

РАЗРАБОТКА ПРОГРАММНОГО РЕШЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ ПОЛЬЗОВАТЕЛЕЙ

**Александрова Н.Р., кандидат экономических наук, доцент
ФГБОУ ВО Ульяновский ГАУ**

**Субаева А.К., доктор экономических наук, доцент,
subaeva.ak@mail.ru**

**Минабутдинов Р.И., студент, Sksenia055@gmail.com
ФГБОУ ВО «Казанский национальный исследовательский
технический
университет им. А.Н. Туполева – КАИ»
Чистопольский филиал «ВОСТОК»**

Ключевые слова: камера видеонаблюдения, контроль, система видеонаблюдения, распознавание.

В статье представлена разработка программного решения для распознавания пользователей. Данная система планируется для контрольно-пропускных пунктов.

Введение. Камеры видеонаблюдения используются в разных сферах, начиная от личной безопасности, до раскрытия тяжких преступлений. В данной статье, будет рассмотрена камера видеонаблюдения с её архитектурой сохранения данных, будут рассмотрены основные функции такие как: записи видео, сохранение данных о человеке, внесение фотографий человека в базу данных, запись о появлении человека и примеры работы камеры.

На предприятии ежедневно требуется контроль времени прихода и ухода работника на предприятие. С точки зрения безопасности требуется реализовать систему контроля без пропусков, таким образом исключая возможность утери пропуска или использования пропуска иными людьми. В рамках исследования разработана система, позволяющая реализовать пропуск человека по его лицу. Для решения данной проблемы выбрана система бесконтактного контроля, камера

видеонаблюдения. В статье рассмотрен функционал, реализованный для камеры видеонаблюдения.

Материалы и методы исследований. Необходимость данной разработки является в том, чтобы создать собственную интеллектуальную систему видеонаблюдения.[1] Добавить функционал отслеживания человека в тот или иной период времени появления на камере, сделать вывод информация о человеке появившимся в кадре и разработать функцию добавления людей, которых камера ещё не знает.

Задачи для создания интеллектуальной системы следующие:

1. Создать базу данных, в которой будут храниться фотографии людей;
2. Создать документ, в котором будет храниться информация о людях;
3. Создать файл, в котором будет отображаться появления людей в каждый момент времени;
4. Создать визуальное приложение, в котором будет удобно отображать информацию о людях;
5. Сделать папку, в которой будут храниться видеозаписи с камеры.

Результаты исследований и их обсуждение. Для решения поставленных задач выбрана среда программирования Visual Code и язык программирования Python. Язык программирования Python – это объектно-ориентированный язык программирования высокого уровня, характеризующийся динамической типизацией и мультиплатформенностью. Основной особенностью Python является наличие многообразных библиотек компьютерного зрения, что необходимо для реализации проекта. [2] Для работы с изображениями использовались библиотеки `face_recordings` и `cv2`. Данные библиотеку отвечают за считывания изображения с камеры, и определения границ лица. Для работы со временем использовалась библиотека `datetime`, чтобы определять момент времени появления человека. Так же использована библиотека `thearding` для работы с потоками. Многопоточность необходима для того, чтобы во время заполнения данных, камера работала в фоновом режиме. Изначально требовалось создать структуру программы. Нужно было создать файлы, в которых

будет храниться информация, для большего удобства всё нужно хранить в одном проекте. В папке «PRO2» были созданы следующие файлы:

1. Папка «Database» в ней хранятся файлы, в которых записывается та или иная информация о пользователях;
2. Файл «Detections.log» хранит в себе время появления пользователей на камере;
3. Файл «Faces.dat» хранит информацию с фотографии из базы данных;
4. Файл «People_info.json» хранит информацию о людях, их номер телефона, адрес и родителей;
5. Папка «Known_faces» хранит предписанные именами папки, в которых хранятся фотографии людей;
6. Папка «Recordings» хранит записанные видео.

Структура программы сохранения данных представлена на рисунке 1.

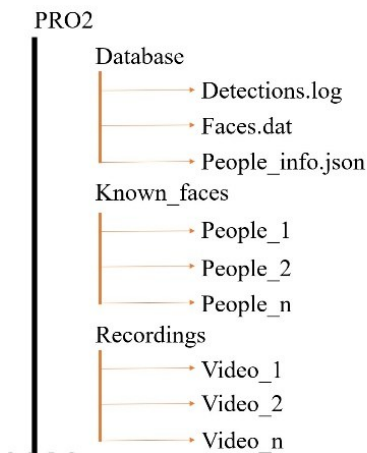


Рисунок 1 – Структура сохранения данных

Далее потребовалось написать программу с содержанием большого количества функций. Вот основные из них:

1. Функция `def load_database(db_path)` проверяет наличие в файле «Faces.dat» информации и загружает её в приложение;
2. Функция `def load_person_info(info_path)` анализирует файл «People_info.json» на наличие информации;

3. Функция `def save_person_info(info_path, name, info)` сохраняет информацию о человеке в файл «People_info.json»;

4. Функция `def capture_faces_for_new_person(name, cap, output_dir, num_shots=15, delay=0.2)` необходима для добавления фотографий новых пользователей;

5. Функция `def log_detection(log_file, name, timestamp, last_log_times)` срабатывает при обнаружении лица;

6. Функция `def add_new_person_flow(tk_manager, cap, faces_dir, db_path, info_path)` выводит на экран окошки в которые нужно ввести информацию о новых пользователях.

7. Функция `def update_database(database_path, faces_dir)` отвечает за обновление базы данных, после внесения нового человека в неё;

8. Class `SimpleDialogManager` необходимый для вызова диалоговых окон поверх приложения;

9. Class `VideoRecorder` запускает начало съёмки видео с камеры в момент запуска приложения и сохраняет видео в папку «Recordings» после окончания работы приложения;

10. Class `FaceRecognitionUI` отвечает за создания графического интерфейса, где с левой стороны показывается изображение с камеры, с правой стороны информация о человеке попавшего на камеру и снизу дополнительная информация о приложении.

