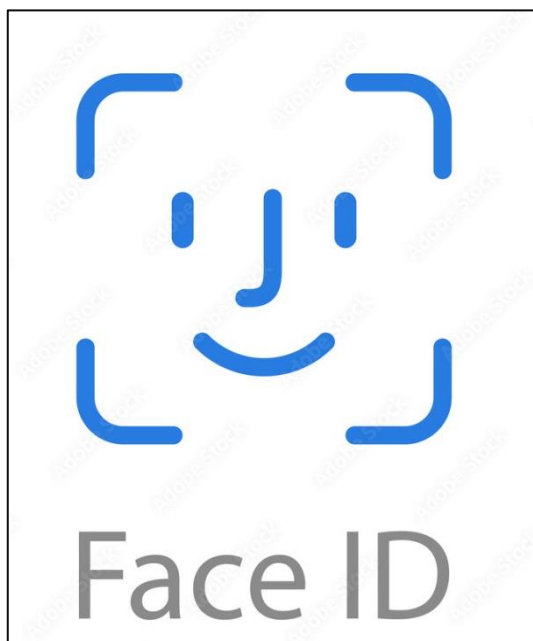


Рисунок 1.2 - Программа Face id

Компания задействовала данное изобретение в качестве универсального способа для защиты данных на устройствах пользователя. И уже после Apple технологию начали использовать и стали выпускать другие компании на примере Android.

Технология Face ID использует биометрическое сканирование лица. При помощи применения камера TrueDepth с инфракрасными проекторами. Вместе камеры и датчики образуют трехмерную модель лица пользователя. И созданный шаблон отправляется непосредственно в процессор.

Face ID — это метод разблокировки электроники и проведения финансовых транзакций, использующий распознавание лиц. Система противопоставляет внешний вид человека, использующего устройство в данный момент, изображению, которое хранится в его памяти.

Перечислим некоторые минусы:

- На блокировку уходит довольно много времени. Нужно включить дисплей, поднести девайс к лицу и ждать ответа. А еще нужно вспомнить о заминке, которая возникает перед тем, как входишь в систему.

- Отметим неудовлетворительную оптимизацию Face ID. Скажем, если сканирование плохое, то смартфон зависает. И понадобится некоторое время, чтобы войти повторно.

- Режим делает отклик на положение глаз. Чтобы с первого раза сбросить блокировку, нужно глядеть на экран и не отводить взгляд.

- Для стабильной работы необходимо, что пользователь располагался вертикально.

«LUNA SDK» — специализированный программный пакет, облегчающий распознавание и отслеживание лиц людей на цифровых изображениях или видеопотоках от корпорации VisionLabs (рис. 1.3).

Двигатель имеет превосходную точность в реальном времени, уступающую только лучшим. Процесс распознавания лиц основан на глубоких нейронных сетях и включает в себя несколько этапов:



Рисунок 1.3 - Программный пакет LUNA SDK

- Определения положения и размеров всех лиц (детекции); определения расположения характерных черт лица и трансформации его в стандартизированную форму (выравнивание);
- Извлечение дескрипторов (числовых векторов, которые суммируют характерные признаки лица); сравнения лиц с базой изображений и предотвращения подмены лиц.

Технические характеристики и особенности LUNA SDK:

- Разработана на языке программирования C++;
- Наблюдение и контроль в настоящее время;
- Использование алгоритма распознавания лиц на основе глубоких нейронных сетей;
- Оптимизированное управление памятью;
- Ёмкие дескрипторы, которые обеспечивают вычислительную эффективность;
- Поддержка разных платформ, включая Windows и Linux.

дескрипторы обучаются на наборах данных из различных источников и обеспечивают высокую точность в различных условиях, например, в банках, системах видеонаблюдения и социальных сетях.

1.2. Описание выбранных технологий и инструментов

Разработка приложения реализована на объектно-ориентированном языке программирования Python в среде разработки PyCharm.

PyCharm — это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик (рис. 1.4).



Рисунок 1.4 - Программа PyCharm

В качестве основного языка разработки использовался Python, так как имеет все необходимые инструменты для реализации, имеет лёгкий синтаксис и имеет хорошую производительность что облегчит реализацию.

Немаловажным критерием в пользу выбора данного языка программирования был опыт разработки на нём, полученный за время обучения.

Для разработки системы, также были использованы библиотеки OpenCV, dlib, Face Recognition. Библиотеки распространяются по лицензии BSD, а значит, может свободно использоваться в академических и коммерческих целях.

Python – это многофункциональный, интерпретируемый язык программирования, который имеет лёгкий синтаксис, поэтому подходит для решения любой задачи, а также подходит для разработки на любые платформы начиная с IOS и Android и заканчивая серверными операционными системами (рис. 1.5).



Рисунок 1.5 - Язык программирования Python

Программа на python – это текстовый файл. Код можно писать в любом текстовом редакторе или в специальной IDE:

- SublimeText;
- Spyder;
- Visual Studio Code;
- PyCharm.

Плюсы языка python:

Простой синтаксис: Синтаксис Python простой и понятный, что облегчает его использование даже для начинающего пользователя. Так же можно отметить хорошую безопасность памяти.

Интерпретируемый: Программа на Python представляет собой код, который можно запустить на любой платформе, где установлен интерпретатор.

Недостатки:

Скорость: Программы на Python являются одними из самых медленных, реализация PyPy по скорости близка к Java, однако в ней есть не все её возможности, для реализации задачи на Python требуется большой объём памяти.

Зависимость от библиотек: Это затрудняет переход на другие системы, для этого был разработан Virtualenv, но даже при этом есть избыток полных методов изоляции и дублирование системных библиотек.

Динамическая типизация: Из-за неё ошибки отображаются во время выполнения исходного кода, а не в компиляции.

Python используется для написания десктопных и мобильных приложений, веб-приложений, нейросетей, а так же игры и в веб-разработке, для этого существуют разные фреймворки:

- FastAPI;
- Flask;
- Tornado;
- CherryPy;
- Django.

Благодаря лаконичности большому набору инструментов и простому синтаксису Язык Python стал часто использоваться учёными для создания графиков, вычисления данных при помощи формул, а также для исследования.

Для этого существуют такие библиотеки как:

- SciPy;
- NumPy;
- Pandas;

- Matplotlib.

Учитывая, что язык не компилируется на нём возможно создавать десктопные программы и приложения.

Созданные программы на Python:

- Google;
- Netflix;
- Spotify;
- Uber.

1.3. Методы распознавания лиц

В данном разделе рассмотрим такие методы как:

- метод опорных векторов;
- метод К-Ближайших соседей;
- метод Виолы — Джонса;
- глубокое обучение;
- метод локального бинарного шаблона.

1. Метод опорных векторов — это метод построения гиперплоскости, оптимально разделяющей объекты выборки. Алгоритм работает следующим образом: чем больше расстояние между разделяющей гиперплоскостью и объектами разделяемых классов, тем меньше средняя ошибка классификатора, а объекты, попадающие на разделительную линию (отклонение от гиперплоскости), называются опорными (рис. 1.6) [2].

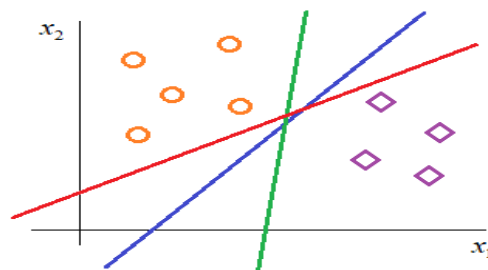


Рисунок 1.6 - Принцип работы метода опорных векторов на графике

2. Метод К-Ближайших соседей — этот метод используется для задач классификации и регрессии. Этот метод строит свои предположения основываясь на том насколько похожи тренировочные наблюдения (те на которых алгоритм был натренирован) и новые входящие наблюдения, то есть чем более похожи значения наблюдения, тем больше вероятность того, что они будут классифицированы одинаковой меткой [3].

Данные к которым применяется метод должны иметь:

- низкий уровень шума;
- данные должны быть размечены;
- отсутствуют нерелевантные признаки;
- данные должны быть разбиты на различимые подгруппы.

Метод не является отличным решением для больших наборов данных, так как будет требоваться большое количество временного ресурса (рис. 1.7).

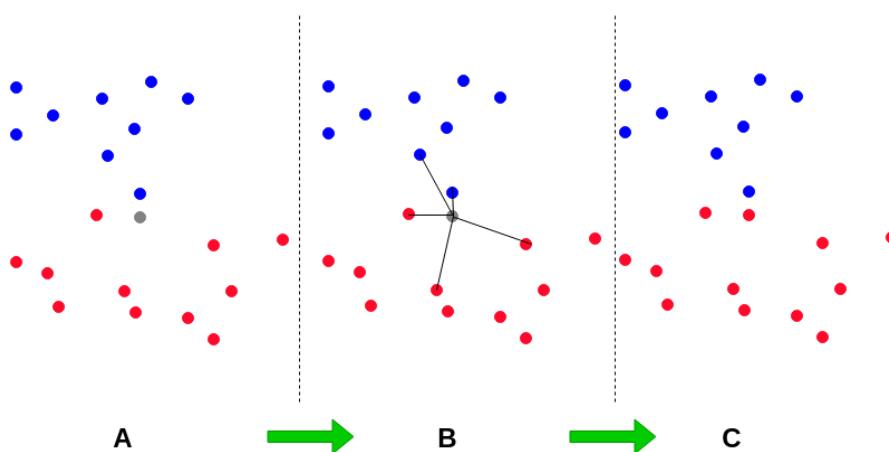


Рисунок 1.7 - Принцип метода «К-Ближайших соседей»

3. Метод Виолы — Джонса — этот алгоритм сканирует изображение используя примитивы Хаара, при помощи этого он находит более светлые, темные изображения, характерные конкретно для человеческих лиц.

4. Глубокое обучение — при глубоком обучении нейронных сетей применяется многоуровневая абстракция, в которой сложность образа увеличивается от уровня к уровню, каждый уровень описывает вид информации упорядочивает ее и передает в следующий уровень. Глубокое

обучение позволяет машинам использовать этот процесс, чтобы создать иерархическую структуру (рис. 1.8) [4].

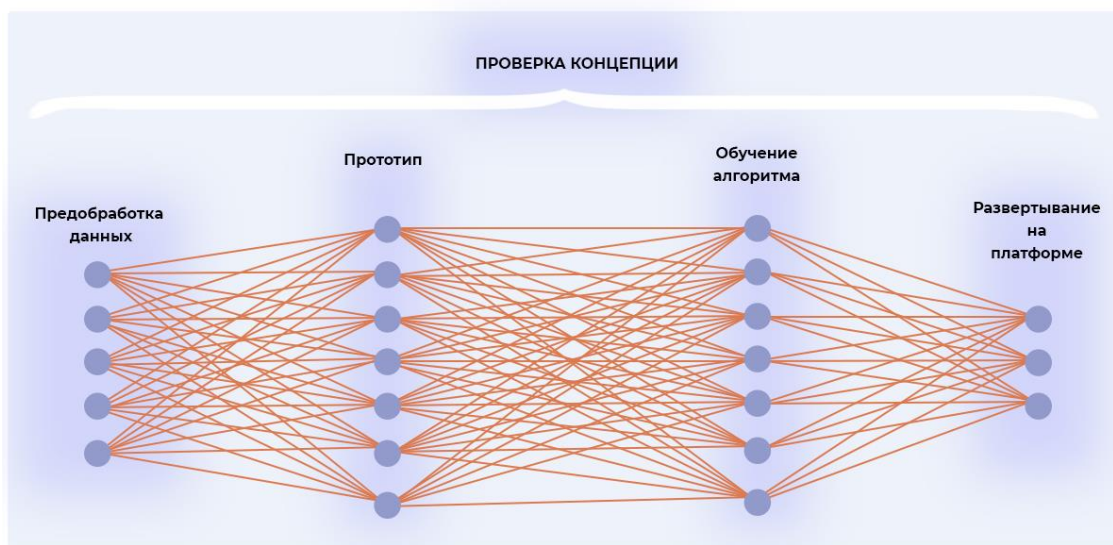


Рисунок 1.8 - Принцип метода глубокого обучения

1.3.1. Метод Виолы-Джонса

Метод Виолы-Джонса – этот метод с высокой скоростью обработки и высоким процентом сходства изображения является основным методов поиска объектов на видео и изображении. Основные принципы, на которых основана работа данного метода:

- интегральное представление;
- использование признаков Хаара;
- каскадная классификация;

Для повышенной эффективности работы с данными, в методе применяется метод интегрального представления, который быстро рассчитывает сумму яркости пикселей произвольного прямоугольника на заданном объекте. Такое представление изображения отображает матрицу одинаковую по размерам с исходным изображением, элемент которой хранит в себе сумму интенсивностей всех пикселей, находящихся над ним и слева от него плюс его собственный вес [5, 6].

В расширенном методе виолы-Джонса, который используется в библиотеке OpenCV и применяется в разрабатываемой системе, используются дополнительные признаки Хаара [7].

Каскады Хаара представляют собой прямоугольные области, которые составлены из нескольких соседних областей, отмеченных как светлая или темная, примеры таких признаков представлены на рисунке 1.9.

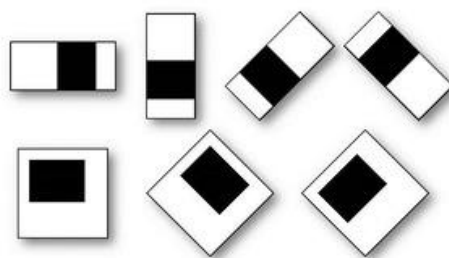


Рисунок 1.9 - Примитивы Хаара

1.3.2. Метод локального бинарного шаблона

Метод локальных бинарных шаблонов (Local Binary Patterns, LBP) — это широко используемый метод для анализа текстуры изображения. Он был предложен Тимо Ойала и Маатт в 1994 году для задач сегментации текстур, но позже его применение значительно расширилось, включая распознавание лиц и классификацию изображений [8, 9].

Данный метод широко используется следующих направлениях:

- обнаружение объектов;
- обнаружение лиц;
- распознавание лиц;
- текстурная классификация.

Данный метод использует обработку каждого окна размером 3X3 на изображении, чтобы преобразовать его в LBP код (рис. 1.10).

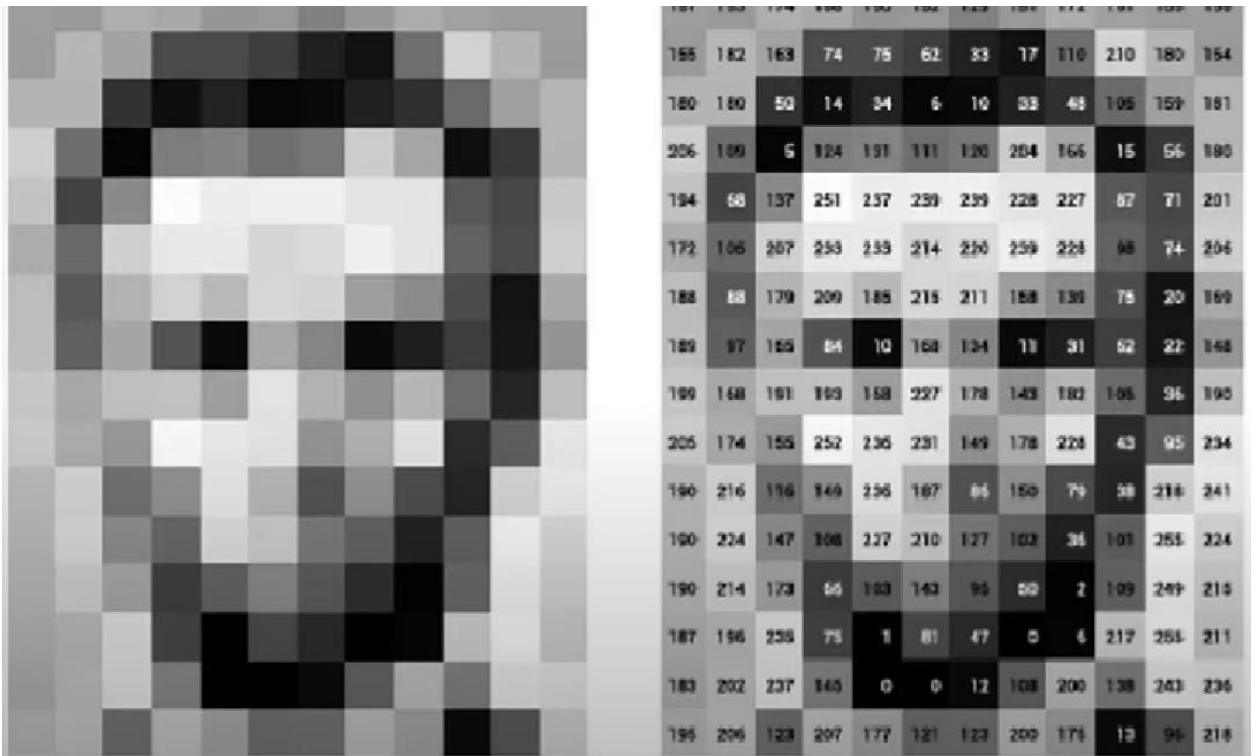


Рисунок 1.10 - Преобразование изображения в LBP код

LBP представляет собой метод, основанный на текстурных характеристиках, который описывает локальные текстурные особенности в изображениях (рис. 1.11).

