УДК 004.032.26

# ЗАЩИТА ДАННЫХ С ПОМОЩЬЮ РАСПОЗНАВАНИЯ ЛИЦ НЕЙРОННОЙ СЕТЬЮ

**Лазуренко Александр Витальевич**
Белгородская область, г.Белгород, ГБОУ «Белгородский инженерный юношеский лицей-интернат», 10 класс
*Научные руководители: Соболевская Маргарита Владимировна, г. Белгород, ГБОУ «Белгородский инженерный юношеский лицей-интернат», руководитель НОУ «Открой себя»; Волков Денис Александрович, г. Белгород, ГБОУ «Белгородский инженерный юношеский лицей-интернат», учитель информатики*

**Аннотация.** Предотвратить попытку взлома персонального устройства на расстоянии не так сложно, если человек знаком с основными способами защиты от вирусов и соблюдает осторожность во время пребывания в сети. Но как же повысить уровень защиты, когда злоумышленник пытается получить доступ к личным данным напрямую? Для решения этой проблемы я решил использовать систему распознавания лиц. Идея заключается в использовании лица пользователя в качестве пароля для дешифровки файлов. На языке программирования Python я написал приложение для операционной системы Windows, способное распознавать лица и шифровать данные на компьютере.
**Ключевые слова:** Безопасность данных, Распознавание лиц, Компьютерное зрение, Шифрование, Файлы, Нейронная сеть.

## Введение

Проблема защиты персональных данных всегда остается актуальной. Существует множество способов сохранить безопасность файлов. Например, защита с помощью распознавания лиц. Такая система защиты данных используется в смартфонах для разблокировки экрана. Всем известная система Face ID установлина на смартфонах от Apple. Но на компьютерах подобные системы не получили широкого распространения. Существуют приложения с функциями распознавания лиц, например, Veri Face от Lenovo, Windows Hello от Microsoft, но они редко используются. Я решил использовать распознавание лиц не для блокировки устройства, а для шифрования отдельных файлов.

## Основная часть

Для достижения цели сперва я решил изучить системы распознавания лиц. В настоящее время широкое распространение получили нейронные сети. В том числе были обучены модели для распознавания лиц. С помощью литературы [1] я изучил, каким образом происходит глубокое обучение нейронных сетей для распознавания лиц. На основе исследований было принято использовать модель точечного распознавания лиц. Для работы я выбрал обученную нейронную сеть dlib_face_recognition_resnet_model_v1, которая позволяет определить полученное лицо с точностью 99,38%. С помощью программы, написанной Андреем Созыкиным [приложение 1] я определил, как использовать функции этой модели с помощью языка программирования Python [2, 3, 4]. В качестве среды программирования я выбрал PyCharm. Первой задачей стало написание функции сравнения лиц. За основу я также использовал программу Андрея Созыкина. Следующей задачей стало подключение к веб камере и сохранение фотографии, если перед камерой появляется чье-то лицо. Для ее решения я использовал библиотеку компьютерного зрения Open CV. С помощью функций этой библиотеки можно найти лица на фотографии и даже в потоке кадров. С помощью автоматического метода определяется наличие свободной веб камеры и ее подключение. Когда программа замечает лицо в видеопотоке, то делает снимок экрана и отправляет его на проверку.

Но проблема в том, что полученное лицо сравнивается только с одной заранее вписанной в код фотографией, поэтому далее необходимо было создать Базу данных, где хранились бы данные о всех пользователях. Для создания таблицы я выбрал библиотеку Sqlite3. Таблица состоит из четырех столбцов: id пользователя, имя, краткое описание роли пользователя, для удобства редактирования базы данных, дескриптор лица в формате двоичного кода. Далее я написал метод для регистрации новых пользователей. Пользователь вводит свое имя и программа автоматически сканирует его лицо. Данные заносятся в таблицу и извлекаются для сравнения во время включения программы. После написания основных функций распознавания лиц я занялся созданием пользовательского интерфейса. Для этого я использовал библиотеку PyQt5 и программу Qt Designer. С помощью этой программы можно очень быстро настроить внешний вид приложения.