Данные функцию позволяют работать с информацией о пользователях. После был создан внешний вид приложения, оформленный в классе «class `FaceRecognitionUI`». Так же с помощью класса «class `VideoRecorder`» была добавлена функция записи видео и сохранения его в папку «Recordings». Благодаря классу «class `TkinterManager`» устанавливается многопоточность, для правильного функционирования программы.

Пример работы программы приведён на рисунке 2.

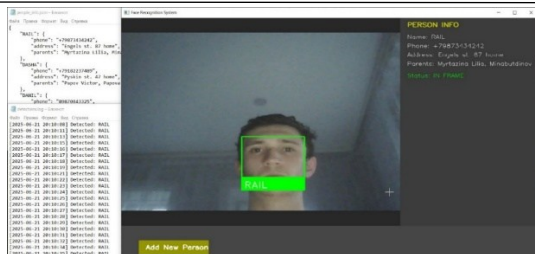


Рисунок 2 – Пример работы программы

В случае, когда на камеру попадает ранее неизвестное лицо, появляется возможность добавить его в базу данных при нажатии на клавишу “А”. Пример приведён на рисунке 3.

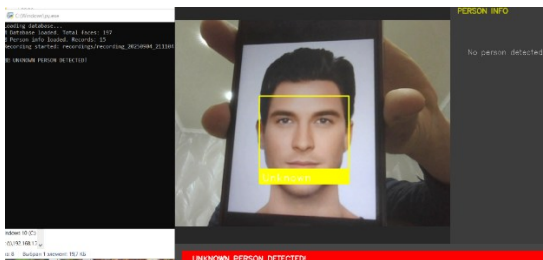


Рисунок 3 – Пример работы программы с неизвестным человеком

Код определение человека в момент времени его появления в кадре приведён на листинге 1.

Листинг 1

```
for (top, right, bottom, left), face_enc in zip(face_locations, face_encodings):
    name = "Unknown" color = ui.colors["unknown"]
    matches = face_recognition.compare_faces(database["encodings"], face_enc,
TOLERANCE)
    if True in matches:
        matched_idx = matches.index(True)
        name = database["names"][matched_idx]
        color = ui.colors["known"]
        current_name = name
        if name in people_info:
            current_person_info = {
```

```

"name": name, "phone": people_info[name].get("phone", "N/A"),
"address": people_info[name].get("address", "N/A"),
"parents": people_info[name].get("parents", "N/A"),
"in_frame": True, "last_seen": time.time()
}

```

Изначально запускается цикл и все выделяются как “Unknown”, то есть неизвестные, за это отвечает строка name = “Unknown”. В строке matches = face_recognition.compare_faces(database[“encodings”], face_enc, TOLERANCE) происходит сравнение изображений из базы данных с изображением на камере, если человек есть в базе данных строка if True in matches: тогда создаётся выводится информация о человеке, что показано в следующем формате:

```

"name": name, "phone": people_info[name].get("phone", "N/A"),
"address": people_info[name].get("address", "N/A"),
"parents": people_info[name].get("parents", "N/A"), "in_frame": True,
"last_seen": time.time()

```

После чего приложение попросит ввести имя, адрес, телефон и родителей человека, попавшего в обзор камеры. Данные сразу записываются в файл «People_info.json». Одновременно с записью данных о человеке в файл, камера делает 10 фото и добавляет их в папку «Known_faces». После заполнения всех полей с информацией о человеке и сделанных фото с камеры, приложение обновляет информацию и при повторном попадании того же человека в объектив камеры о нём уже появляется информация. Пример работы с занесённым в базу ранее неизвестного человека приведён на рисунке 4.

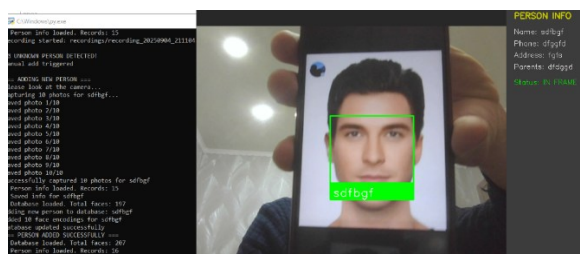


Рисунок 4 – Пример работы программы с занесением данных о человеке

В результате работы удалось создать функционал для камеры видеонаблюдения, которая способна определять человека по

изображению. Камера выполняет все поставленные задачи, и имеет возможность добавления новых функций таких как подключения к интернету и поиска информации из всемирной сети. Помимо этого, существует возможность вынесения информации о людях в локальную сеть, что позволяет сохранять данные не в приложении, а в отдельной сети.

Библиографический список:

1. Маккаллоу У., Питтс У. Логическое исчисление идей, заложенных в нервной деятельности // *Bulletin of Mathematical Biophysics*. 1943. Т. 5. С. 115–133.
2. Holmes W. et al. Ethics of AI in Education: Towards a Community-Wide Framework // *International Journal of Artificial Intelligence in Education*. — 2021. — Vol. 32. — P. 504–526.

DEVELOPMENT OF A PROGRAM SOLUTION FOR USER RECOGNITION

Aleksandrova N.R.

FSBEI HE Ulyanovsk SAU

Subaeva A.K., Minabutdinov R.I.

**FGBOU VO "Kazan National Research Technical University named
after A.N. Tupolev – KAI"
Chistopol branch "Vostok"**

Keywords: *video surveillance camera, control, video surveillance system, recognition.*

The article presents the development of a software solution for user recognition. This system is planned for checkpoints.