

УДК 004.584

ИСПОЛЬЗОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ДЛЯ ОХРАНЫ ПРЕДПРИЯТИЯ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ РАСПОЗНОВАНИЯ ЛИЦ

Маняков.С.Е., студент гр. ИТб-211, IV курс

Сыркин И.С., доцент (к.н.) кафедры ИиАПС

Кузбасский государственный технический университет

имени Т. Ф. Горбачева

г. Кемерово

Технология распознавания лиц позволяет идентифицировать и установить личность человека, чтобы предотвратить несанкционированный доступ к территории предприятия и подтвердить личность каждого человека.

В настоящее время эта технология находит широкое применение в различных сферах, от обеспечения безопасности до улучшения пользовательского опыта. Например, в крупных городах России, таких как Москва и Санкт-Петербург, установлены камеры, которые могут автоматически распознавать лица и сопоставлять их с базами данных разыскиваемых преступников. Однако важно отметить, что система распознавания лиц требует регулярного обновления базы данных лиц, что может привести к задержкам в работе охраны, если не будет своевременного контроля.

На данный момент в большинстве предприятий процесс уведомления сотрудников об обновлении их данных в системе распознавания лиц осуществляется вручную. Сотрудник службы безопасности проверяет список всех работников, зарегистрированных в системе, и информирует их о необходимости обновления информации.

Для автоматизации данного процесса разрабатывается система автоматического уведомления сотрудников с минимальным человеческим вмешательством. Эта система, названная «Система автоматического оповещения о распознавании лиц» (СООРЛ), будет включать в себя следующие механизмы информирования:

- Уведомление по электронной почте;
- Уведомление через чат-бота;
- Уведомление с помощью автозапускаемого приложения на персональном компьютере сотрудника.

В данной статье мы рассмотрим, как можно интегрировать приложение в систему для эффективного информирования сотрудников об их статусе в системе распознавания лиц.

Само приложение будет разработано с использованием Python и библиотеки tkinter для создания графического интерфейса. В систему также будет включена база данных (БД), созданная в системе управления базами данных Microsoft SQL Server. Эта база данных будет хранить информацию о сотрудниках, включая их ФИО, паспортные данные и фотографии.

Предполагается, что приложение будет работать на сервере в круглосуточном режиме для обработки данных о лицах пользователей. Система будет функционировать на основе концепции «Открытого турникета», что позволит избежать образования больших очередей. При каждом входе человека в здание программа будет запрашивать информацию из базы данных. Если система распознавания лиц не сможет идентифицировать посетителя, она автоматически заблокирует турникет и вызовет охрану. В случае успешной идентификации приложение отправит уведомления всем сотрудникам охраны о поступивших данных, обеспечивая тем самым высокий уровень безопасности и контроля доступа. Диаграмма деятельности процесса оповещения представлена на рисунке 1.