Метод LBP является популярным подходом к извлечению текстурных признаков из изображений. Он основан на анализе локальных паттернов пикселей в изображении и позволяет выявить характеристики текстур, которые могут быть полезными для задачи распознавания эмоций [10, 11].

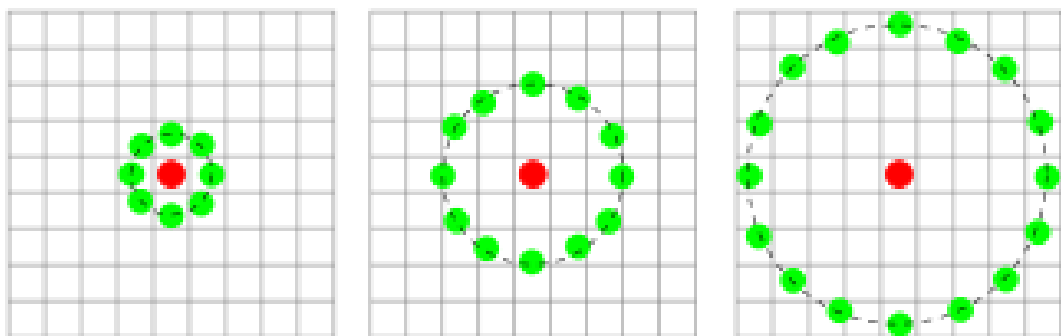


Рисунок 1.11 - Примеры определения текстуры и вычисления LBP

LBP-признаки извлекаются путем анализа значений пикселей и их соседей в заданной окрестности. Для каждого пикселя определяется локальный паттерн, который представляет собой бинарную последовательность, где каждый бит указывает на разницу между значением центрального пикселя и значениями его соседей. Затем эти локальные паттерны агрегируют для создания глобального дескриптора текстуры, который представляет собой гистограмму распределения локальных паттернов в изображении [12, 13]. Гистограмма отображает количество встречающихся локальных паттернов, что позволяет охарактеризовать текстурные свойства изображения. Таким образом, LBP-признаки предоставляют компактное представление текстурной информации, которое может быть использовано для задач классификации, распознавания или анализа текстурных данных.

Алгоритм LBP включает следующие шаги:

1. Задание размера окрестности вокруг каждого пикселя. Обычно используется окрестность размером 3x3 или 5x5 пикселей.
2. Для каждого пикселя в изображении:
 - Выбирается значения яркости пикселей в заданной окрестности.
 - Сравнивается значение центрального пикселя с каждым из его соседей.
 - Если яркость соседнего пикселя больше или равна яркости центрального пикселя, записывается 1, иначе записать 0.
 - Составляется бинарная последовательность из полученных значений для окрестности пикселя. Эта последовательность называется локальным бинарным шаблоном (LBP) для данного пикселя.
3. Строится гистограмма распределения локальных бинарных шаблонов по всем пикселям в изображении. Гистограмма представляет собой вектор, где каждый элемент соответствует количеству пикселей с определенным локальным бинарным шаблоном.

4. Полученную гистограмма используется в качестве дескриптора текстуры для классификации или распознавания объектов.

Вычисление шаблона LBP для пикселя представлено на рисунке 1.12.

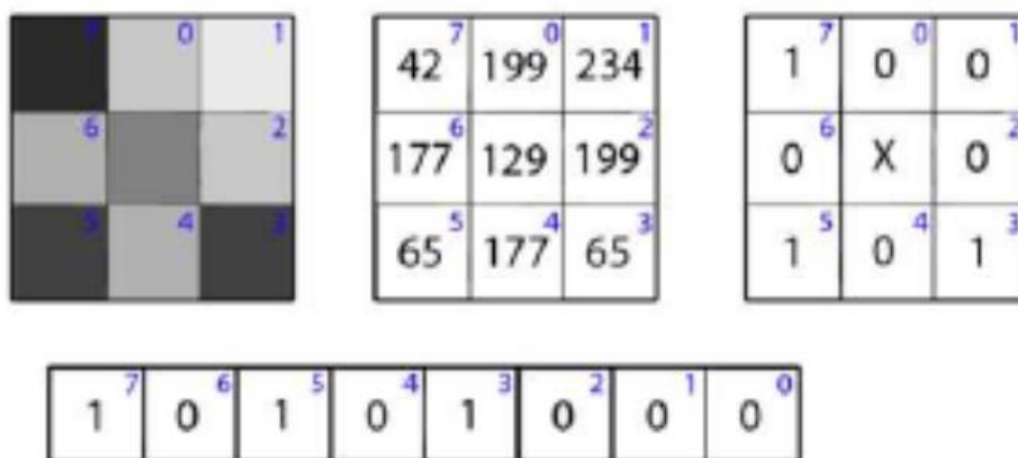


Рисунок 1.12 - Вычисление шаблона LBP для пикселя

В контексте задачи распознавания эмоций, LBP-признаки могут быть полезными для выявления текстурных особенностей, связанных с различными эмоциональными выражениями лица. Например, определенные текстурные шаблоны могут быть характерны для улыбки, грусти или злости. Использование LBP-признаков позволяет описать эти текстурные особенности и создать компактное представление, которое можно использовать для классификации изображений по эмоциональным категориям [14, 15].

Задача распознавания эмоций включает в себя определение эмоционального состояния человека на основе его лицевых выражений. Целью данной задачи является автоматическое классифицирование изображений лиц на соответствующие эмоциональные категории, такие как радость, грусть, злость, страх и т. д.

1.4. Общая характеристика предприятия ООО «Рубеж безопасности»

Название подразделения: отдел продаж ПО.

Фамилия, имя, отчество начальника: Пейль Александр Константинович, телефон: +7-932-321-56-30

Фамилия, имя, отчество заместителя начальника отдела: Пейль Андрей Константинович телефон заместителя начальника отдела +7(3452)615-630

Основные функции отдела продаж: Прием заказов; составление договора; организация оплаты услуги; контроль выполнения услуги; формирование отчета; составление выписки в отдел договоров.

Состав сотрудников отдела: Директор; менеджер продаж, сотрудник логистики, сотрудник документации.

Документы, поступающие в отдел продаж из других подразделений: Договор об оказании услуг – заказчик; договор о найме сотрудников и реализации услуги– субподрядчик; отчет о проделанной работе – субподрядчик.

Физическое представление, время и частота поступления, вид обработки и требования к безопасности каждого из этих документов: Табель учета рабочего времени – цифровой документ, ежемесячно, 1 числа текущего месяца, хранение в специальной папке, использование для предложений продажи покупателю.

Штатное расписание– цифровой документ, ежемесячно, 1 числа текущего месяца, хранение в специальной папке, использование для учета рабочего времени сотрудника

Продолжительность хранения каждого из документов и условия перевода в архив: номенклатура и текущие цены в течение трех лет, копии накладных – в течение года. Передача в архив по истечении срока хранения. Копии накладных в архив не передаются. Копии договоров – в течение года, в архив не передаются.

Документы, формирующиеся в отделе продаж: Журнал отчета реализованных услуг; счет на оплату; требования заказчика на выполнение услуги; отчеты продаж; правки на исполнение; выписки для закрытия договоров; отказ от предоставления услуг.

Физическое представление время, частота и вид обработки и требования к безопасности каждого из этих документов: Журнал отчета реализованных услуг представляет собой книгу с записями о предоставленных ранее услугах, договоров, копий, накладных; Счет на оплату – бумажный документ необходимый для заказчика; Требования заказчика на выполнение услуги, документ содержащий четкое задание выполнения работы;

Отчет продаж – папка с файлами, где хранятся все проведенные транзакции;

Правки на исполнение – документ, содержащий изменения в работе или дополнения к ней; Выписки для закрытия договора формируются ежемесячно, поквартально, ежегодно в виде бумажного документа; Отказ от предоставления услуг – это запись на бланке заказа или в договоре в том случае, если заказчик отказался от реализации услуги.

Продолжительность хранения каждого из документов и условия перевода в архив: Документы в архив не передаются.

Документы, передающиеся в другие подразделения: Правки на исполнения - передаются заказчику для ознакомления и согласию к реализации; Счет на оплату – бухгалтеру; отчеты – Генеральному директору

Физическое представление время и частота отправки, вид обработки и требования к безопасности каждого из этих документов: Счета на оплату выписываются после составления договора и утверждения даты;

Ежемесячные отчеты формируются вручную, представляют собой выписки из журналов, предназначаются руководителю и имеют гриф для служебного пользования.

Продолжительность хранения каждого из документов и условия перевода в архив: Документы в архив не передаются.

Информация, поступающая в отдел продаж из внешних к Рубеж безопасности источников (банк, заказчик, налоговые органы и т.д.): Запрос отчета о доходах – налоговая; Информация о счете – банк.

Физическое представление время и частота поступления, вид обработки и требования к безопасности каждого из видов информации: Запрос отчетов поступает в бумажном или цифровом виде; Информация счета поступает в цифровом виде при запросе директора.

Продолжительность хранения каждого из документов и условия перевода в архив: Данные документы хранятся в цифровом формате неограниченное количество времени.

Информация, передающаяся из отдела продаж во внешние органы: информация не поступает.

Описание информационной инфраструктуры (техническое оснащение и программные продукты) отдела продаж: 4 компьютера; 2 принтера, 1 сервер.

Основные недостатки бизнес-процессов и существующей системы обработки документов в отделе продаж: Сложность в введении учета рабочего времени в цифровом варианте вручную сотрудником

Предложения по реорганизации бизнес-процессов и существующей системы обработки документов в отделе продаж: Разработать приложение для учета рабочего времени на основе распознавания лиц

2. ПРОЕКТНЫЙ РАЗДЕЛ

2.1. Описание стадий процессов и их взаимосвязь

Первым этапом необходимо определить потребности предприятия, ожидаемый результат и главные функции программы. Провести анализ технических требований к системе, основными из которых являются:

- технические требования к системе;
- реализация внедрения на предприятии;
- безопасность и сохранность данных.

Так же, необходимо определить какие технологии будут использоваться при написании, кода и определить набор инструментов для их реализации.

Разработка интерфейса для сотрудника

Необходимо разработать простой и удобный в использовании интерфейс, который будет интуитивно понятен и не требующий большого количество ресурсов системы.

Разработать логику и взаимодействие программы с базой данных для обмена, сохранения и использования данных, а также создать функцию распознавания лиц и внедрить её в вывод видеопотока в интерфейс.

Тестирование программы:

Для корректного использования приложения необходимо провести тестирование системы и определить, как проходит процесс взаимодействия функций программы. Данная процедура предназначена для выявления и устранения ошибок.

Тестирование подтверждает правильность выполнения предыдущих этапов и обеспечивает качество конечного продукта. Обратная связь от тестирования может требовать доработок и корректировок на стадии разработки.

Внедрение и обучение:

- Развертывание системы: установка приложения на серверы компании, настройка баз данных и серверов хранения данных.
- Обучение пользователей: проведение тренингов и создания обучающих материалов для сотрудников по использованию новой системы.
- Обратная связь и доработка: сбор отзывов от пользователей, исправление выявленных недостатков и обеспечение поддержки на этапе перехода.

Эффективное внедрение и обучение непосредственно зависят от качества тестирования и разработки. Необходимо обеспечить, чтобы все функциональности и интерфейсы были понятны и доступны конечным пользователям.

Поддержка и развитие:

- Техническая поддержка: обеспечение бесперебойной работы системы, быстрое реагирование на возможные проблемы и инциденты.
- Обновления и улучшения: регулярное обновление и улучшение приложения на основе обратной связи от пользователей и новых технологических возможностей.
- Мониторинг и аналитика: постоянный анализ работы системы, сбор статистических данных и улучшение алгоритмов распознавания лиц.

Эта стадия является долгосрочной и непрерывной, обеспечивая жизненный цикл и актуальность системы. Удачная поддержка зависит от правильно выстроенного процесса внедрения и качественной архитектуры.

Каждая из описанных стадий является критически важной и тесно связана с предыдущими и последующими этапами. Хорошо организованный процесс разработки и внедрения приложения учета рабочего времени на основе распознавания лиц обеспечит высокую эффективность и удовлетворение потребностей компании.

2.2. Техническое задание на моделирование

Настоящее техническое задание направлено на разработку инновационного приложения для учета рабочего времени на основе технологии распознавания лиц, применимого в компании с численностью сотрудников 12 человек. Данное приложение призвано не только автоматизировать процесс регистрации времени прихода и ухода сотрудников, но и существенно упростить процедуру мониторинга рабочего времени, повысить уровень безопасности и доверия в коллективе, а также предоставить аналитические инструменты для управления персоналом.

1. Наименование разрабатываемого программного продукта:

"VisWork"

2. Назначение разрабатываемого программного продукта:

Разрабатываемое приложение предназначено для автоматизации процесса учета рабочего времени сотрудников организации с использованием технологии распознавания лиц.

3. Требования к программному продукту

3.1. Функциональные требования:

- регистрация и хранение данных о сотрудниках (ФИО, должность, отдел, личное фото);
- возможность ведения учета рабочего времени (фиксация времени прихода, ухода);
- распознавание лиц сотрудников при входе и выходе с работы;
- формирование отчетности по рабочему времени (ежедневные, еженедельные, ежемесячные отчеты);
- административная панель управления (добавление, изменение, удаление данных о сотрудниках).

3.2. Требования к надежности:

- защита данных от несанкционированного доступа;

- резервное копирование данных;
- восстановление данных после аварийных ситуаций.

3.3. Требования к интерфейсу:

- интуитивно понятный интерфейс для пользователей всех уровней подготовки;
- адаптивность интерфейса под различные устройства (ПК, планшеты, смартфоны).

3.4. Технические требования:

- компьютер или сервер с установленной ОС Windows или Linux;
- веб-камера с разрешением не ниже 720p;
- доступ в интернет для синхронизации данных.

4. Требования к документации:

- руководство пользователя;
- инструкция по установке и настройке;
- руководство системного администратора.

5. Стадии и этапы разработки:

- техническое задание;
- проектирование программного продукта;
- кодирование;
- тестирование;
- внедрение;

6. Требования к квалификации и численности исполнителей:

- руководитель проекта - 1 человек;
- аналитик - 1 человек;
- проектировщик интерфейса - 1 человек;
- программисты - 1 человек;
- тестировщики - 1 человек;
- системный администратор - 1 человек.

7. Источники и порядок финансирования

Финансирование проекта осуществляется за счет средств заказчика.

8. Сроки выполнения проекта

Проект должен быть выполнен в течение 3 месяца с момента утверждения технического задания.

Система на базе распознавания лиц позволит исключить возможность неправомерного использования учетных данных, свести к минимуму человеческие ошибки и упростить процесс учета рабочих часов, что особенно актуально в условиях малых и средних предприятий. Будучи высокотехнологичным решением, данное приложение создаст устойчивую основу для дальнейшего роста и совершенствования корпоративных процессов.

