**Государственное бюджетное общеобразовательное учреждение города Москвы "Школа № 1564 имени Героя Советского Союза А.П. Белобородова"**

**Интегрированная система входа на портал ГосУслуг для распознавания лиц Vega**

Участники проекта :

Михедов Константин Константинович, учащийся 10А класса ГБОУ школы №1564

Мехтиев Владимир Денисович, учащийся 10А класса ГБОУ школы №1564

Руководитель проекта :

**Москва, 2021**

# Содержание

[Введение 3](#_Toc1520087647)

[Актуальность работы 3](#_Toc404240286)

[Причина выбора темы 3](#_Toc2053957234)

[Потенциальное применение 4](#_Toc580465453)

[Анализ текущих решений 4](#_Toc482271106)

[Цель и задачи 4](#_Toc609462404)

[Задачи 4](#_Toc289922465)

[Система управления базой данных 5](#_Toc2145028673)

[Распознавание биометрических данных 6](#_Toc428845959)

[Разработка серверной части системы 7](#_Toc673348231)

[Создание клиентского приложения 8](#_Toc1922199359)

[Заключение 9](#_Toc929308083)

[Список литературы 10](#_Toc132475011)

# Введение

## Актуальность работы

В современном мире людям часто приходится проходить аутентификацию, подтверждать свою личность. Для этой задачи используются специальные символьные последовательности - пароли, но к сожалению, единственное преимущество этого метода - его простота. Данный способ подтверждения личности обладает рядом существенных недостатков: пользователь может как просто забыть пароль, так и использовать слишком простую последовательность, подобрать которую не составит труда.

Гораздо более актуальными и быстро развивающимися являются способы авторизации, основанные на сборе и анализе биометрических и физических данных пользователя, таких как отпечаток пальца, скан сетчатки глаза, тембр голоса или снимок лица. Данные методы просты и удобны в использовании: они не требуют запоминания каких-либо символьных последовательностей, вероятность совпадения эксплуатируемых для входа данных у разных людей очень мала.

## Причина выбора темы

При использовании большинства описанных методов требуется специальное оборудование: для сканирования отпечатка пальца необходим дактилоскоп, для сбора информации о сетчатке глаза - специальный сканер. Это накладывает определенные ограничения на использование таких систем.

Но существуют также и менее требовательные к аппаратной части методы идентификации, например - установление личности, основываясь на фотографии лица пользователя. Для использования такого метода требуется лишь Web-камера, которую можно найти в каждом смартфоне или ноутбуке.

Использование таких систем очень удобно в тех случаях, когда утечка пользовательских данных является крайне опасной, например в системах, связанных с документами, и для избежания кражи информации, сессия завершается каждый определенный промежуток времени. Таким образом требуется регулярно вводить пароль для продолжения работы, что не очень удобно. Идентификация при помощи сканирования снимка лица пользователя значительно упростила и ускорила бы этот процесс.

## Потенциальное применение

Системы подтверждения личности при помощи биометрических и физических данных пользователя имеют широчайший спектр применений. С их помощью можно не только предоставлять способ получения доступа к конфиденциальной информации, но и осуществлять контрольно-пропускной режим на объектах государственной важности, идентифицировать человека в таких учреждинях, как поликлиники и библиотеки.

## Анализ текущих решений

На данный момент не существует каких-либо централизованных систем для выполнения такого действия, как идентификация пользователя по фотоснимку его лица, но некоторые крупные предприятия используют это в своих продуктах.

Компания Apple предоставляет функцию FaceID, позволяющую идентифицировать владельца устройства, работающего на базе IOS или MacOS. Такой же тенденции придерживается Microsoft, внедряя в свою ОС подсистему Windows Hello. К сожалению, данные функции заточены под строго определенные устройства и цели, что не позволяет использовать их каждому желающему.

## Цель и задачи

Цель: разработать систему, позволяющую входить в электронный дневник школьника, расположенный по адресу https://dnevnik.mos.ru/, идентифицируя пользователя при помощи фотографии его лица.

Задачи:

1. Спроектировать базу данных, содержащую в себе авторизационные данные пользователей, и наполнить её необходимой информацией
2. Обучить нейронную сеть, которая будет идентифицировать пользователя
3. Разработать серверную инфраструктуру авторизационной системы
4. Создать конечное клиентское приложение

# Система управления базой данных

Для аутентификации на сайте электронного дневника требуются некоторые пользовательские данные, такие как логин и пароль. Так как система разрабатывается для массового использования - данные будут храниться в специальной базе.