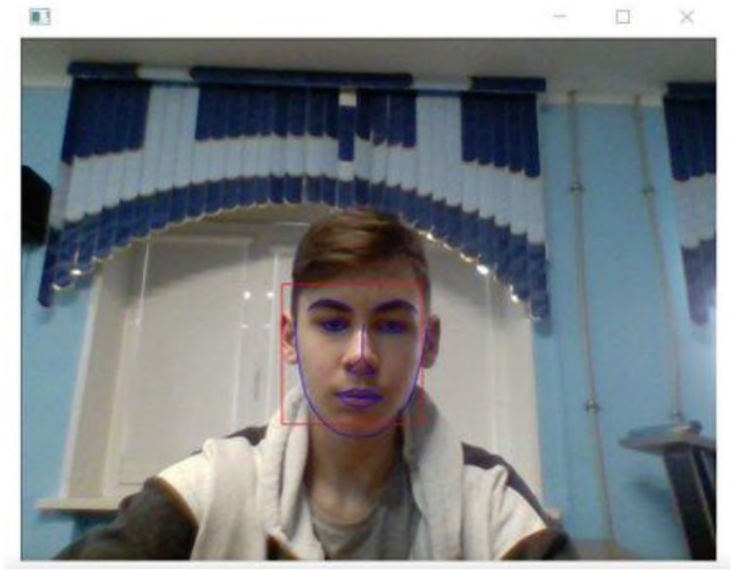

Рис. 1

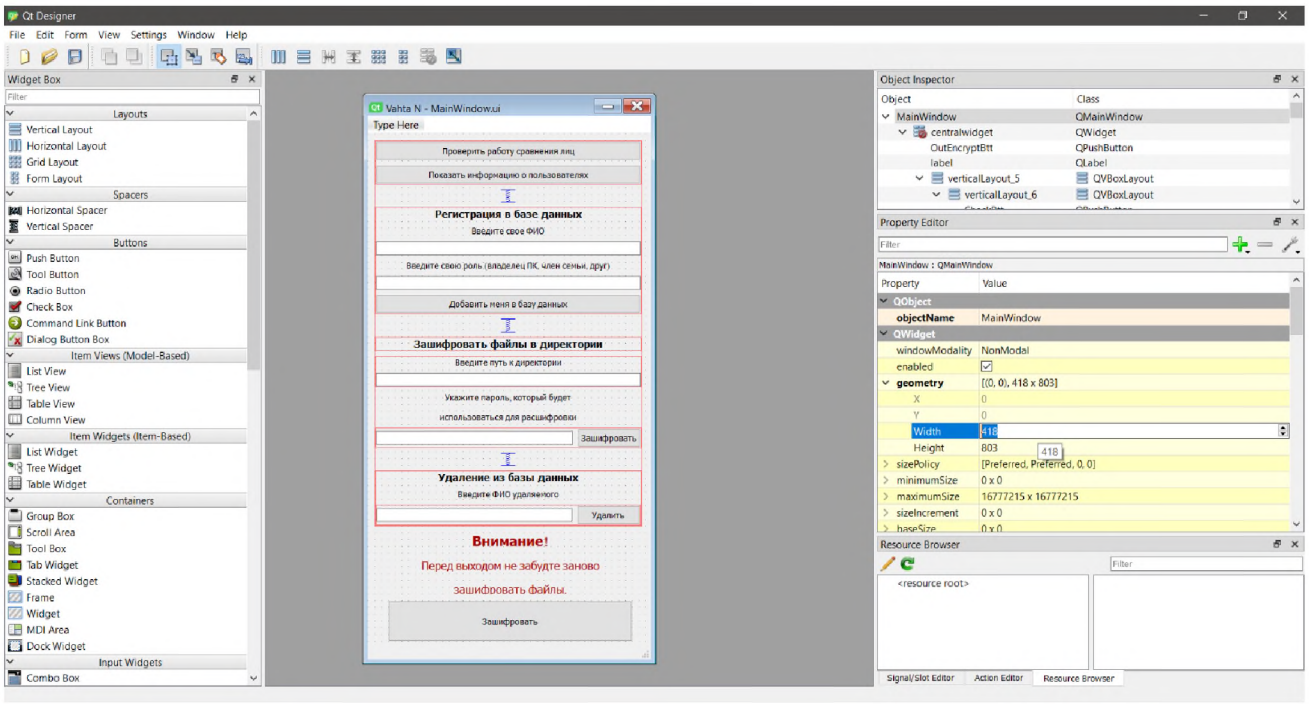


Рис. 2

Теперь пользователь может обращаться к методам программы с помощью удобного интерфейса. Далее я начал работу над системой шифрования файлов. Идея заключается в том, что доступ ко всем функциям приложения осуществляется через распознавание лиц. Открыть приложение могут только пользователи, чьи дескрипторы лица занесены в базу данных. С помощью приложения можно выбрать любой файл и зашифровать его специальным паролем. Пароль от файла сохраняется в специальном файле, который в свою очередь зашифрован надежным паролем. В отдельном поле в окне приложения составляется список из всех