2.3. Проектирование системы

Для реализации проекта были выбраны следующие технологии:

- язык программирования: Python;
- СУБД: SQLite;
- среда разработки: PyCharm.

Python является высокоуровневым интерпретируемым языком программирования, широко признанным благодаря своей простоте и читабельности синтаксиса. Спектр его применения крайне широк, включая области веб-разработки, анализа данных, машинного обучения и автоматизации процессов.

Python обладает обширной стандартной библиотекой и поддерживается широким сообществом разработчиков, что способствует быстрому развитию и доступности множества дополнительных инструментов.

SQLite — это встраиваемая, легковесная система управления базами данных, не требующая отдельного серверного процесса и предоставляющая

все необходимые возможности для локального хранения данных. SQLite особенно подходит для разработок, где не требуется обеспечение высоких нагрузок и масштабирования. Её интеграция с Python осуществляется через модуль `sqlite3`, что упрощает взаимодействие между приложением и базой данных (рис. 2.1).



Рисунок 2.1 - СУБД SQLite

PyCharm — интегрированная среда разработки (IDE) от компании JetBrains, созданная специально для языка программирования Python. Она предоставляет мощный набор инструментов, включая автодополнение кода, средства рефакторинга, встроенный отладчик, тестирование, поддержку работы с системами контроля версий (например, GIT), а также обширные возможности настройки интерфейса пользователя. PyCharm поддерживает интеграцию с различными фреймворками и библиотеками, что делает её идеальным выбором для комплексных проектов на Python.

2.4. Разработка архитектуры системы

Программа должна иметь несколько частей ее использования:

- Клиентская часть;

- Управление и администрирование;
- Отчетность.

В клиентской части должно быть реализована возможность просмотра отработанного времени, а также взаимодействие с данными внесенными при устройстве на работу (ФИО, возраст и тд.)

Администрирование системы предоставляет возможность управление приложением, возможностью добавления и удаления пользователя из системы, сохранение фото сетов пользователя, вывод основной камеры для идентификации сотрудников (рис. 2.2).

Добавление сотрудников будет осуществляется следующим образом:

- Сотрудник предоставляет свои данные
- Администратор вводит заданные данные, генерирует пароль и выдает роль сотрудника.
- Полученные данные записываются и сохраняются в базе данных, пользователю присваивается универсальный id.
- Администратор делает несколько снимков для создания датасета, по которому в дальнейшем будет идентификация сотрудника.
- Полученный датасет сохраняется в ячейке с присвоенным ранее id сотрудника.

Таким образом будет реализована база банных с содержанием в ней всех работников предприятия.

Отчетная часть должна выполнять функцию сбора и анализа посещаемости сотрудников, а также реализовывать подсчет общего отработанного времени сотрудника на предприятии.

Сбором и анализом данных должен заниматься отведенный сотрудник с ролью «Бухгалтер». Сбор информации будет реализован следующим образом:

- Сотрудник заходит в свой аккаунт
- Наживает кнопку «Отчет»

Из базы данных будет экспортироваться вся информация по приходам и уходам сотрудников, и экспортироваться в excel формат.

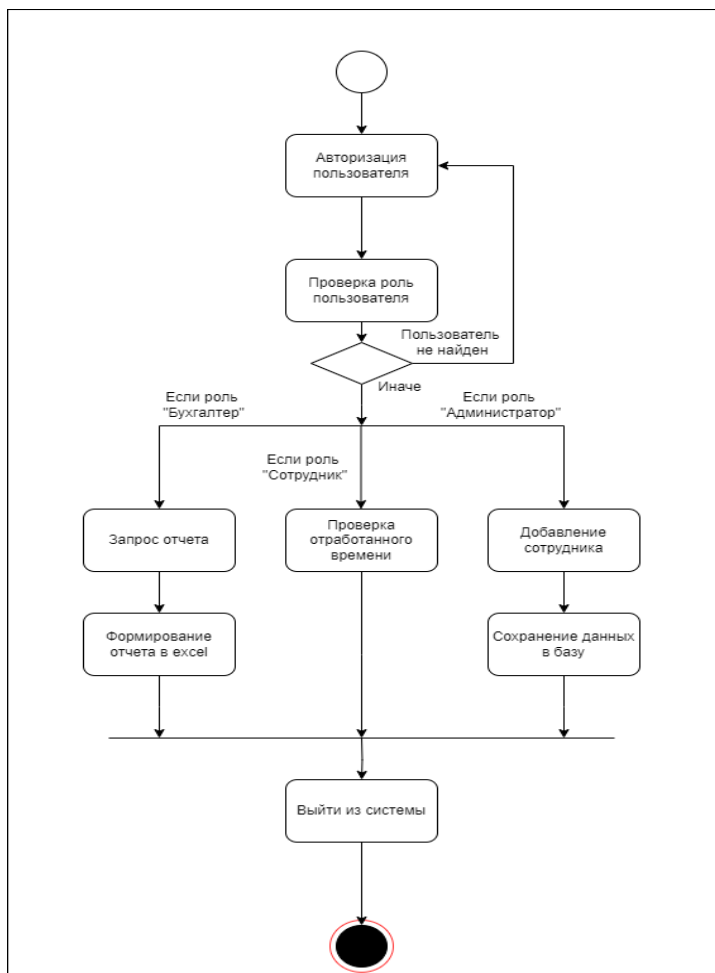


Рисунок 2.2 - Диаграмма деятельности архитектуры системы

2.5. Проектирование базы данных

База данных, которая будет реализована в проекте, имеет следующую структуру:

Создается таблица «Пользователей», в которой будут храниться данные необходимые для процесса регистрации и входа сотрудника (рис. 2.3). В таблице представлено несколько ячеек

Ячейка ID - предназначена для присвоения универсального идентификатора для каждого пользователя

ФИО - служит для сохранения фамилии имя отчества сотрудника, а также будет использоваться как логин для входа в систему.

Ячейка пароль – в ней пользователь задает свой пароль для входа в систему.

Роль – необходима для обозначения пользователя, а также чтобы программа могла определять какими правами может пользоваться сотрудник. Изначально в программе будет заложено три роли «Администратор», «Бухгалтер», «Сотрудник»

Фото – ячейка имеет тип данных BLOB, в ней будет храниться созданный датасет фотографий конкретного ID. Хранящийся датасет будет принимать двоичный вид, что позволит наиболее комфортно обрабатывать данные, которые хранятся в ячейке.

После создания таблицы «Пользователи» создается таблица «Контроль времени», она предназначена для хранения информации о приходе и уходе сотрудника с работы. В ней имеются следующие ячейки:

Дата – при фиксации лица будет сохраняться дата, благодаря чему появится возможность рассчитывать отработанное время за разные промежутки.

Время прихода – ячейка для записи прихода на работу

Время ухода – ячейка для записи ухода с работы

Фото подтверждение – данная ячейка необходима для контроля достоверности информации, так если при сбое программы определится другой человек, то при помощи фото подтверждения администратор сможет проверить тот ли человек идентифицировался в определенную дату.

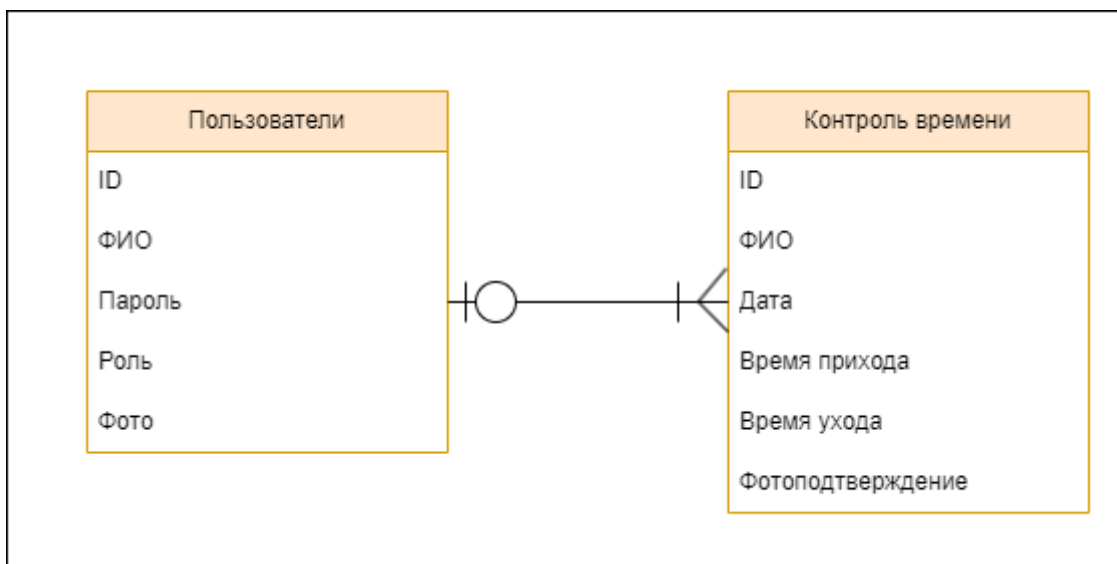


Рисунок 2.3 - ER-диаграмма базы данных

Проектирование пользовательского интерфейса (UI)

Разработка пользовательского интерфейса является важной частью проекта, поскольку работа в консоли вызывает трудности у большинства пользователей, было принято решение разработать простой интерфейс, который будет удобен и интуитивно понятен в использовании

Первым делом необходимо реализовать интерфейс входа в систему, в ней должны быть предусмотрены:

- ввод логина;
- ввод пароля;
- кнопка входа.

Предоставленные данные внесут понимание пользователю, что необходимо для использования приложения

Следующим шагом является отображение рабочей зоны пользователя с добавленными функциями для реализации работы.

Для примера подберем вид рабочей зоны для роли «Администратор», в ней должны быть встроены функции «Добавления пользователя», «Удаления пользователя», «Создание датасета», вывод изображения с камеры для начала

сканирования сотрудников и выход из аккаунта, учитывая все необходимые элементы управления был разработан прототип интерфейса администратора.

2.6. Проектирование модуля распознавания лиц

В данном разделе будут прописаны основные методы проектирования.

Обнаружение лиц:

Модуль должен обнаруживать лица на входящих кадрах из видеопотока.

Подготовка данных для обучения:

Для обучения модели необходимо извлечь данные об изображениях лиц и их соответствующих идентификаторах из базы данных.

Обучение модели распознавания:

Для обучения модели использовался метод локальных бинарных шаблонов, собранный датасет должен пройти функцию сравнения и успешно идентифицировать сотрудника. Далее идет визуализация данных, где функция выводит информацию о сотруднике (ФИО, должность) и делать фотоотчет, который сохраняется в файлах программы

2.7. Разработка основной функции

Для создания функции были использованы необходимые библиотеки для работы с изображением и видеопотоком.

Создаем класс «VideoWindow» для написания основного кода распознавания (рис. 2.4).

Был создан код функцией которого является получение и обработка данных пользователя из базы данных:

```

def fetchTrainingData(self):
    conn = sqlite3.connect("users.db")
    cursor = conn.execute("SELECT id, photo, username FROM users")
    faces = []
    ids = []
    names = []
    for row in cursor:
        user_id = row[0]
        photo_data = row[1]
        user_name = row[2]
        if photo_data is not None:
            try:
                # Десериализация данных из формата pickle
                dataset = pickle.loads(photo_data)
            except pickle.UnpicklingError:
                print(f"Error unpickling data for user_id {user_id}")
                continue # Пропускаем и переходим к следующей записи

            for face_image in dataset:
                if isinstance(face_image, bytes):
                    face_image = np.frombuffer(face_image, np.uint8)
                    face_image = cv2.imdecode(face_image, cv2.IMREAD_GRAYSCALE)
                elif face_image.ndim == 3:
                    face_image = cv2.cvtColor(face_image, cv2.COLOR_BGR2GRAY)
                faces.append(face_image)
                ids.append(user_id)
                names.append(user_name)

    conn.close()
    return faces, ids, names

```

Рисунок 2.4 - Код распознавания

Этот метод извлекает датасеты из базы, а также id сотрудника, ФИО, должность. Эти данные используются для обучения модели [16, 17].

```

def normalize_image(self, image):
    if image.ndim == 3:
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    return cv2.equalizeHist(image)

```

Рисунок 2.5 - Код изменения изображения

Код показанный на рисунке 2.5 изменяет полученный датасет, делая его серым и использует выравнивание для достижения лучшего качества изображения перед обработкой.

В данной работе рассматривается разработка системы распознавания лиц с использованием библиотек Python, таких как OpenCV, tkinter, и PIL. Представленный код включает класс VideoWindow, который обеспечивает непрерывное обновление изображений с камеры, детекцию лиц, распознавание лиц и отображение информации о них. Используется база данных SQLite для хранения информации о пользователях и их обучающих изображениях [18].

В работе изучается и реализуется алгоритм обработки изображений с использованием библиотек Python для машинного зрения. Обучение модели распознавания лиц происходит на основе предварительно подготовленных изображений из базы данных. Для обеспечения реального времени используются методы из библиотек OpenCV и Tkinter. Результаты распознавания отображаются на экране пользователя в реальном времени вместе с дополнительной информацией о текущем времени и профиле пользователя [19].

Предложенная система успешно определяет и распознает лица на видеопотоке, отображая информацию о пользователях, если они были ранее идентифицированы. Работоспособность системы подтверждается путем экспериментов и тестирования на различных данных. При успешном распознавании система также сохраняет скриншоты и выводит временные метки, что позволяет вести журнал событий (рис. 2.6).

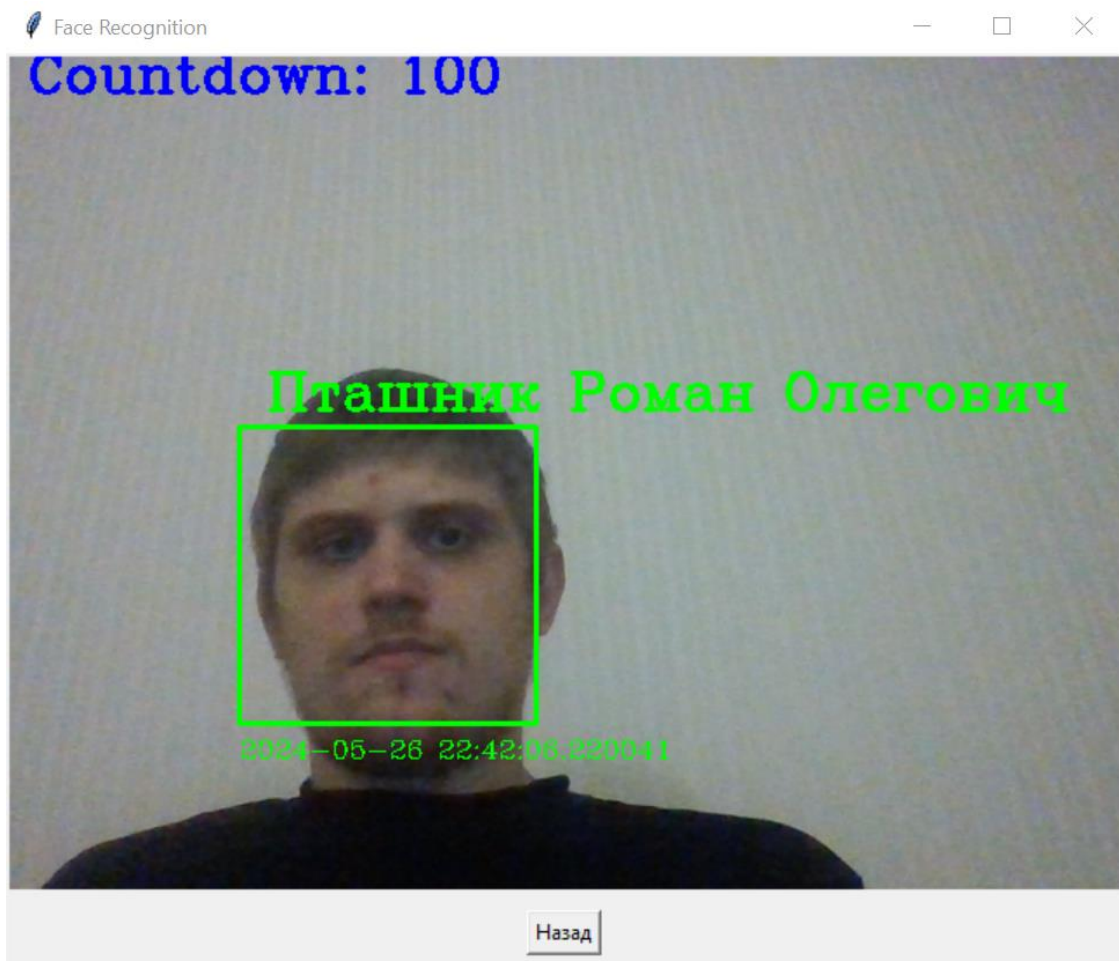


Рисунок 2.6 - Пример идентификации сотрудника

Созданный программный код является эффективным решением для определения лица сотрудника из видеопотока в настоящее время, главным преимуществом разработанной системы является простота в использовании и низкие требования системы, что является главным плюсом для внедрения её на предприятие.