Она будет основываться на нереляционной БД - MariaDB (ответвление от MySQL), защищена паролем и находиться на одном хосте с серверной частью системы. Данные будут распределены по следующим ключам:

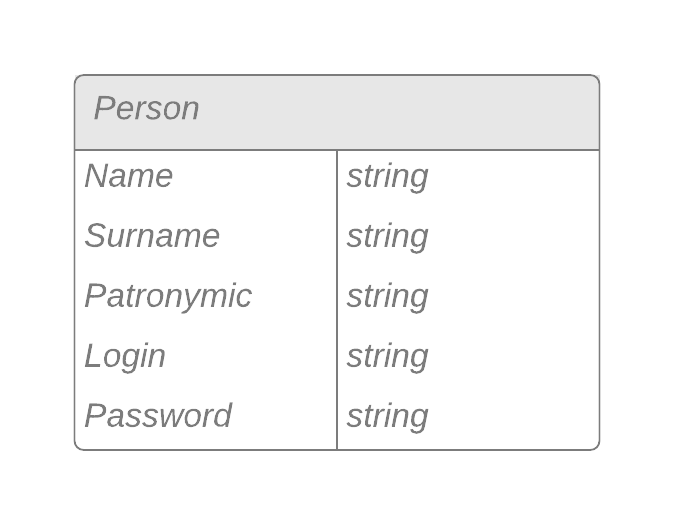


Рисунок 1. Ключи базы данных

Все пользовательские данные будут храниться в явном виде. Доступ к ним будет осуществляться при помощи библиотеки для языка программирования Python3 - PonyORM. Она предоставляет удобный и быстрый API для работы с MariaDB. Наполнение базы данных будет осуществляться администратором, так как данная система нацелена на учащихся школ, ВУЗов и других учебных заведений.

# Распознавание биометрических данных

Процесс установления личности происходит благодаря обработке биометрических данных пользователя при помощи нейронной сети. Для полноценной работы системы используется заранее обученная модель, имеющая следующие характеристики: 50 и более эпох (в зависимости от количества лиц в нейронной сети), размер подаваемого на вход пакета данных - 16, коэффициент обучения - 0.001, формат входного изображения - черно-белое с разрешением 244\*244px.

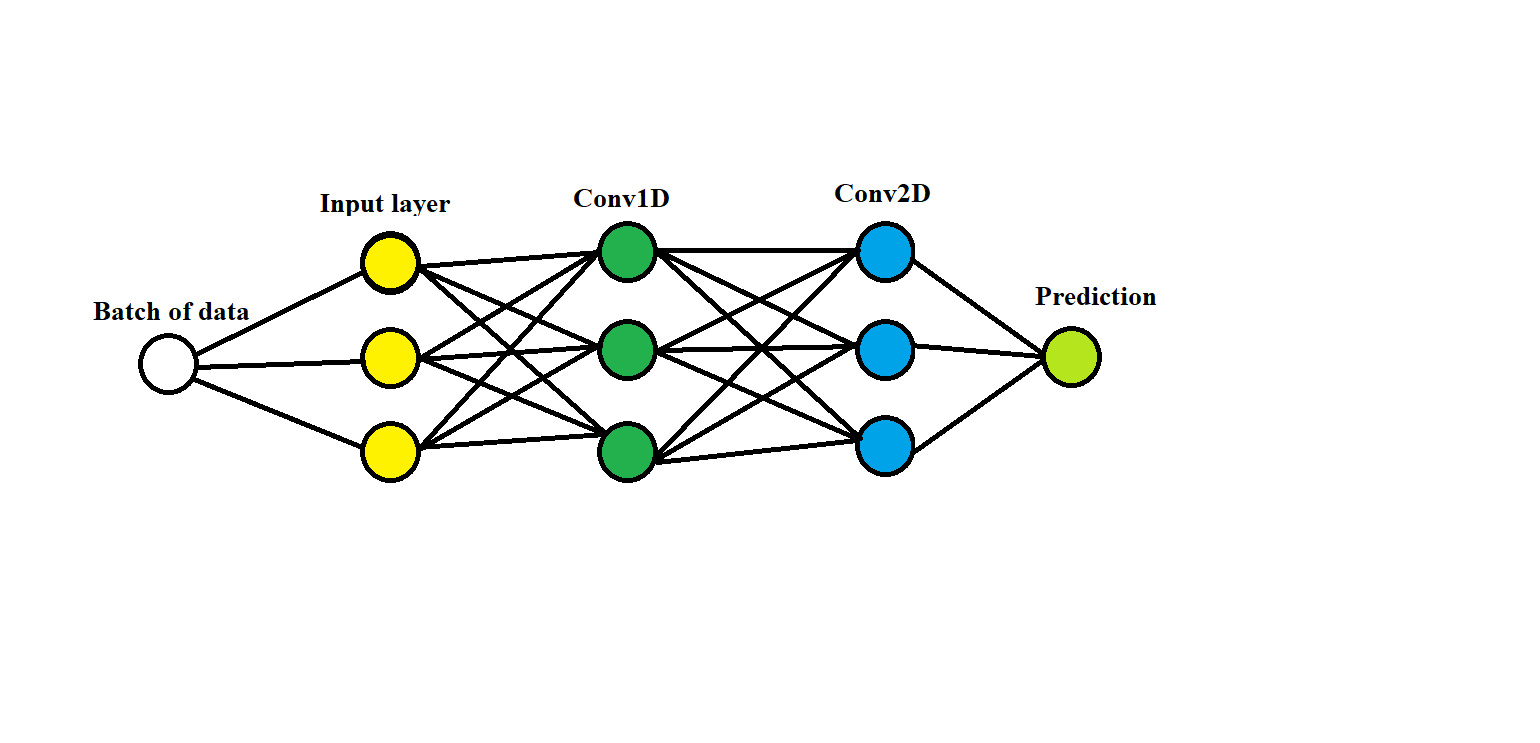


Рисунок 2. Структура нейронной сети

Обучение нейронной сети происходит при помощи инструмента Teachable Machine, предоставляемого компанией Google. Для качественного распознавания требуется минимум 10 исходных фотографий для каждого класса (пользователя).