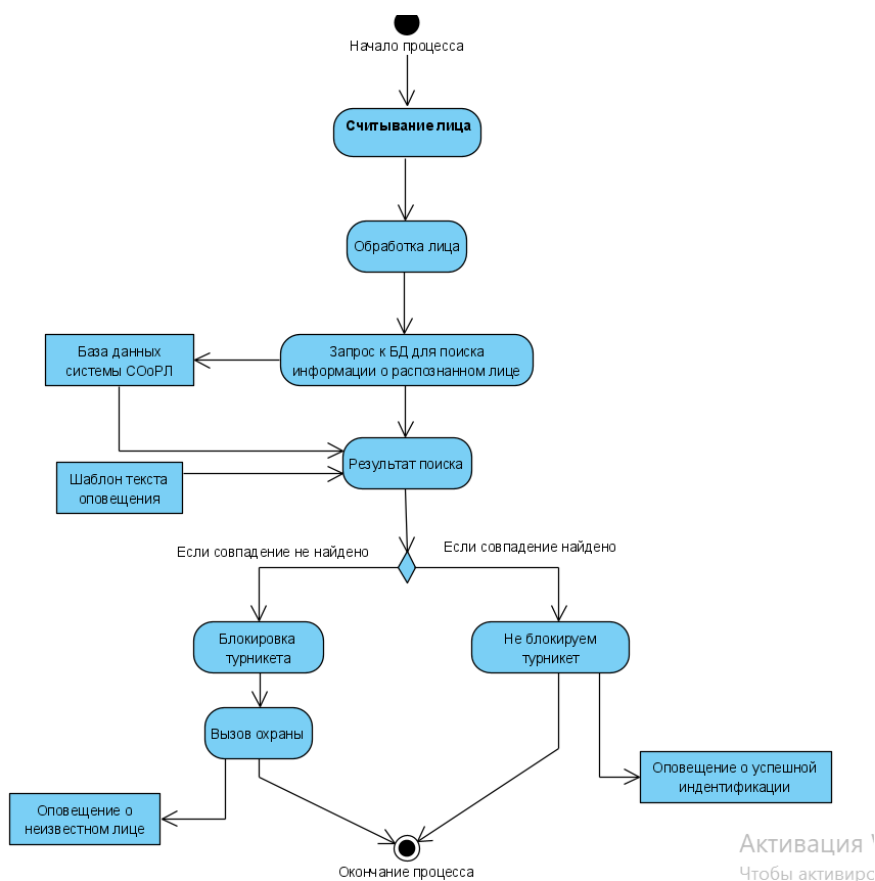


Рисунок 1 – Диаграмма деятельности процесса оповещения.

В автоматизированной системе охраны предприятия, использующей технологию распознавания лиц, процесс идентификации пользователя начинается с захвата изображения лица сотрудника с помощью камеры. Это изображение затем передается в систему для обработки. Система сравнивает полученное изображение с уже сохраненными данными в базе данных, где хранятся фотографии сотрудников, а также их ФИО и другие идентифицирующие данные.

Если система находит совпадение, она извлекает информацию о сотруднике и отправляет уведомление о успешной идентификации. В ответ пользователю приходит сообщение с приветствием, где указывается его имя и фамилия, подтверждающее успешное распознавание.

Если же система не находит совпадений, она автоматически блокирует турникет и вызывает охрану, что дополнительно усиливает уровень безопасности на предприятии. Таким образом, процесс идентификации не только минимизирует возможность ошибок, но и предотвращает несанкционированный доступ, обеспечивая надежный контроль за входом на территорию. Диаграмма деятельности для идентификации пользователя представлена на рисунке 2.

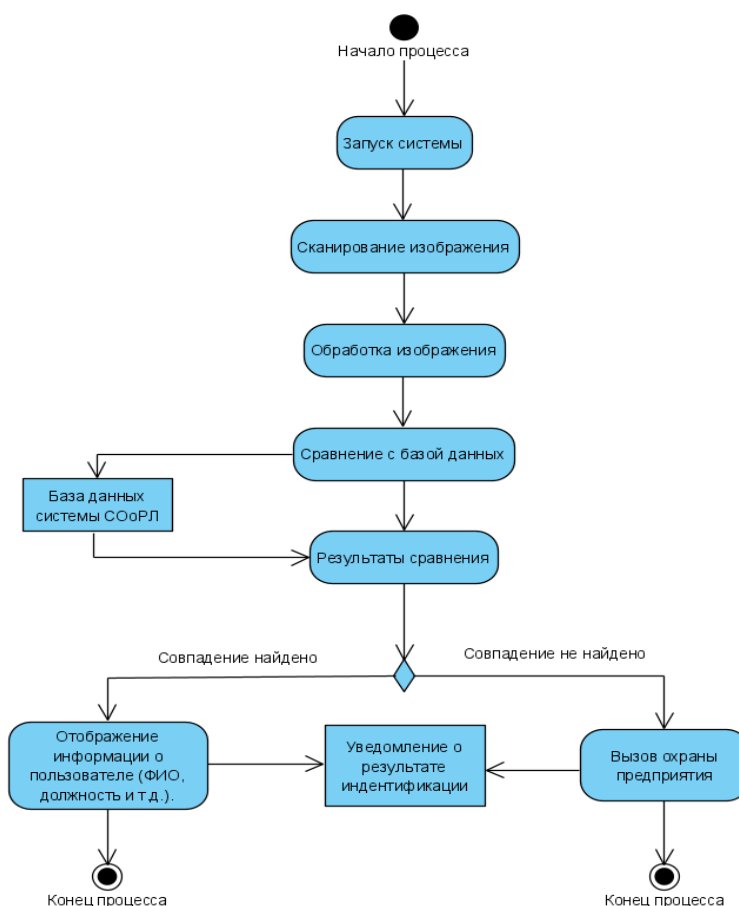
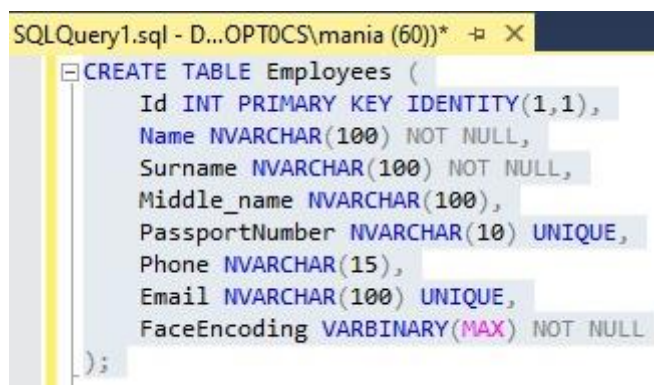


