УДК 004.5
DOI: 10.15827/0236-235X.134.245-256
Дата подачи статьи: 26.10.20
2021. Т. 34. № 2. С. 245–256

Разработка системы контроля доступа на основе распознавания лиц

С.А. Антипова [1], *к.ф.-м.н., старший научный сотрудник, samiraspb11@gmail.com*

[1] *Военная академия материально-технического обеспечения им. генерала армии А.В. Хрулева, г. Санкт-Петербург, 199034, Россия*

Основой данной работы является разработанная система контроля управления доступом на основе распознавания лиц. Программа состоит из четырех независимых компонентов: детекции лиц в видеопотоке, распознавания конкретного лица на основе сверточной нейронной сети, контроля открытием двери/турникета и клиентского веб-сервиса. Каждый модуль работает в своем процессе, поэтому при развитии проекта можно перенести каждый из них на отдельный сервер. Для обеспечения высокой скорости и качества системы использовалась гибкая методология разработки (agile-подход).

Ключевой особенностью работы является обоснованное применение технологии глубокого обучения на примере созданного макета программного средства с использованием технологии «одновыстрельного» обучения, или сиамских сетей, реализованных с помощью фреймворка PyTorch. В качестве предобученной нейросети использовалась MobileNetV3. Сиамские сети удобны для применения, так как нет необходимости в формировании огромного датасета с данными, что особенно важно для распознавания лиц. Архитектура таких сетей состоит из двух одинаковых нейросетей, имеющих одинаковый вес и структуру, а результаты их работы передаются в одну функцию активации – таким образом, определяется одинаковость входных данных (оценка сходства) на основе сравнения значений двух векторов.

Система протестирована на контрольно-пропускном пункте предприятия с учетом биометрических данных сотрудников, на которых нейросеть была обучена, и показала высокую точность при идентификации лиц. Предложенная сервис-ориентированная архитектура позволяет масштабировать систему аутентификации и верификации горизонтально и вертикально. При необходимости компоненты системы физически можно размещать на разных серверах, увеличивая пропускную способность системы в целом.

Ключевые слова: *система контроля и управления доступом, глубокое обучение, сверточные нейронные сети, распознавание лиц.*

Современный уровень развития систем распознавания лиц делает возможным использование технологии глубокого обучения в целом ряде программных продуктов: в системах контроля доступа, авторизации и др. *Системы контроля и управления доступом* (СКУД) на предприятиях позволяют автоматизировать процесс контроля пропуска сотрудников на объекты, анализировать время их нахождения там. Автоматизированный контроль позволяет не только устанавливать правила доступа индивидуально для каждого сотрудника, но и производить мониторинг деятельности каждого сотрудника в том или ином помещении. В БД сохраняется информация о том, когда пришел или ушел сотрудник, его должность и какое количество времени он провел на объекте. Такие системы позволяют сократить затраты на ручной контроль, а также время самого процесса прохода через входной турникет или дверь из-за отсутствия необходимости фиксации данных сотрудника и времени входа/выхода в бумажном журнале с альтернативным учетом в электронном журнале регистрации. Помимо этого, у специалиста по безопасности всегда имеется актуальная информация о нахождении на предприятии каждого сотрудника. При необходимости возможна немедленная блокировка пропуска.

Типичная СКУД [1], как правило, построена на основе сети контроллеров, подключаемых к компьютеру, и включает:

– устройства преграждающие управляемые (турникеты, двери, оборудованные управляемыми замками, ворота, шлагбаумы);

– устройства ввода идентификационных признаков (считыватели);

– электронные микропроцессорные модули, реализующие аутентификацию объектов, логику авторизации для доступа в те или иные помещения;

– ПО, позволяющее осуществлять централизованное управление контроллерами с персонального компьютера, формирование отче-

тов, разнообразные дополнительные функции;

– конверторы (интерфейсы) среды для подключения аппаратных модулей СКУД друг к другу и к персональному компьютеру;

– вспомогательное оборудование (камеры, блоки питания, маршрутизаторы и др.).