зашифрованных файлов. Чтобы открыть нужный файл, нужно просто кликнуть на его название. Далее программа дешифрует “хранилище” паролей, находит необходимый пароль для дешифровки нужного файла и открывает этот файл с помощью найденного пароля. Данные действия осуществляются с помощью функций библиотеки pyAesCrypt. С помощью данной функции можно с легкостью зашифровывать текстовые файлы, картинки, презентации, аудио и даже небольшие видео файлы в формат .crp. Алгоритм работы программы можно ознакомиться на рисунке ниже.

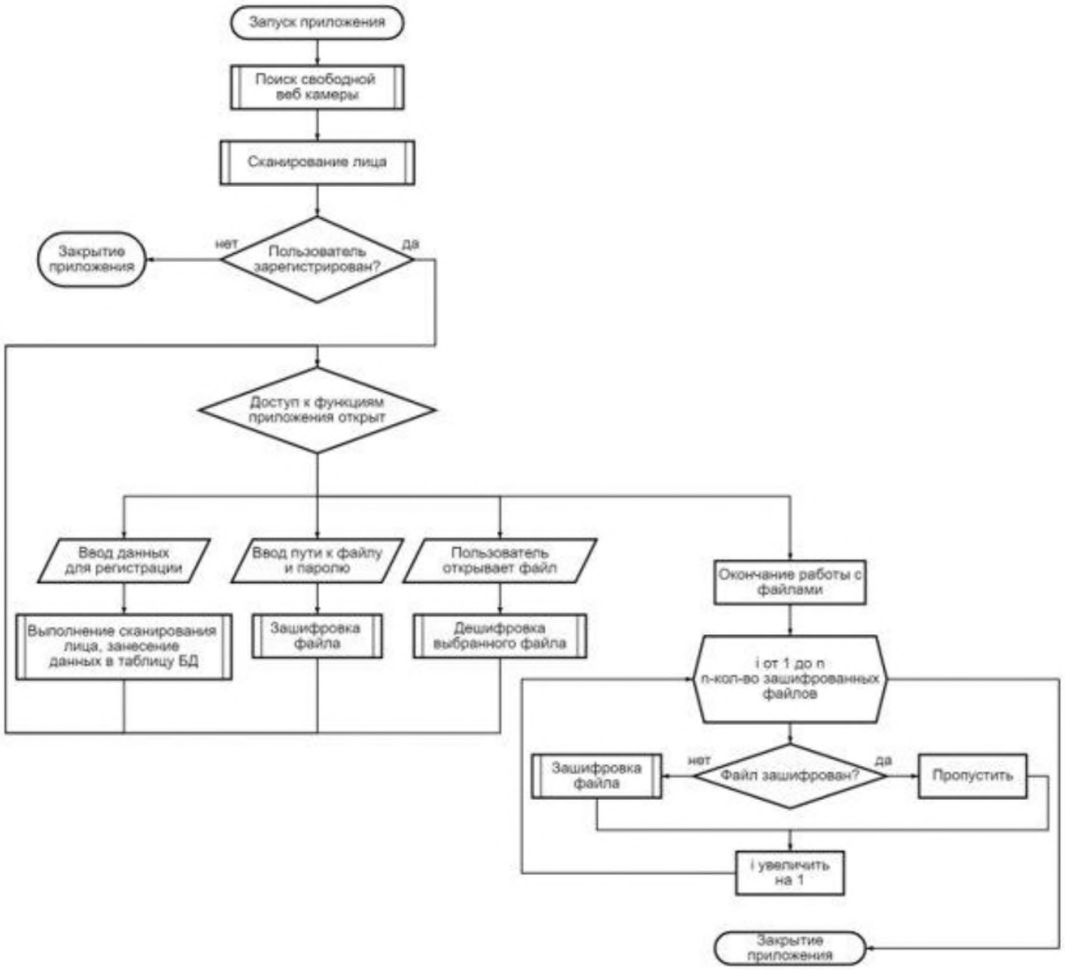


Рис. 3

**Выводы:**

В результате тестов приложения было выявлено, что приложение полностью справляется со своими функциями.