Рисунок 2 – Диаграмма деятельности процесса идентификации пользователя.

Перед началом разработки приложения необходимо создать базу данных в Microsoft SQL Server. Эта база данных будет служить основным хранилищем информации о сотрудниках, включая их ФИО, паспортные данные и фотографии, которые будут использоваться для процесса идентификации.

На первом этапе проектирования базы данных следует определить структуру таблиц, которые будут содержать необходимые атрибуты. Таблица "Employees" будет включать следующие поля: "ID", "ФИО", "Номер паспорта", "Фото", "Email" и "Телефон". Каждый сотрудник будет иметь уникальный идентификатор, который позволит легко управлять записями и выполнять запросы.

После проектирования структуры базы данных необходимо создать её в SQL Server. Для этого можно воспользоваться SQL Server Management Studio (SSMS), где с помощью SQL-запросов создаются таблицы и устанавливаются связи между ними. Также важно предусмотреть индексы для оптимизации быстродействия запросов при работе с большими объёмами данных. Запрос на создание таблицы изображена на рисунке 3.

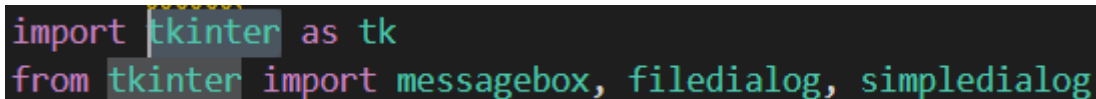


```
SQLQuery1.sql - D:\OPT0CS\mania (60)) * -> X
CREATE TABLE Employees (
    Id INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(100) NOT NULL,
    Surname NVARCHAR(100) NOT NULL,
    Middle_name NVARCHAR(100),
    PassportNumber NVARCHAR(10) UNIQUE,
    Phone NVARCHAR(15),
    Email NVARCHAR(100) UNIQUE,
    FaceEncoding VARBINARY(MAX) NOT NULL
);
```

Рисунок 3 – Создание запроса на создание таблицы в SQL Server

Для разработки приложения на Python с использованием библиотеки tkinter необходимо установить саму библиотеку, если она еще не установлена. Tkinter является стандартной библиотекой для создания графических интерфейсов в Python и обычно поставляется вместе с установкой Python.

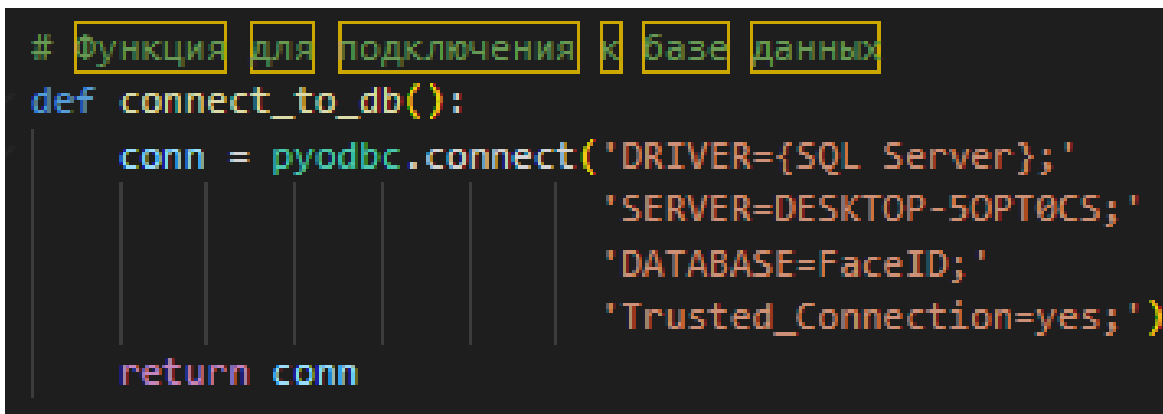
После установки Python, можно начать разработку приложения, создав новый файл с расширением .py. В файле нужно импортировать библиотеку tkinter, а также необходимые модули для работы с графическим интерфейсом. Библиотека изображена на рисунке 4.



```
import tkinter as tk
from tkinter import messagebox, filedialog, simpledialog
```