Среди биометрических идентификаторов наиболее привлекательны системы, использующие не менее двух методов идентификации с включением биометрии по идентификации пользователя по лицу. Среди вещественных идентификаторов следует уделить внимание бесконтактным идентификаторам на базе RFID с обязательной защитой информации от копирования и несанкционированной перезаписи как дополнительному инструменту к распознаванию лиц.

На сегодняшний день на российском рынке существует большое количество решений СКУД на основе распознавания лиц [2–4]. Тем не менее, большая часть информации об используемых архитектурных решениях, а также алгоритмах распознавания скрыта, то есть является коммерческой тайной – неким черным ящиком. В то же время существует небольшое количество отрытых работ в российских журналах, посвященных полной реализации СКУД с использованием биометрических данных. Как правило, в них освещаются вопросы непосредственной модификации существующих алгоритмов глубокого обучения, а в качестве источника данных довольно часто выступает ImageNet – открытая БД размеченных изображений [5].

Большой популярностью у зарубежных разработчиков, специализирующихся на системах распознавания лиц, пользуется фреймворк глубокого обучения TensorFlow с высокоуровневым API-интерфейсом Keras [6, 7]. Тем не менее, на взгляд автора настоящей статьи, намного проще использовать PyTorch из-за более оптимизированного и целенаправленного подхода к глубокому обучению и дифференциальному программированию. PyTorch использует динамические вычислительные графы, а TensorFlow – статические. Динамические вычислительные графы не требуют компиляции перед каждым выполнением, поэтому можно спокойно изменять входные данные в процессе работы для изучения различных результатов. Помимо этого, уменьшается время на проведение множества экспериментов.

На выбор подходящей архитектуры всей программы в целом повлияло множество факторов. Исходя из функциональных требований клиент должен быть «тонким», то есть на стороне клиента никакое ПО не должно быть установлено. В данном случае для взаимодействия с пользователем был выбран http-протокол. В этом случае клиент подключается к веб-серверу и получает необходимые данные от сервера. Помимо html-страниц, клиенту требуется получать видеопоток от IP-камеры для детектирования лиц.

Количество кадров с камеры определяется аппаратными характеристиками самой камеры, а также скоростью передачи данных от камеры к компьютеру. Если учитывать, что в охранной системе для камеры выделяется индивидуальный канал, то скорость работы камеры не будет зависеть от загруженности линии (ethernet) и определяется аппаратными характеристиками камеры.

Одной из основных особенностей работы с видеопотоком по сравнению с отдельными изображениями является объем данных – даже просто 10 секунд видео по объему равны примерно 200 картинкам.

В системах видеоаналитики требуется минимальная задержка на принятие решения, значит, система должна уметь обрабатывать 20 и более кадров в секунду. Однако в системах видеонаблюдения зачастую используются десятки, сотни, а иногда и тысячи камер. Даже при поиске по видеоархиву неразумно заново обрабатывать видео при каждом запросе, так как отклик на запрос будет долгим, а значит, вся разметка кадров должна появиться в нем вместе с видео, что опять-таки приводит к требованию работы алгоритмов в реальном времени.

Таким образом, к вычислительным ресурсам предъявляются высокие требования, что зачастую приводит к экономической нецелесообразности использования многих решений, которые хорошо работают с отдельными изображениями.

Для решения проблем вычислительных ресурсов обычно используют следующие подходы:

– нейросетевые архитектуры, способные работать в режиме реального времени [8–10];

– предварительная фильтрация видеопотока с использованием как простого прореживания кадров, так и классических алгоритмов, реализованных в библиотеке OpenCV (например, определение оптического потока, фона и т.п.) [11].

В данном проекте был применен модифицированный алгоритм Виолы–Джонса, позволяю-

щий обнаруживать объекты в режиме реального времени.

Вместе с тем клиенты могут подключаться к серверу и по-разному его загружать. Конечно, в данном охранном комплексе каждому клиенту выделяется своя линия связи, но скорость подключения может варьироваться от разных факторов, а в случае сбоя веб-сервера камера должна работать независимо.