При настройке исходной системы необходимо стремиться к уменьшению количества ложно-положительных срабатываний (False/Positive). Добиться этого можно на этапе обучения нейронной сети при помощи точечной регулировки параметров модели.

# Разработка серверной части системы

Серверная часть данной системы написана на высокоуровневом языке программирования - Python3 и выполняет следующие функции : получает от клиента фотографию пользователя, обращается к нейронной сети и базе данных, высылает ответ.

Реализация приёма и передачи авторизационных данных пользователя выполнена на основе библиотеки Flask. Отправка изображения на сервер выполняется при помощи POST запроса по пути ‘/verify’. Ответ посылается текстом в формате JSON и имеет следующие поля: status, name, surname, patronymic, login и password. Ключ status может иметь одно из двух значений: ‘ok’ или ‘error’ в зависимости от успешности верификации.

Полученное изображение обрабатывается нейронной сетью при помощи Keras. Для этого оно локально сохраняется и преобразуется к необходимому для работы формату. Если вероятность совпадения личности менее 95, то пользователю будет отказано в доступе и предложено попробовать пройти идентификацию ещё раз или обратиться к администратору.

После работы нейронной сети определяется фамилия пользователя, по которой далее осуществляется поиск в базе данных. Если поиск завершится удачно, то клиенту будет отправлен ответ с необходимыми для авторизации данными.

# Создание клиентского приложения

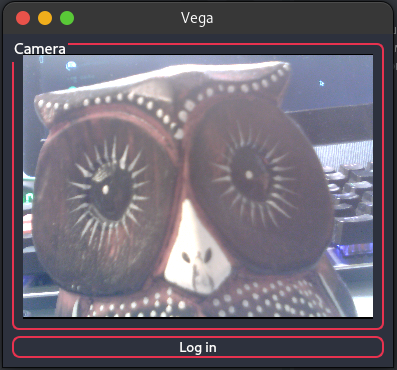
Клиентское приложение также написано с использованием Python3. Для построения пользовательского интерфейса был использован порт библиотеки QT5 для необходимого языка программирования.

Приложение имеет свой стиль и ресурсы в виде иконок. Для того, чтобы их было удобно использовать, они сразу включены в исходный код проекта.

При включении приложения выполняется первичная инициализация и проверка наличия камеры. Если таковой не найдётся, то будет выведено сообщение об ошибке. Работа с камерой реализована встроенными в QT средствами - QCamera, QCameraViewFinder и т.д.

Отправка изображения на сервер происходит при помощи встроенного в Python модуля Requests. Адрес сервера хранится в специальном конфигурационном файле в формате JSON, находящемся в папке с приложением.

При получении ответа будет выведено сообщение об ошибке или произвед`н вход в электронный дневник. Авторизация на сайте выполняется при помощи инструмента Selenium, позволяющего вводить необходимые значения логина и пароля прямо в браузере. На данный момент имеется поддержка Google Chrome и Firefox, но также при небольшой модификации можно добиться работоспособности со следующими парсерами HTML: Internet Explorer, Opera, Safari.



Изображение 3. Клиентское приложение

# Заключение

При выполнении данной работы была создана полностью работоспособная система авторизации на сайте электронного дневника (https://dnevnik.mos.ru/) при помощи физических данных пользователя.

Данная система состоит из нескольких составных модулей : клиентское и серверные приложения, база данных и нейронная сеть. Она легко поддаётся масштабированию при незначительных изменениях в исходном коде.

При незначительных доработках её можно использовать для авторизации на большинстве Интернет-ресурсов и в других системах.

Исходный код системы находится на GitHub: [ссылка](https://github.com/KonstantIMP/vega)

# Список литературы

1. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition. O’Reilly Media, 2019, .-856 p.
2. MariaDB technical docs. URL: <https://mariadb.org/documentation/> (дата обращения: 20.02.2021).
3. PonyORM API reference. URL: <https://docs.ponyorm.org/api_reference.html> (дата обращения: 20.02.2021).
4. Flask Official Docs. URL: <https://flask.palletsprojects.com/en/1.1.x/> (дата обращения: 20.02.2021).
5. Метрики в задачах машинного обучения, Open Data Science. URL: <https://habr.com/ru/company/ods/blog/328372/> (дата обращения: 20.02.2021).