Рисунок 4 – Импорт библиотеки tkinter

Необходимо настроить подключение к базе данных SQL Server. Для этого используем библиотеку pyodbc. Создадим функцию, которая будет устанавливать соединение с базой данных. Функция показана на рисунке 5.



```
# Функция для подключения к базе данных
def connect_to_db():
    conn = pyodbc.connect('DRIVER={SQL Server};'
                          'SERVER=DESKTOP-50PT0CS;'
                          'DATABASE=FaceID;'
                          'Trusted_Connection=yes;')
    return conn
```

Рисунок 5 – Функция с подключением к БД SQL Server

После этого можно реализовать функционал для распознавания лиц. Для этого создадим функционал, который будет использовать

библиотеку `face_recognition` для обработки изображений и сравнения лиц. Функционал с распознаванием лица показана на рисунке 6.

```
# Загружаем изображение и преобразуем его для отображения
image = face_recognition.load_image_file(file_path)
face_encodings = face_recognition.face_encodings(image)

if face_encodings:
    face_encoding = face_encodings[0]
    employee_info = get_employee_info(face_encoding)

    if employee_info:
        messagebox.showinfo("Сотрудник найден", f"Сотрудник: {employee_info[0]} {employee_info[1]}")
    else:
        messagebox.showwarning("Неизвестное лицо!", "Неизвестное лицо! Уведомление отправлено охране.")
```

Рисунок 6 – Распознавание и сравнение лица

С помощью этих функций мы можем создать интерфейс, который позволит пользователю загружать изображения и распознавать лица. Это делается с помощью класса `Tk`, который представляет собой корневое окно. Мы можем задать заголовок окна и его размеры. После этого можно добавить различные элементы интерфейса, такие как метки, кнопки и текстовые поля. Например, после проверки вывести сообщение «Сотрудник найден» или «Сотрудник не найден». Форма охраны изображена на рисунке 7.

```
# Функция для охраны
def security_form():
    security_window = tk.Tk()
    security_window.title("Охрана")

    # Кнопка для загрузки изображения
    tk.Button(security_window, text="Загрузить изображение для распознавания", command=lambda: load_image_for_security(security_window)).pack(pady=10)

    tk.Label(security_window, text="Лицо человека вошедшее в здание:").pack()

    # Место для отображения изображения
    global img_label
    img_label = tk.Label(security_window)
    img_label.pack()

    tk.Button(text="Добавить сотрудника", command=check_password).pack()
    security_window.mainloop()
```

Рисунок 7 – Интерфейс приложения Python

Тестирование приложения «СООРЛ» показано на рисунках 8 и 9.

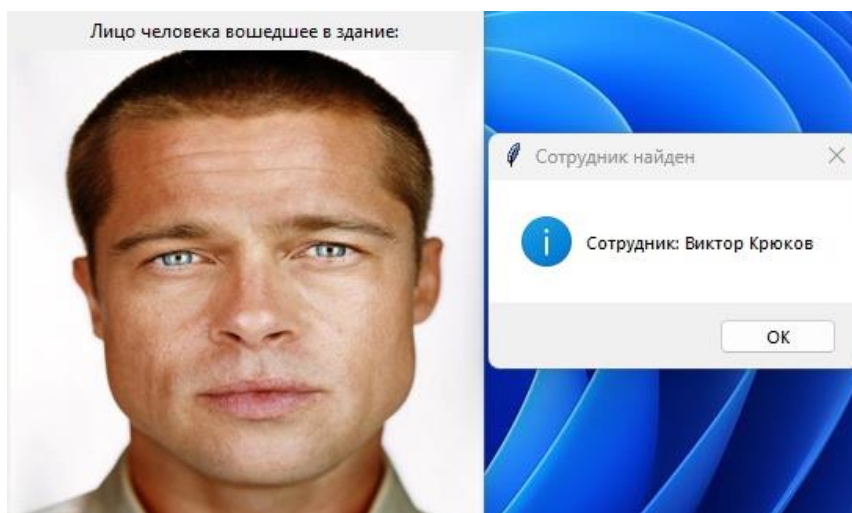


Рисунок 8 – Тестирование приложения «СОоРЛ» (Часть 1)