2.8. Пользовательский интерфейс

Пользовательский интерфейс был реализован следующим образом:

Было разработано окно для авторизации пользователя, где необходимо ввести логин и пароль сотрудника (рис. 2.7).

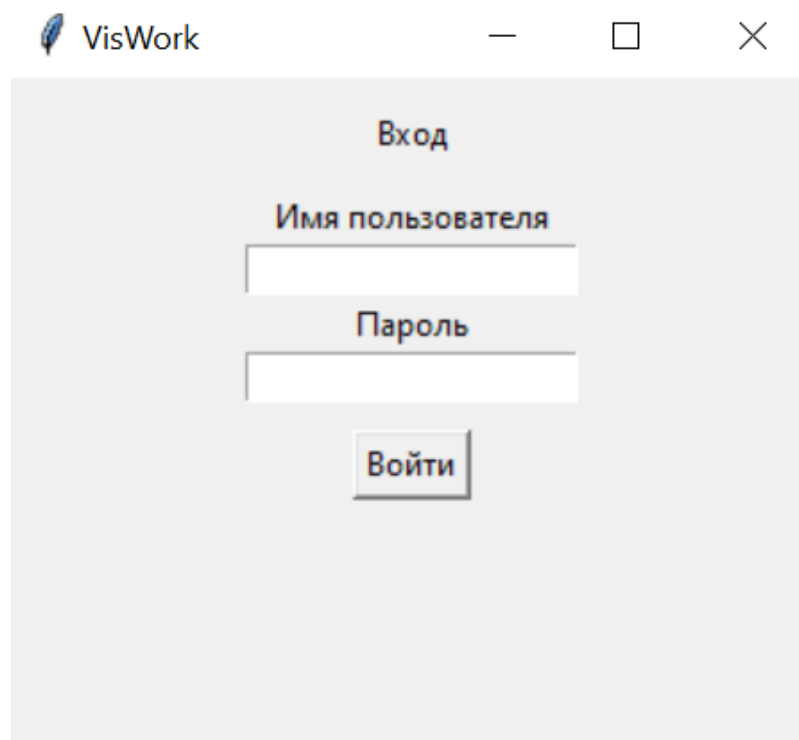


Рисунок 2.7 - Окно авторизации

После авторизации пользователя перекидывает на рабочую станцию программы, показанную на рисунке 2.8, где выполняются все основные действия:

- Добавление пользователя – функция позволяет занести данные нового сотрудника в базу для дальнейшего взаимодействия.
- Удаление пользователя – отвечает за удаление данных из базы определённого сотрудника;
- Добавление фото – необходима для создания датасета сотрудника, по которому будет происходить идентификация с камеры;
- Вывод камеры – вывод видеопотока для начала рабочего дня;
- Выход – выход из аккаунта.

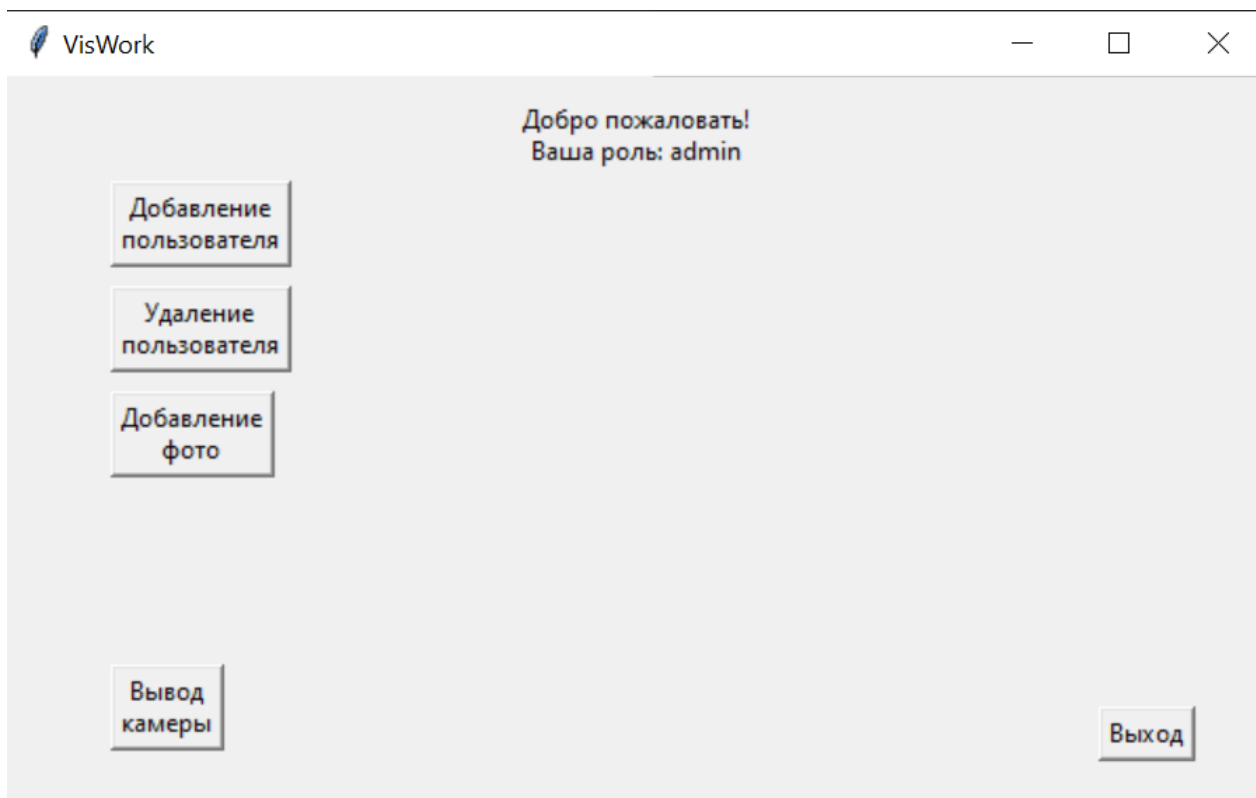


Рисунок 2.8 - Интерфейс рабочей зоны администратора

Для отчётности о отработанном времени был создан отдельный интерфейс, он включает в себя генерацию отчётов за неделю, месяц, год (рис. 2.27).

2.9. Процесс тестирование системы

Тестирование разработанной программы «VisWork» проводилось в несколько этапов:

Тестирование интерфейса: Первым этапом тестирования была проверка на ввод данных сотрудника и обработка введенных данных в базе. При тестировании программа показал отличный результат с определением данных сотрудника и предоставлением допуска к рабочему интерфейсу с учетом роли.

Функции рабочего интерфейса также дали положительный результат, функция добавления пользователя внесла все введенные данные в базу, а

кнопка «Сделать фото» создала датасет и сохранила его в базу под id конкретного сотрудника.

При создании отчёта о рабочем времени был создан файл excel в котором содержалась информация о сотруднике и отработанное время за месяц.

Тестирование системы на нагрузку системы: было проведено тестирование программы на работоспособность при высокой нагрузке системы. Для тестирования программы использовалась компьютерная программа «Furmark» показанная на рисунке 2.9.

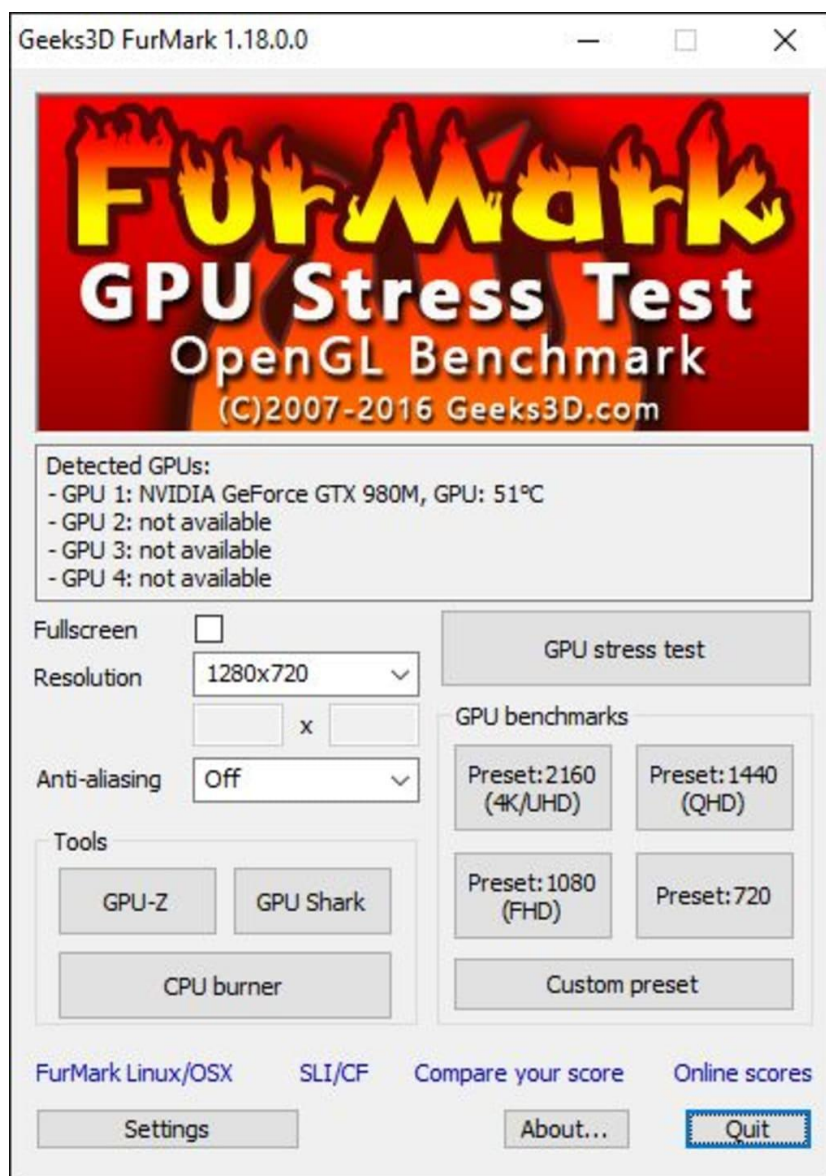


Рисунок 2.9 - Интерфейс программы «Furmark»

Система продемонстрировала высокую скорость работы при обработке видеопотоков и распознавании лиц даже при больших нагрузках, показывая стабильность производительности.

Реакция системы на различные ошибочные ситуации была проверена, и система успешно справилась с потерей видеосигнала, проблемами с базой данных и другими непредвиденными сценариями.

Система успешно справилась с различными условиями освещения, углами обзора и другими факторами, влияющими на точность распознавания лиц.

Проведенное тестирование подтвердило высокое качество и надежность работы системы распознавания лиц "VisWork". Система успешно прошла все этапы тестирования, что гарантирует ее стабильную и эффективную работу в реальных условиях эксплуатации. Реализация всех предложенных тестовых сценариев помогла выявить и устранить потенциальные проблемы на ранних этапах разработки, обеспечив надежную и качественную работу системы [20].

2.10. Обзор работоспособности функции определения

Для проверки работоспособности функции определения лиц было создано несколько датасетов лиц разных людей, после внесения их в базу функция определения дала следующие результаты:

Для начала были внесены данные первого пользователя в программу, как показано на рисунке 2.10.

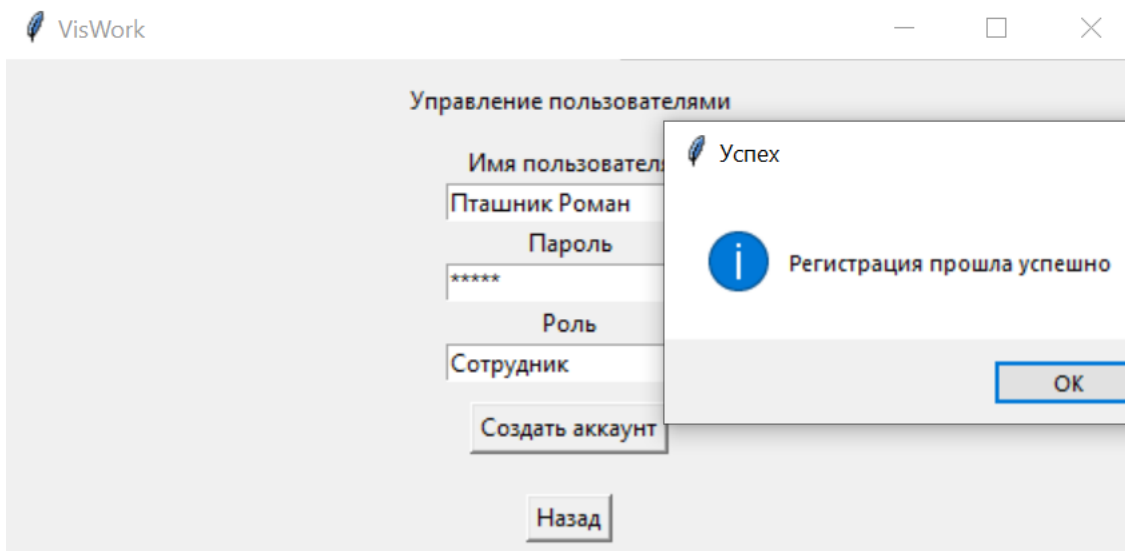


Рисунок 2.10 - Данные первого пользователя

После создания был создан датасет из тридцати фотографий лица пользователя.

Для идентификации отрывается окно «Вывод камеры», где происходит определение и идентификация, как показано на рисунке 2.11.