Распознавание лиц конкретных людей происходит с меньшей скоростью, чем работа камеры. Стоит учитывать, что распознавание необходимо выполнять только в случае обнаружения одного лица.

Управление доступом двери/турникета должно выполняться независимо от каждой части приложений. Открытие и закрытие двери управляются микроконтроллером.

В итоге получается, что все части программы должны работать независимо. Для независимой работы подпрограмм существуют два подхода для их распараллеливания: многопоточность и распараллеливание процессов. Многопоточные программы работают в едином адресном пространстве. Такой программе требуется синхронизация при работе с разделяемыми ресурсами. Для этого используются различные примитивы синхронизации: мьютексы, семафоры, атомарные переменные и события. Однако в языке Python, использованном для реализации проекта, не существует истинной параллельности из-за способа синхронизации потоков Global Interpreter Lock (GIL). Преимуществом многопроцессорности является то, что подпрограммы работают в своем адресном пространстве, а недостатком – необходимость синхронизации между процессами.

Существует множество механизмов синхронизации процессов – через файл, отображенный на оперативную память, через разделяемую память, через именованные и неименованные каналы в оперативной памяти. Однако у всех этих механизмов синхронизации существует определенный недостаток: при необходимости перенести физически процесс на другую машину представленные механизмы синхронизации процессов работать не будут. Самым оптимальным решением является использование сокетов [12]. Сокеты позволяют обмениваться данными между процессами на одной или разных машинах. Единственное условие – компьютеры должны быть объединены сетью.

На основе предварительного анализа в данной работе был выбран способ взаимодействия между процессами через компьютерные сети. Система реализована с помощью языка программирования Python 3.7. Основными фреймворками, использованными в работе, явились следующие свободные пакеты: PyTorch – библиотека глубокого обучения, OpenCV 3.4.11 – для обработки изображений, Django 3.0 – для разработки веб-приложений.

Архитектура приложения

В разработанном комплексе (рис. 1) существуют четыре модуля, взаимодействующих между собой через http-протокол. Каждый модуль работает в своем процессе, поэтому при развитии проекта можно перенести его на отдельный сервер, потребуется только изменить IP-адрес и/или номер порта конфигурационных файлов модулей. Первый модуль осуществляет управление камерой и обнаружение лица на экране. Второй модуль определяет конкретного человека на переданной модулем фотографии и возвращает номер пользователя или информацию о том, что данный пользователь не существует. Третий модуль управляет дверью. Он работает на отдельном микроконтроллере, открывающем и закрывающем дверь/турникет. Четвертый модуль – это веб-сервер для обеспечения взаимодействия с клиентом.

Модули включают в себя три компонента (FaceDataTypes, Interfaces и Realization) и связаны между собой через абстрактные классы таким образом, что можно изменять реализацию без изменения их логики. Компонент FaceDataTypes содержит определения необходимых типов классов данных. В компоненте Interfaces определены интерфейсы к классам, общие типы данных, а в Realization реализованы конкретные классы.

Общие для всех модулей типы данных определены в FaceDataTypes (рис. 2). Главным общим классом данных для всех модулей является Faces, в котором хранятся текущий кадр с камеры, кадр с нарисованными боксами на найденных лицах, список с областями, которые находятся на текущем кадре. Классы данных предназначены для инкапсуляции данных и наследования, поэтому поля оставляются открытыми.

Компонент Realization

В данной работе компоненты системы реализованы с помощью сторонних библиотек, а также собственных классов (рис. 3).