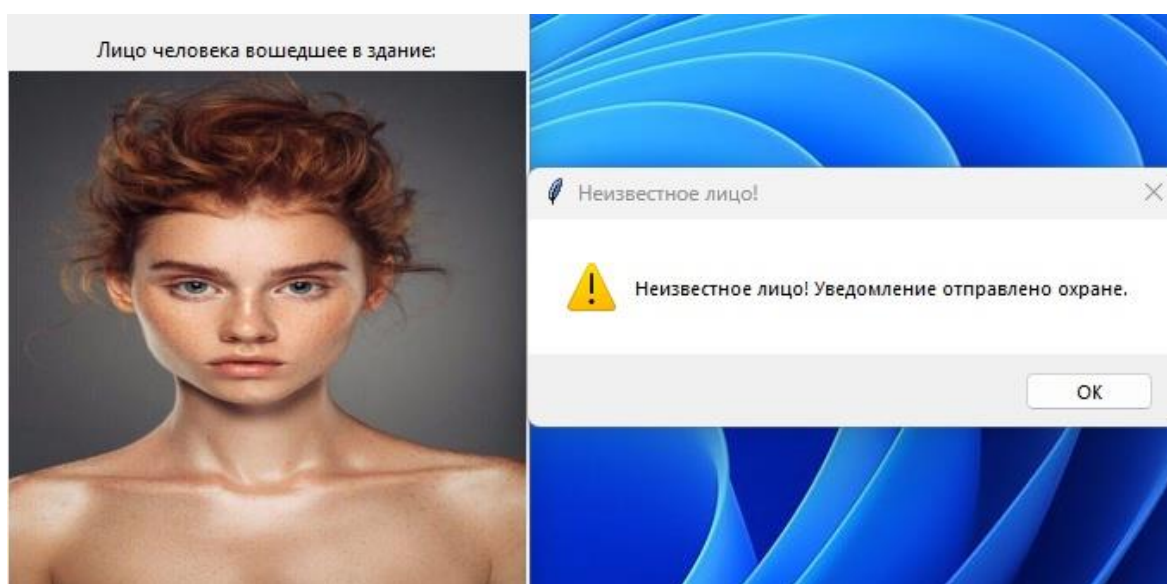


Рисунок 9 – Тестирование приложения «СОоРЛ» (Часть 2)

Приложение «СОоРЛ», разработанное с использованием технологии распознавания лиц, значительно упрощает процессы идентификации сотрудников и повышает уровень безопасности на территории предприятия. Благодаря этому приложению, необходимость в ручной проверке документов и личных данных сотрудников минимизируется. Система автоматически распознает лица сотрудников, что позволяет быстро и точно идентифицировать их при входе на объект.

Список литературы:

1. Документация к библиотеке Face Recognition [Электронный ресурс] — URL: <https://face-recognition.readthedocs.io/en/latest/readme.html> (дата обращения: 24.03.2025)

2. Работаем с SQL Server с помощью Python [Электронный ресурс] – URL: <https://proglib.io/p/rabotaem-s-sql-server-s-pomoshchyu-python-2020-04-18> (дата обращения: 23.03.2025)
3. OpenCV в Python. Часть 1 [Электронный ресурс] – URL: <https://habr.com/ru/articles/519454/> (дата обращения: 20.03.2025)
4. OpenCV в Python. Часть 2 [Электронный ресурс] – URL: <https://habr.com/ru/articles/528144/> (дата обращения: 20.03.2025)
5. OpenCV в Python. Часть 3 [Электронный ресурс] – URL: <https://habr.com/ru/articles/539228/> (дата обращения: 21.03.2025)
6. OpenCV в Python. Часть 4 [Электронный ресурс] – URL: <https://habr.com/ru/articles/547218/> (дата обращения: 21.03.2025)
7. Моделирование на языке UML в среде Visual Paradigm 14. [Электронный ресурс] – URL: <http://sp.cs.msu.ru/ooap/exer2017.html#exer34> (дата обращения: 24.03.2025).