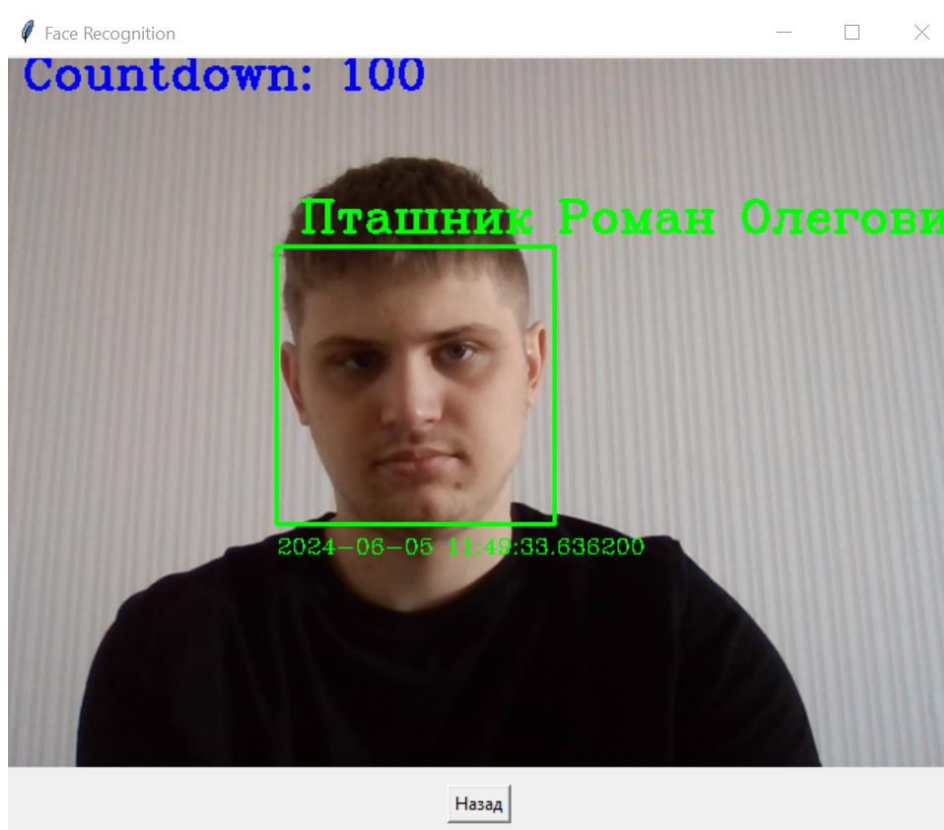
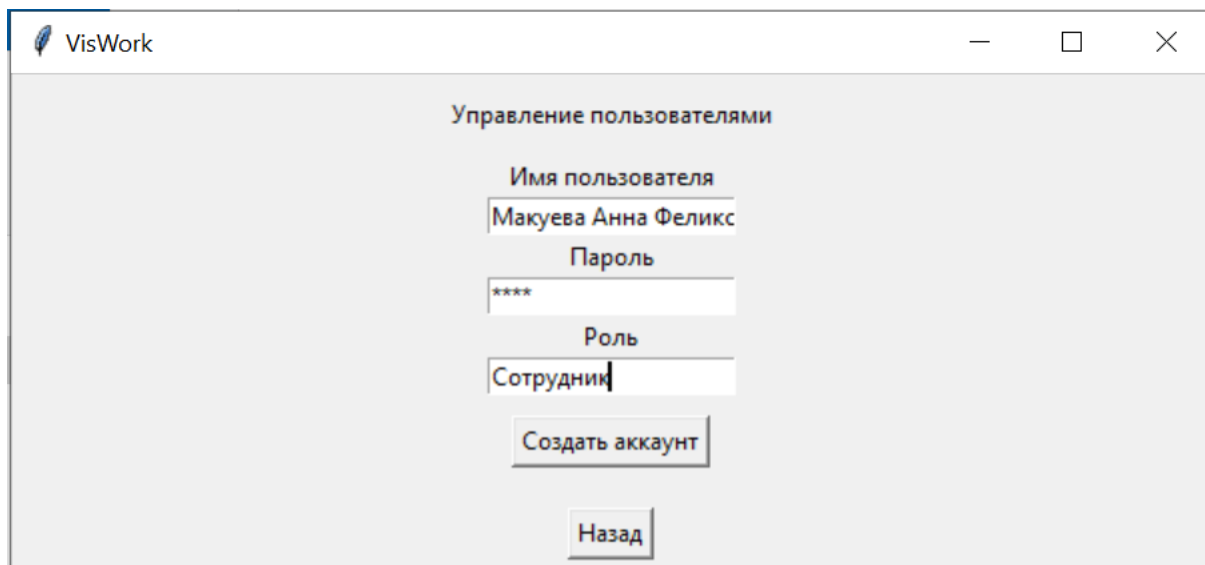


Рисунок 2.11 - Определение лица первого пользователя

Для проверки на точность определения было решено внести больше пользователей:



VisWork

Управление пользователями

Имя пользователя
Макуева Анна Феликс

Пароль

Роль
Сотрудник

Создать аккаунт

Назад

Рисунок 2.12 - Данные второго пользователя

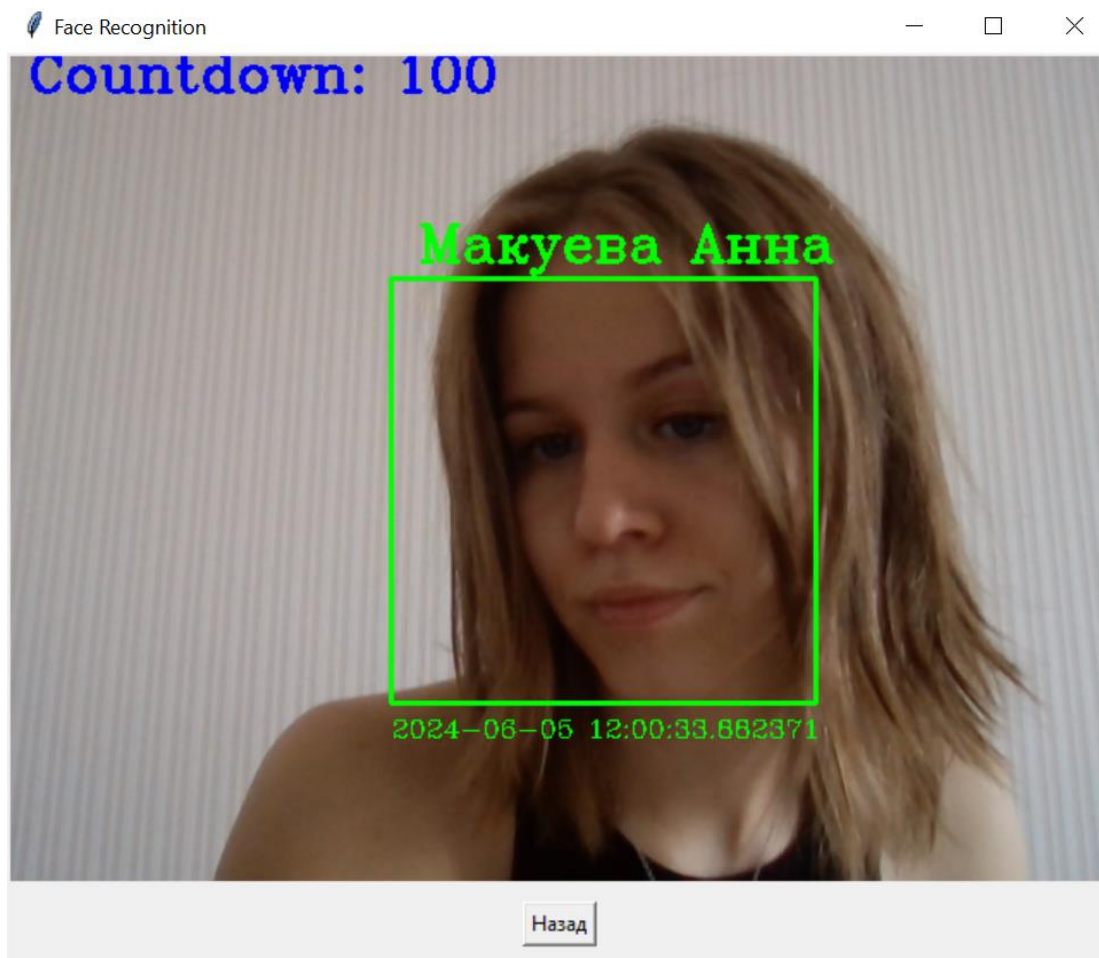


Рисунок 2.13 - Определение лица второго пользователя

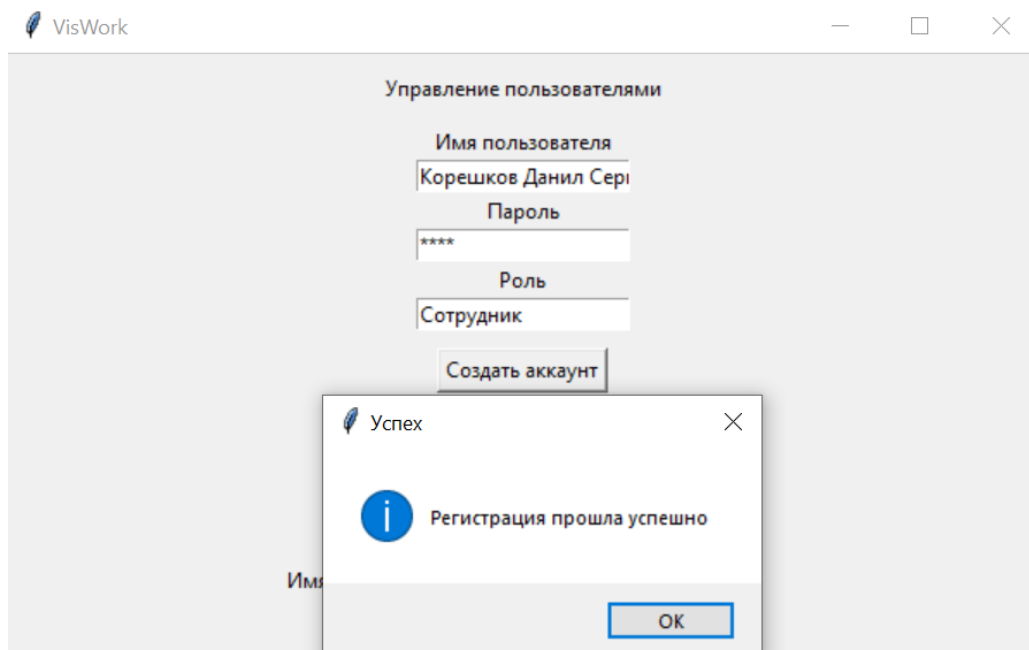


Рисунок 2.14 - Данные третьего пользователя

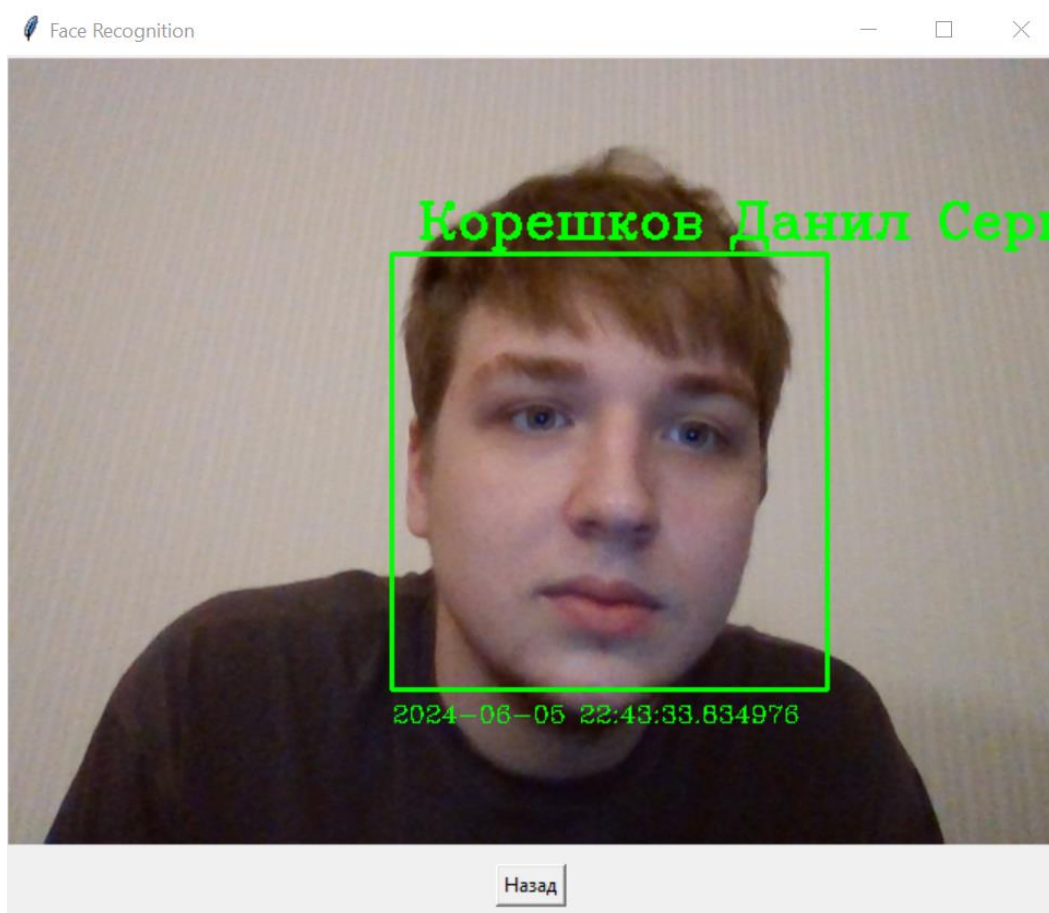


Рисунок 2.15 - Определение лица третьего пользователя

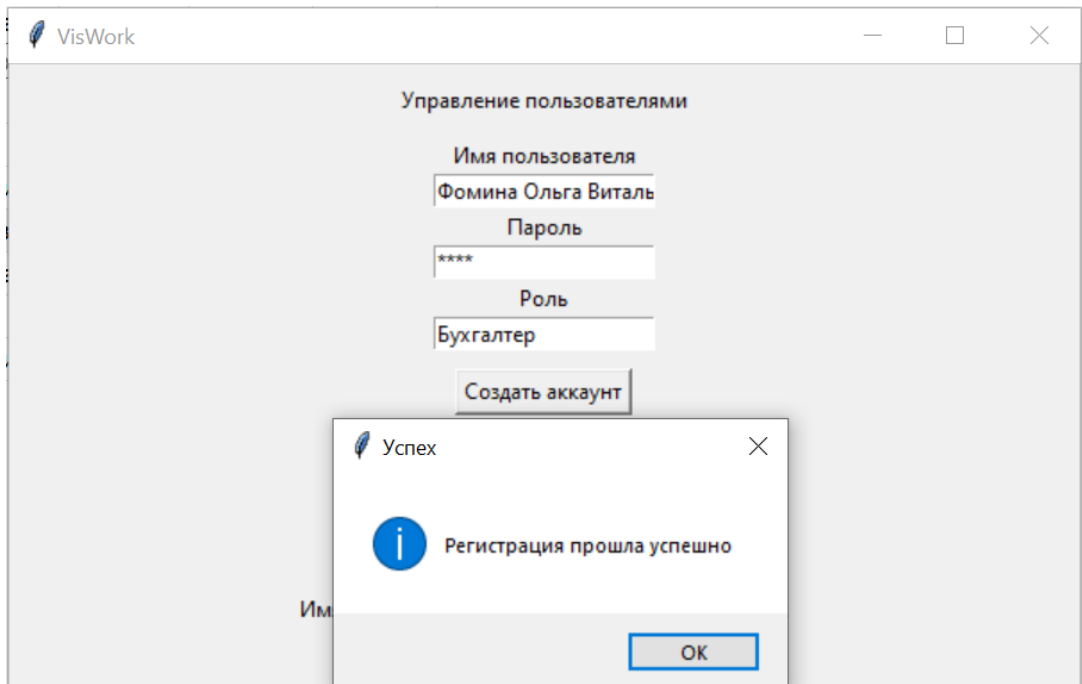


Рисунок 2.16 - Данные четвертого пользователя

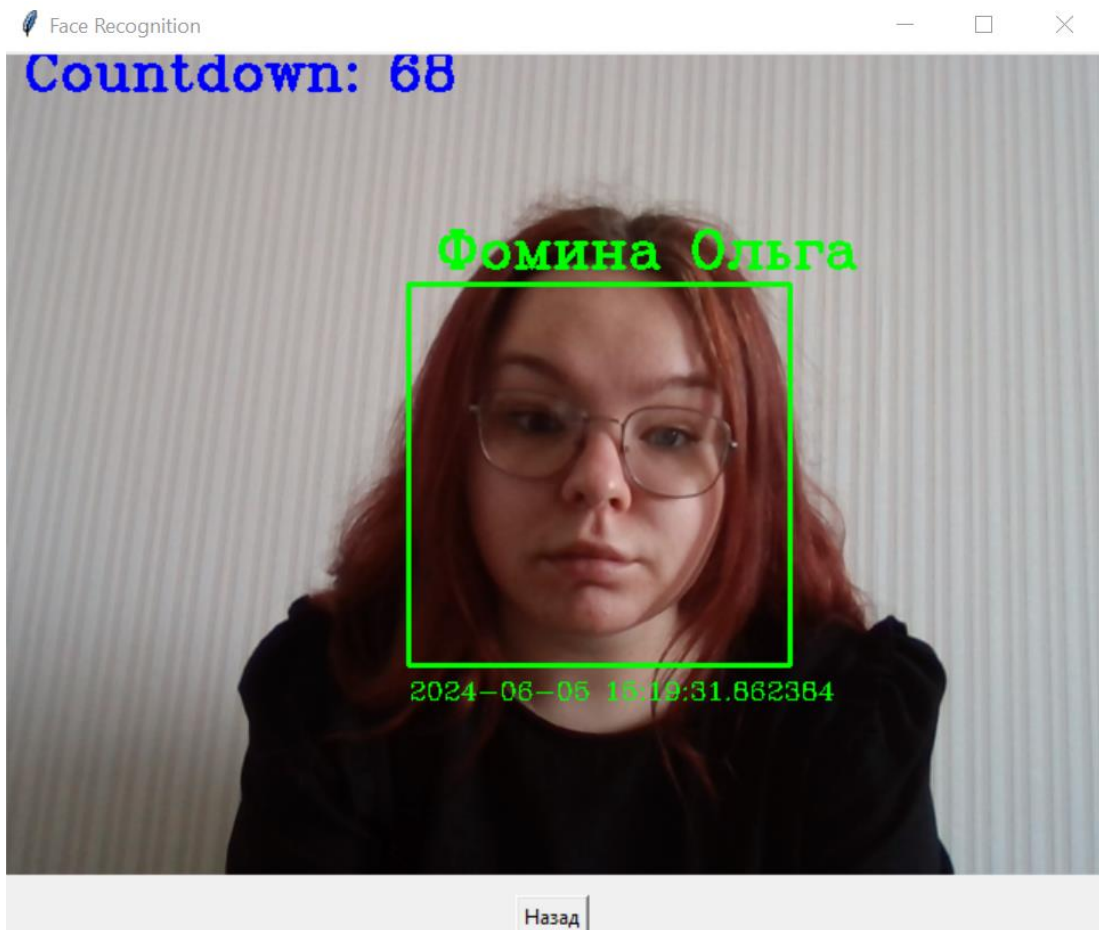


Рисунок 2.17 - Определение лица четвертого пользователя

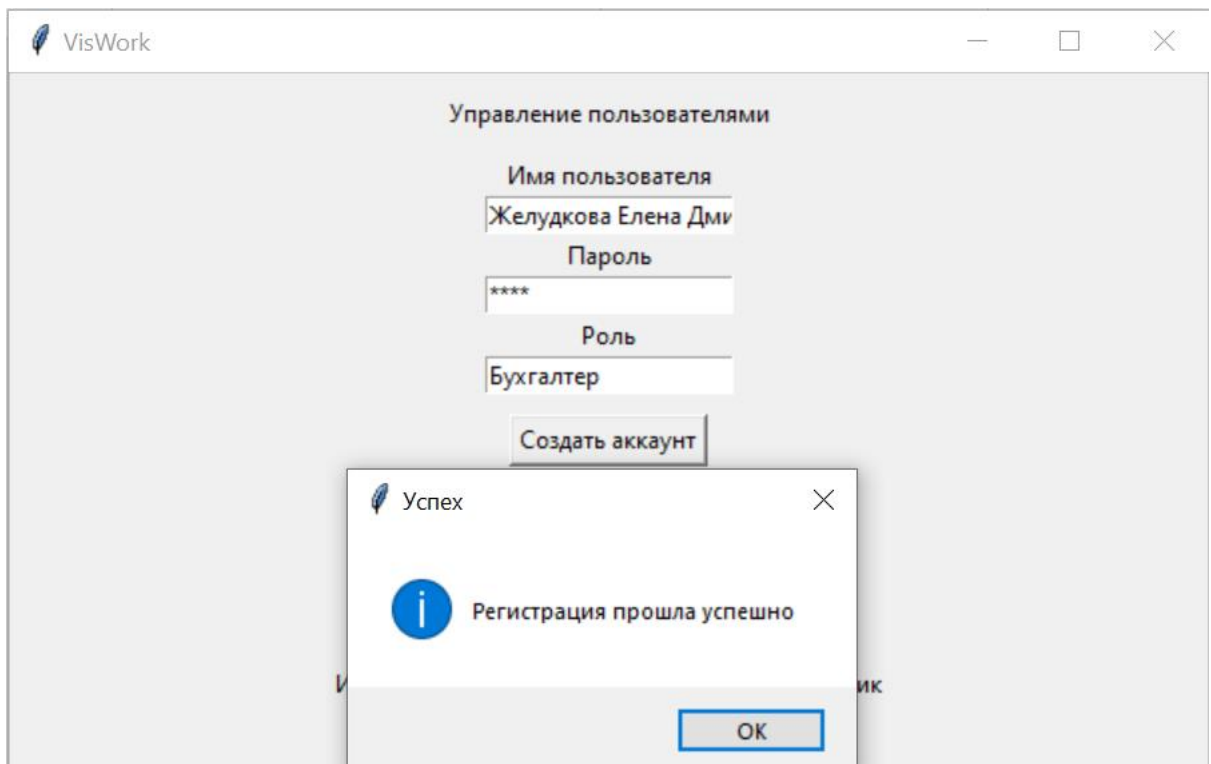


Рисунок 2.18 - Данные пятого пользователя

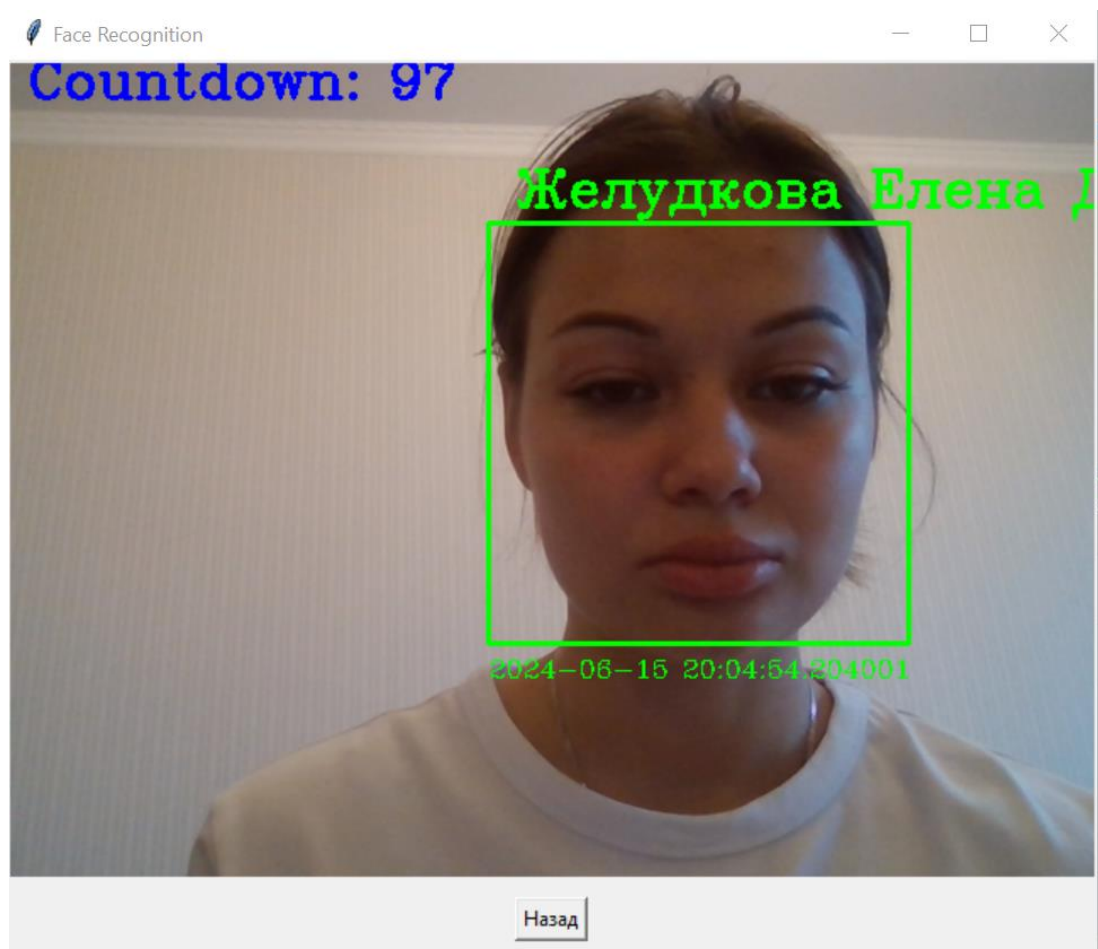


Рисунок 2.19 - Определение лица пятого пользователя

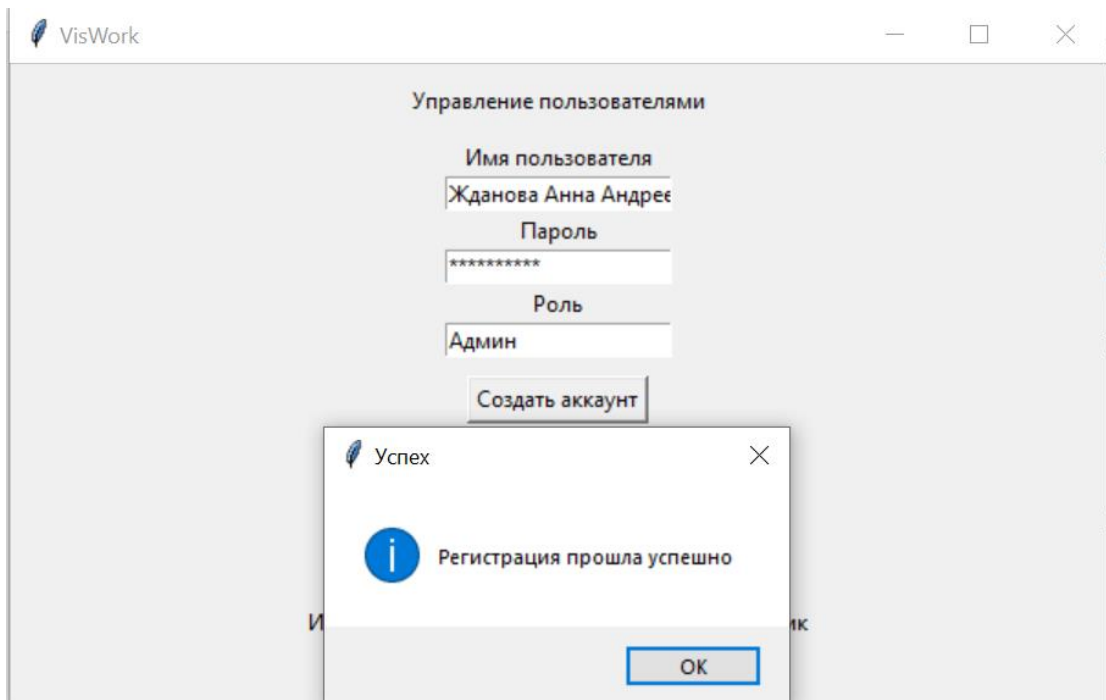


Рисунок 2.20 - Данные шестого пользователя

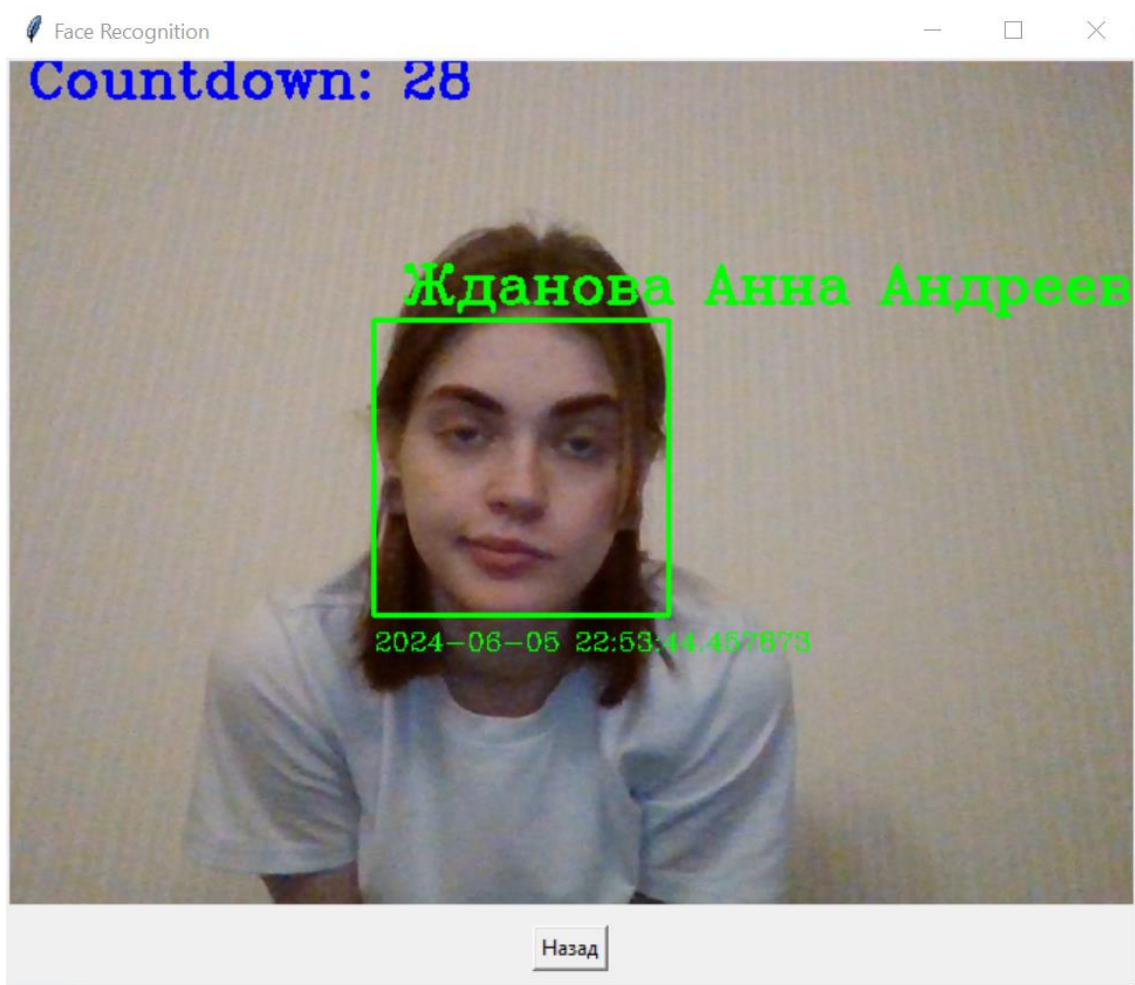


Рисунок 2.21 - Определение лица шестого пользователя

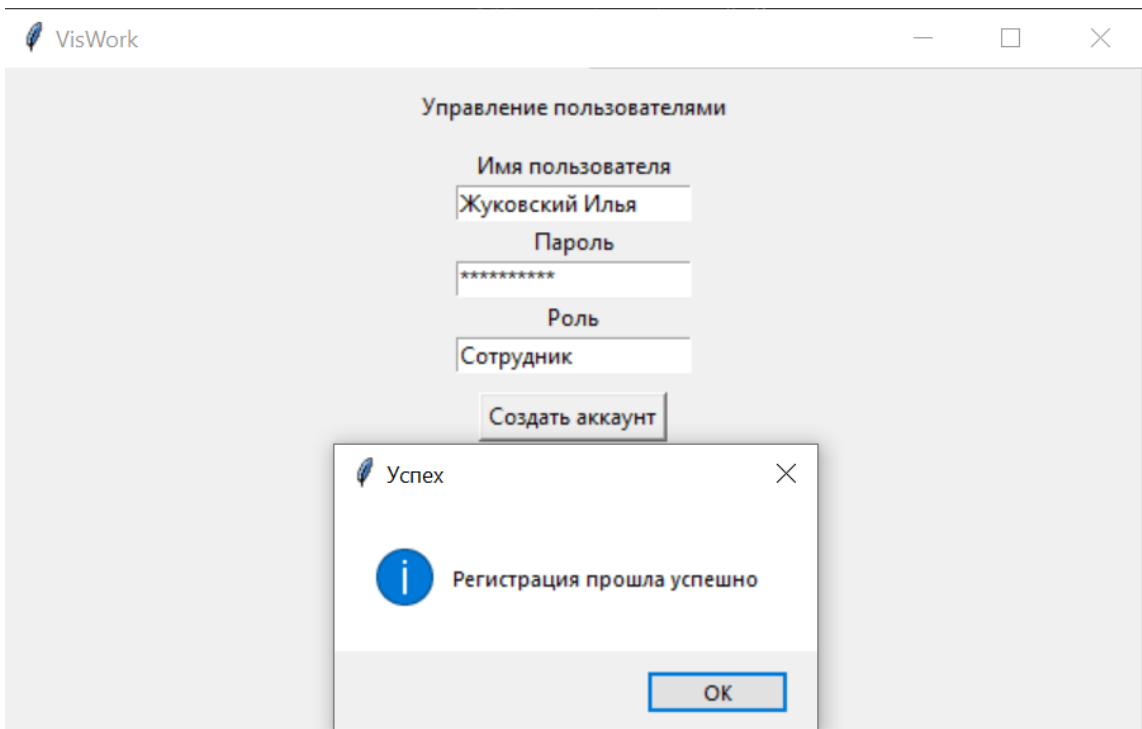


Рисунок 2.22 - Данные седьмого пользователя

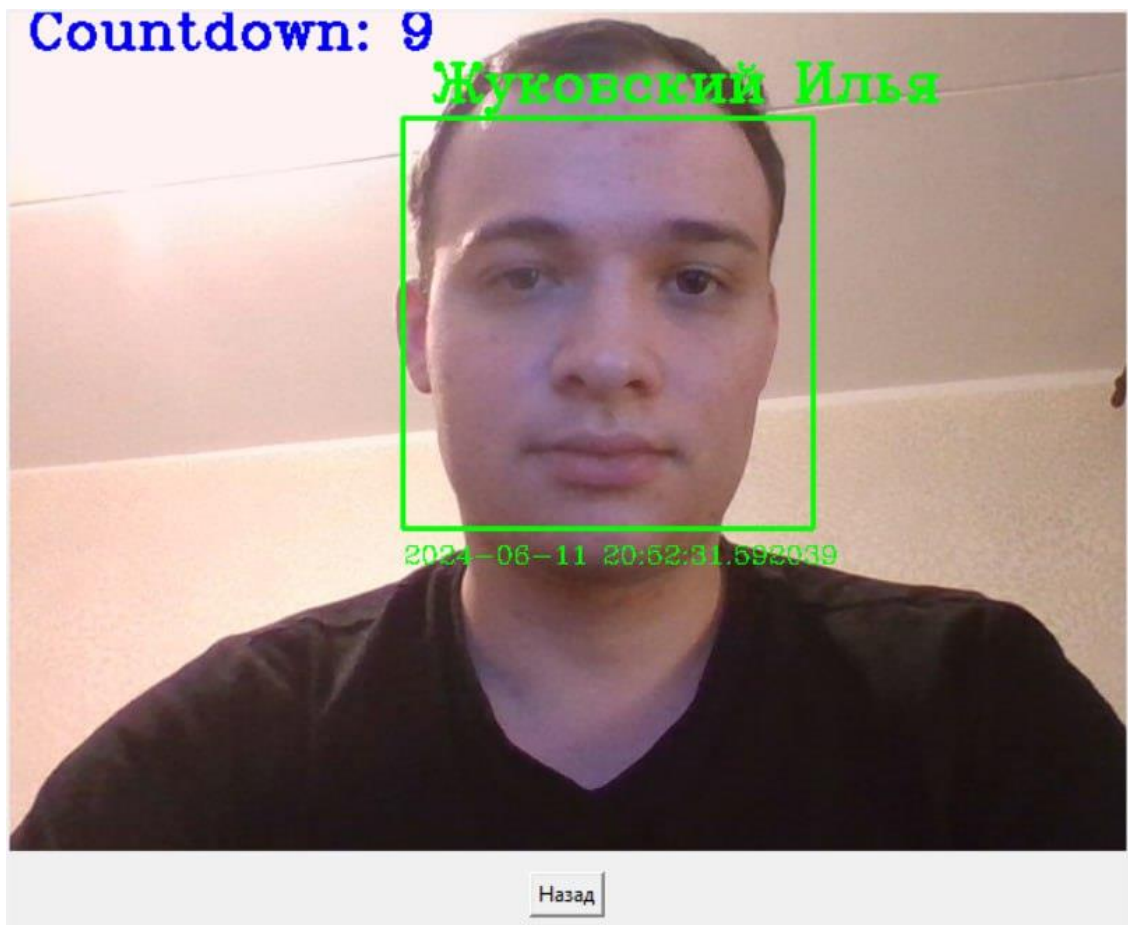


Рисунок 2.23 - Определение лица седьмого пользователя

При удалении из базы данных информации о сотруднике лицо данного человека перестанет идентифицироваться, а, следовательно, появится надпись «Unknown», как показано на рисунке 2.24.

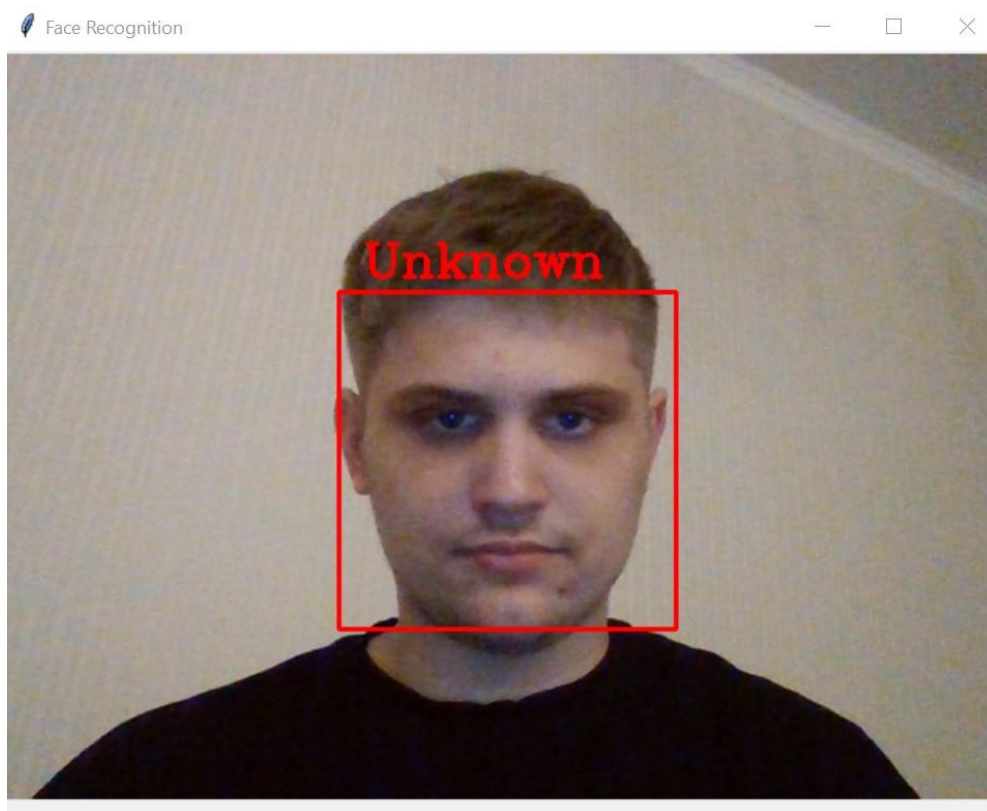


Рисунок 2.24 - Не идентифицированный пользователь

После внесения и открытия камеры программа показала отличные результаты определения пользователя из базы данных. Данный обзор демонстрирует работоспособность программы и модулей распознавания лиц.

2.11. Инструкция пользователю по работе с программным обеспечением

При открытии приложения появится окно авторизации в которое необходимо ввести персональные данные (рис. 2.25).

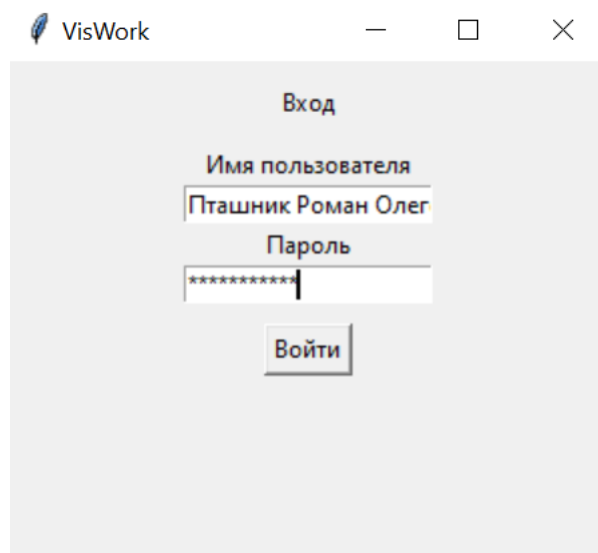


Рисунок 2.25 - Авторизация пользователя

Далее в зависимости от роли осуществляется переход на рабочий интерфейс.