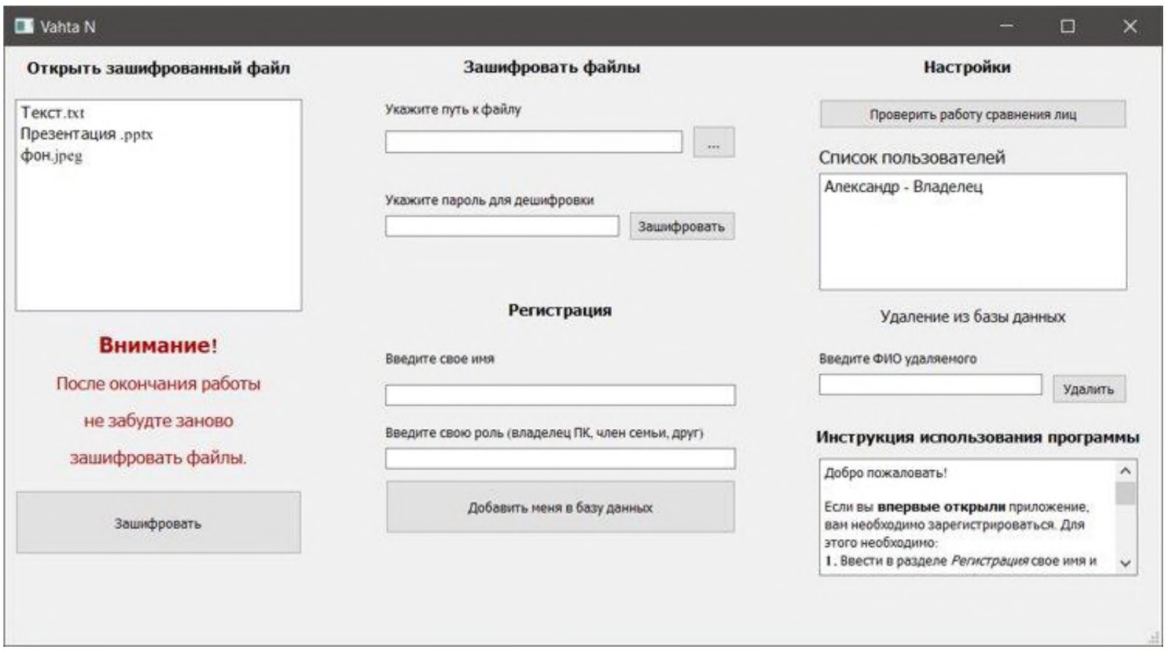


Рис. 4

Я обнаружил, что библиотека pyAesCrypt позволяет зашифровывать уже зашифрованный файл, что в теории повышает уровень защиты данных. Также были обнаружены минусы. Из-за большого веса модели нейронной сети увеличивается размер самого приложения и время, необходимое для запуска. Нейронная сеть не способна распознать лицо при низком или слишком высоком уровне освещения или лицо в маске.

**Заключение**

Разработанное приложение полезно использовать для защиты важных данных, но рядовому пользователю такие функции не обязательны для постоянного использования. Можно добавить синхронизацию “хранилищ” паролей между несколькими устройствами, чтобы пользователи могли обмениваться зашифрованными файлами и открывать их на своем компьютере.

Список литературы:

1. Pavel Pleskov. Машинное обучение это весело! Часть 4: Распознавание лиц с помощью глубокого обучения //medium.com. [Электронный ресурс]. Режим доступа : https://medium.com/@ppleskov/машинное-обучение-это-весело-часть-4-14931b3c912b
2. Howse J., Joshi P., Beyeler M. OpenCV: Computer Vision Projects with Python / Packt Publishing Ltd: October 2016. 539 c.
3. Мэтиз Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. СПб.: Питер, 2020. 512 с.
4. Хеллман Д. Стандартная библиотека Python 3: справочник с примерами: Пер. с англ. 2-е изд. СПб.: Диалектика, 2019. 1376 с.

**Приложения:**

Приложение 1:
https://github.com/sozykin/dlpython_course/blob/master/computer_vision/foto_comparison/foto_verification.ipynb