Если ваши права «Администратор», то перед вами откроется интерфейс, показанный на рисунке 2.26.

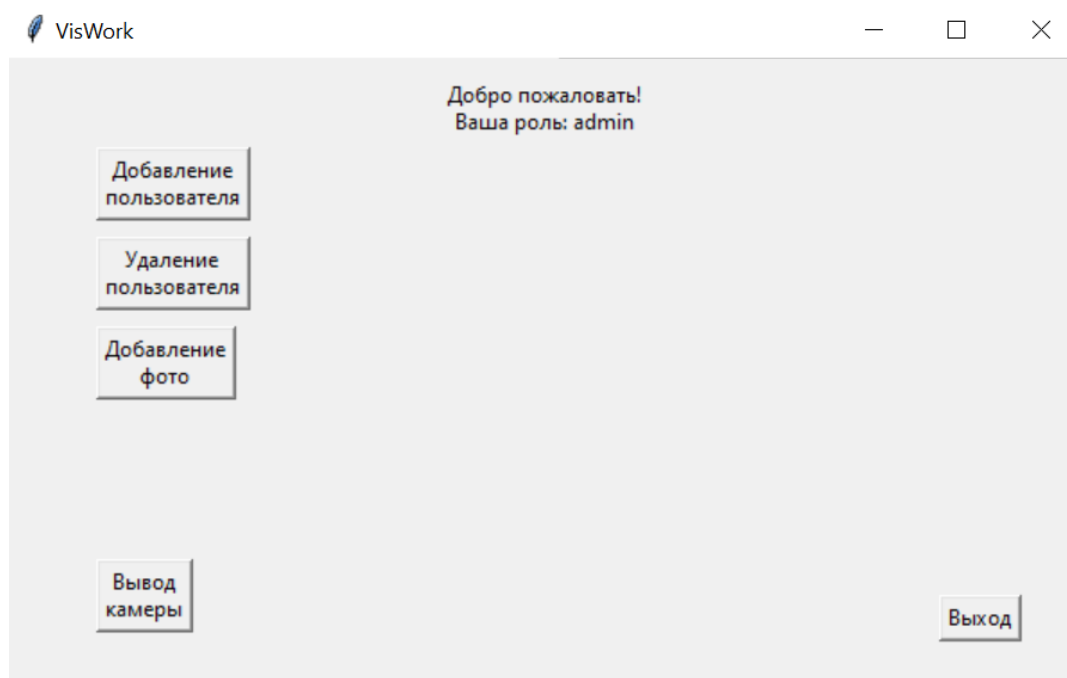


Рисунок 2.26 - Интерфейс пользователя «Администратор»

Каждая кнопка имеет свой функционал:

- **Добавление пользователя:** необходима для добавления сотрудника в базу данных и выдачи роли.
- **Удаление пользователя:** Удаления сотрудника из базы данных
- **Добавление фото:** После добавления пользователя, необходимо создать его личный датасет состоящий из фотографий, кнопка «Добавление фото» выполняет функцию создания датасета и сохранение его в базе данных конкретного пользователя.
- **Вывод камеры:** Осуществляет вывод камеры, которая определяет лица при помощи датасетов сохраненных в базе данных, а также фиксирует приход сотрудника на рабочую смену и выход из неё
- **Выход:** Выход из аккаунта.

Если ваша роль «Бухгалтер», то перед вами откроется следующее окно:

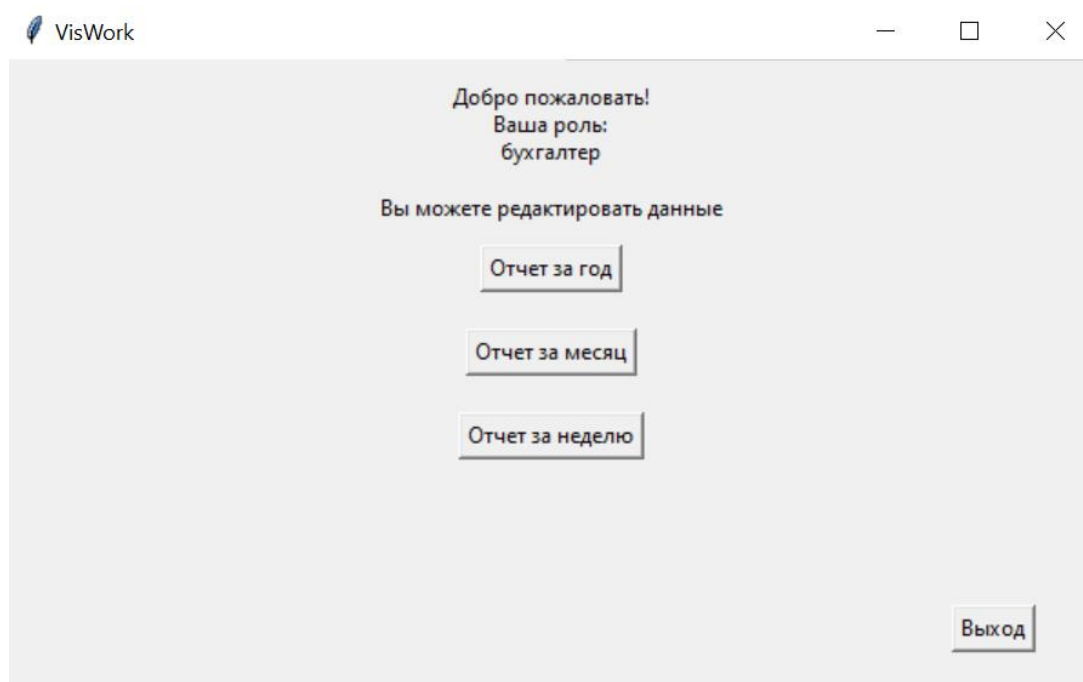


Рисунок 2.27 - Интерфейс пользователя «Бухгалтер»

Функционал кнопок:

- **Отчет за год:** При нажатии на кнопку, создается отчет о отработанном времени сотрудников за год

– Отчет за месяц: При нажатии на кнопку, создается отчет о отработанном времени сотрудников за месяц

– Отчет за неделю: При нажатии на кнопку, создается отчет о отработанном времени сотрудников за неделю.

– Выход: Выход из аккаунта.

Данный функционал реализован и уже может использоваться предприятиями.

ЗАКЛЮЧЕНИЕ

В ходе данного исследования была разработана и реализована программа для учета рабочего времени на основе технологии распознавания лиц. Актуальность данной работы обусловлена непрерывным развитием технологий в современной экономике и необходимостью повышения эффективности процессов управления персоналом. В условиях возрастающих требований к точности и надежности учета рабочего времени, традиционные методы регистрации устаревают и требуют замены на более современные и точные решения.

Разработанная система продемонстрировала возможность автоматизации процесса учета рабочего времени, что не только упрощает процедуру регистрации прихода и ухода сотрудников, но и способствует снижению человеческого фактора и уменьшению рисков, связанных с мошенничеством. Программа позволяет осуществлять следующие функции: распознавание лиц сотрудников; внесение новых сотрудников в базу данных; генерация отчетов за различные периоды (год, месяц, неделя).

Результаты тестирования подтвердили, что система распознавания лиц обладает высокой точностью и надежностью при работе в различных условиях: изменяющемся освещении, различных углах обзора и при наличии потенциальных препятствий. Интеграционная проверка показала корректное взаимодействие всех компонентов системы, обеспечивая согласованность данных и их оперативную обработку.

Также была проверена производительность системы: она демонстрирует стабильные результаты даже при высокой нагрузке и работе с большими объемами данных. Система прошла тестирование на моделирование исключительных ситуаций, таких как потеря видеосигнала и проблемы с базой данных, что свидетельствует о ее готовности к эксплуатации в реальных коммерческих или промышленных условиях.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Analysis market recognition technology [Электронный ресурс]. // TrendForce. Режим доступа: <https://www.trendforce.com/>
2. Кузнецов Д.А., Дамм В.А., Кузнецов А.В., Трегубов Р.Б. Сравнительный анализ популярных методов распознавания лиц на изображениях // Научный результат. Информационные технологии. - Т.4, №4, 2019
3. Методы ближайшего соседа и k-ближайших соседей [Электронный ресурс]. // Режим доступа: http://studbooks.net/2429081/informatika/metody_blizhayshego_soseda_blizhayshih_sosedey
4. Асташенкова, Л. К. Распознавание ключевых точек лица на изображении человека / Л. К. Асташенкова, С. Н. Рощупкина, К. В. Кудринская. - Текст: непосредственный // Молодой ученый. - 2019. - № 26
5. Метод Виолы — Джонса [Электронный ресурс]. // Wikipedia. – Режим доступа: https://ru.wikipedia.org/wiki/Метод_Виолы_—_Джонса
6. Видеоаналитика и распознавание лиц [Электронный ресурс]. // Wikipedia. – Режим доступа: https://video-praktik.ru/st_videoanalitika.html/ (дата обращения: 22.05.2024).
7. About OpenCV [Электронный ресурс]. // Режим доступа: <http://opencv.org/about.html>
8. Локальные бинарные шаблоны [Электронный ресурс]. // Режим доступа: https://ru.wikipedia.org/wiki/Локальные_бинарные_шаблоны (дата обращения: 01.06.2024).
9. Rasras R.J., Zahran B., Sara M.R. A developing digital signal clustering method using local binary pattern histogram // International Journal of Electrical and Computer Engineering. -2021. - V. 11. - № 1. - P. 872-876.
10. Rosebrock A. OpenCV EigenFaces for face recognition. 2021. URL: <https://pyimagesearch.com/2021/05/10/opencv-eigenfaces-for-face-recognition/>

11. Sanjeevkumar A., Suvarna N. Human Identification using Histogram of Oriented Gradients (HOG) and Non-Maximum Suppression (NMS) for ATM Video Surveillance. 2021. 10 с.
12. Сармьенто К., Саваж Х. (2020). Сравнение двух методов классификации объектов с использованием скрытых марковских моделей и сверточных нейронных сетей. 2020. 33 с. URL: <http://ia.spcras.ru/index.php/sp/article/view/14007/14825>
13. Hongda Tian, Smoke Detection in Videos Using Non-Redundant Local Binary Pattern-Based Features. Wollongong, Australia: Advanced Multimedia Research Lab, ICT Research Institute. [Электронный ресурс]. // Режим <https://scholar.google.com.au/citations?user=fZ9ln0oAAAAJ&hl=en>
14. Jinwang Feng, Dominant-Completed Local Binary Pattern for Texture Classification. Materials of International Conference on Information and Automation. Lijiang, China. [Электронный ресурс]. // Режим <https://dblp.org/pid/36/6962.html>
15. Junding Sun, New Local Edge Binary Patterns For Image Retrieval. [Электронный ресурс]. // Режим <https://dblp.org/pid/36/6962.html>
16. Хабрахабр - крупнейший ресурс для IT-специалистов. [Электронный ресурс].
17. Броневич А. Н. Лекции по методам машинного обучения [Электронный ресурс]. // Режим http://window.edu.ru/resource/files/lect_Lepskiy_Bronevich.pdf
18. Селянкин, В.В. Компьютерное зрение. Анализ и обработка изображений: Учебное пособие / В.В. Селянкин. - СПб.: Лань, 2019. - 152 с.
19. Гонсалес Рафаэл С., Вудс Ричард Е. Цифровая обработка изображений. - 3-е изд. - Техносфера, 2019. - 1104 с.
20. Фаулер, М Шаблоны корпоративных приложений / М Фаулер. – Уолсалл, Великобритания : Вильямс, 2020. – 544 с.

Листинг программы

```
import tkinter as tk
from tkinter import messagebox
from tkinter import simpledialog
import sqlite3
import cv2
from PIL import Image, ImageTk
from main import VideoWindow
import pickle
import numpy as np
from datetime import datetime

conn = sqlite3.connect('users.db')
c = conn.cursor()

c.execute('PRAGMA foreign_keys = ON;')

c.execute('''
    CREATE TABLE IF NOT EXISTS users (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        username TEXT UNIQUE NOT NULL,
        password TEXT NOT NULL,
        role TEXT NOT NULL,
        photo BLOB
    )''')

c.execute('''
    CREATE TABLE IF NOT EXISTS time_control (
        id INTEGER PRIMARY KEY,
        user_id INTEGER,
        date DATE NOT NULL,
        arrival_time TIME NOT NULL,
        departure_time TIME NOT NULL,
```



```

        screenshot BLOB,
        FOREIGN KEY(user_id) REFERENCES users(id)
    )'''
conn.commit()

class App(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Авторизация")
        self.geometry("300x250")
        self._init_login_screen()
        self.photos = []

    def _init_login_screen(self):
        for widget in self.wininfo_children():
            widget.destroy()

        tk.Label(self, text="Вход").pack(pady=10)

        tk.Label(self, text="Имя пользователя").pack()
        self.username_entry = tk.Entry(self)
        self.username_entry.pack()

        tk.Label(self, text="Пароль").pack()
        self.password_entry = tk.Entry(self, show='*')
        self.password_entry.pack()

        tk.Button(self, text="Войти",
command=self.login).pack(pady=10)

    def login(self):
        username = self.username_entry.get()
        password = self.password_entry.get()

        c.execute('SELECT * FROM users WHERE username=? AND
password=?', (username, password))
        user = c.fetchone()

```

```

    if user:
        self._init_main_screen(user[3])
    else:
        messagebox.showerror("Ошибка", "Неверное имя пользователя
или пароль")

def open_video_window(self):
    self.video_window = VideoWindow()

def _init_main_screen(self, role):
    for widget in self.wininfo_children():
        widget.destroy()
    self.geometry("600x350")
    role = role.strip().lower()
    tk.Label(self, text=f"Добро пожаловать! Ваша роль: {role}",
wraplength=120).pack(pady=10)
    if role == "admin" or role == "администратор":
        tk.Button(self, text="Добавление пользователя",
command=self._manage_users, wraplength=120).place(x=50, y=50)
        tk.Button(self, text="Удаление пользователя",
command=self._manage_users, wraplength=120).place(x=50, y=100)
        tk.Button(self, text="Добавление фото",
command=self._add_photo, wraplength=80).place(x=50, y=150)
        tk.Button(self, text="Вывод камеры",
command=self.open_video_window, wraplength=80).place(x=50, y=280)
    elif role == "employee" or role == "сотрудник":
        tk.Label(self, text="Вы можете просматривать
данные").pack()
    elif role == "accountant" or role == "бухгалтер":
        tk.Label(self, text="Вы можете редактировать
данные").pack()
        tk.Button(self, text="Отчет за год",
command=self._add_photo).pack(pady=10)
        tk.Button(self, text="Отчет за месяц",
command=self._add_photo).pack(pady=10)
        tk.Button(self, text="Отчет за неделю",
command=self._add_photo).pack(pady=10)

```

```
tk.Button(self, text="Выход", command=self._init_login_screen,
wraplength=100).place(x=520, y=300)
```

```
def _manage_users(self):
```

```
    for widget in self.wininfo_children():
        widget.destroy()
```

```
tk.Label(self, text="Управление пользователями").pack(pady=10)
```

```
tk.Label(self, text="Имя пользователя").pack()
```

```
self.new_username_entry = tk.Entry(self)
```

```
self.new_username_entry.pack()
```

```
tk.Label(self, text="Пароль").pack()
```

```
self.new_password_entry = tk.Entry(self, show='*')
```

```
self.new_password_entry.pack()
```

```
tk.Label(self, text="Роль").pack()
```

```
self.new_role_entry = tk.Entry(self)
```

```
self.new_role_entry.pack()
```

```
tk.Button(self, text="Создать аккаунт",
command=self.register).pack(pady=10)
```

```
tk.Button(self, text="Назад", command=lambda:
self._init_main_screen("admin")).pack(pady=10)
```

```
c.execute('SELECT username, role FROM users')
```

```
users = c.fetchall()
```

```
for user in users:
```

```
    tk.Label(self, text=f"Имя: {user[0]}, Роль:
{user[1]}").pack()
```

```
def register(self):
```

```
    username = self.new_username_entry.get()
```

```
    password = self.new_password_entry.get()
```

```
    role = self.new_role_entry.get()
```

```

try:
    c.execute('INSERT INTO users (username, password, role)
VALUES (?, ?, ?)', (username, password, role))
    conn.commit()
    messagebox.showinfo("Успех", "Регистрация прошла успешно")
    self._manage_users()
except sqlite3.IntegrityError:
    messagebox.showerror("Ошибка", "Имя пользователя уже
существует")

def _add_photo(self):
    username = simpledialog.askstring("Имя пользователя", "Введите
ИМЯ ПОЛЬЗОВАТЕЛЯ:")
    if not username:
        return

    c.execute('SELECT * FROM users WHERE username=?', (username,))
    user = c.fetchone()

    if not user:
        messagebox.showerror("Ошибка", "Пользователь не найден")
        return

    self.camera_window = tk.Toplevel(self)
    self.camera_window.title("Камера")

    self.lmain = tk.Label(self.camera_window)
    self.lmain.pack()

    self.cap = cv2.VideoCapture(0)

    tk.Button(self.camera_window, text="Сделать фото",
command=lambda: self._take_photos(username)).pack()

    self._show_frame()

def _show_frame(self):
    _, frame = self.cap.read()

```

```

        frame = cv2.flip(frame, 1)
        cv2image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)
        img = Image.fromarray(cv2image)
        imgtk = ImageTk.PhotoImage(image=img)
        self.lmain.imgtk = imgtk
        self.lmain.configure(image=imgtk)
        self.lmain.after(10, self._show_frame)

def _take_photos(self, username):
    if len(self.photos) < 30:
        _, frame = self.cap.read()
        frame = cv2.flip(frame, 1)
        self.photos.append(frame)
        count = len(self.photos)
        messagebox.showinfo("Фото сделано", f"Фото {count}/30
сделано")

        if len(self.photos) == 30:
            self.cap.release()
            self.camera_window.destroy()
            img_bytes = pickle.dumps(self.photos)
            c.execute('UPDATE users SET photo=? WHERE username=?',
(img_bytes, username))
            conn.commit()

            messagebox.showinfo("Успех", "Фото добавлены успешно")

if __name__ == "__main__":
    app = App() # Создание экземпляра класса App
    app.title("VisWork")
    app.mainloop()

conn.close()

class VideoWindow:
    def __init__(self):
        self.root = tk.Toplevel()

```

```

self.root.title("Face Recognition")
self.countdown = 100
#self.recognizer = recognizer
self.time_displayed = False
self.screenshot_taken = False
self.training_available = True

self.display = tk.Label(self.root)
self.display.pack()

self.back_button = tk.Button(self.root, text="Назад",
command=self.on_closing)
self.back_button.pack(pady=10)

self.facedetect =
cv2.CascadeClassifier("haarcascade/haarcascade_frontalface_default.xml
")
self.recognizer = cv2.face.LBPHFaceRecognizer_create()

self.cam = cv2.VideoCapture(0)

self.faces, self.ids, self.names = self.fetchTrainingData()
self.faces = [self.normalize_image(face) for face in
self.faces]

self.training_available = len(self.faces) > 0 and
len(self.ids) > 0
if self.training_available:
self.recognizer.train(self.faces, np.array(self.ids))

self.update_frame()

self.root.protocol("WM_DELETE_WINDOW", self.on_closing)

def getProfile(self, user_id):
conn = sqlite3.connect("users.db")
cursor = conn.execute("SELECT * FROM users WHERE id=?",
(user_id,))

```

```

profile = cursor.fetchone()
conn.close()
return profile

def fetchTrainingData(self):
    conn = sqlite3.connect("users.db")
    cursor = conn.execute("SELECT id, photo, username FROM users")
    faces = []
    ids = []
    names = []
    for row in cursor:
        user_id = row[0]
        photo_data = row[1]
        user_name = row[2]
        if photo_data is not None:
            try:
                dataset = pickle.loads(photo_data)
            except pickle.UnpicklingError:
                print(f"Error unpickling data for user_id
{user_id}")
                continue

            for face_image in dataset:
                if isinstance(face_image, bytes):
                    face_image = np.frombuffer(face_image,
np.uint8)

                    face_image = cv2.imdecode(face_image,
cv2.IMREAD_GRAYSCALE)
                elif face_image.ndim == 3:
                    face_image = cv2.cvtColor(face_image,
cv2.COLOR_BGR2GRAY)

                faces.append(face_image)
                ids.append(user_id)
                names.append(user_name)

    conn.close()
    return faces, ids, names

def normalize_image(self, image):

```

```

    if image.ndim == 3:
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    return cv2.equalizeHist(image)

def update_frame(self):
    ret, img = self.cam.read()
    if not ret:
        print("Не удалось прочитать с камеры")
        return

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    gray = self.normalize_image(gray)
    faces_detected = self.facedetect.detectMultiScale(gray, 1.3,
5)

    if len(faces_detected) > 0:
        for (x, y, w, h) in faces_detected:
            username = "Unknown"
            color = (0, 0, 255)
            recognized = False

            if self.training_available:
                face_to_predict = gray[y:y + h, x:x + w]
                face_to_predict = cv2.resize(face_to_predict,
(200, 200))

                id, conf =
self.recognizer.predict(face_to_predict)
                if conf < 90:
                    profile = self.getProfile(id)
                    if profile:
                        username = profile[1]
                        color = (0, 255, 0)
                        recognized = True

            cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
            cv2.putText(img, f" {username}", (x, y - 10),
cv2.FONT_HERSHEY_COMPLEX, 1, color, 2)

```



```

        if recognized:
            current_datetime = datetime.now()
            date = current_datetime.strftime("%Y-%m-%d")
            time = current_datetime.strftime("%H:%M")
            cv2.putText(img, f"{current_datetime}", (x, y + h
+ 20), cv2.FONT_HERSHEY_COMPLEX, 0.5, color, 1)

            if self.countdown > 0:
                cv2.putText(img, f"Countdown:
{self.countdown}", (10, 20), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0),
2)

                self.countdown -= 1
            else:
                if not self.screenshot_taken:
                    cv2.imwrite("screenshot.jpg", img)
                    self.screenshot_taken = True

            if not self.time_displayed and self.countdown <=
0:

                print(current_datetime)
                self.time_displayed = True
                print(current_datetime)
                print(f"Recognized: (ID: {id}, Username:
{username}))")
            else:
                self.countdown = 100
                self.screenshot_taken = False
                self.time_displayed = False
            self.img = img
        else:
            self.countdown = self.screenshot_taken = False
            self.time_displayed = False
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            img = Image.fromarray(img)
            img_tk = ImageTk.PhotoImage(image=img)

            self.display.imgtk = img_tk

```

```
self.display.config(image=imgTk)
self.display.after(10, self.update_frame)
def on_closing(self):
    self.cam.release()
    self.root.destroy()
    cv2.destroyAllWindows()
```