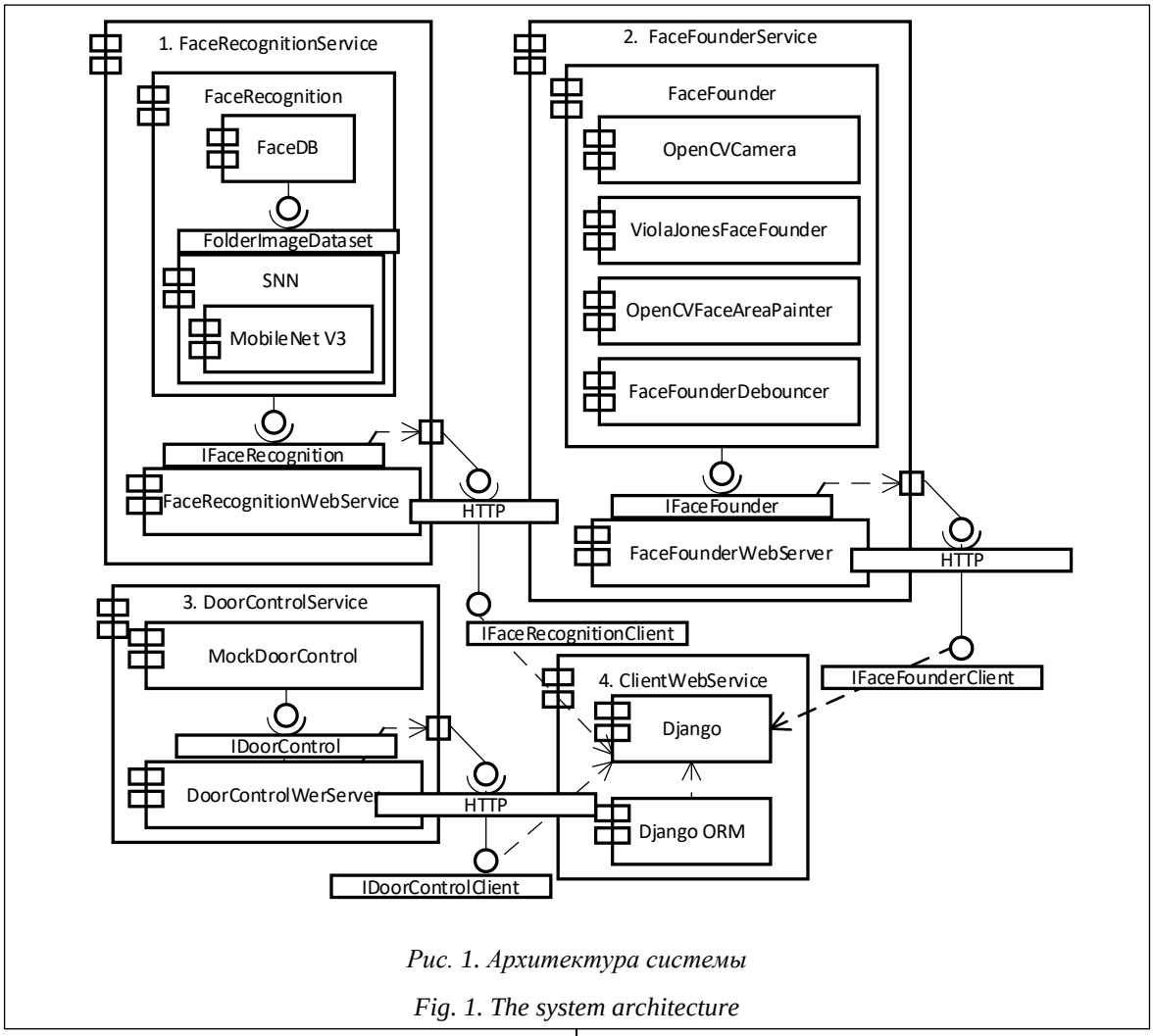

Рис. 1. Архитектура системы

Fig. 1. The system architecture

Абстрактный класс ICamera реализован с помощью класса OpenCVCamera. В инициализаторе данного класса создается объект cv2.VideoCapture. Входным аргументом является уникальный номер камеры (camera_id), который передается через инициализатор класса OpenCVCamera. При создании объекта включается камера, которая работает до завершения времени жизни объекта, то есть при вызове метода OpenCVCamera._del_.

Поиск лиц на кадре производится с помощью каскадного классификатора Хаара, реализованного в библиотеке OpenCV. В библиотеке есть готовый класс cv2.CascadeClassifier для детектирования объектов в видеопотоке. Этому объекту требуется указать путь к файлу в формате XML с необходимыми параметрами для детектора. Для метода Виолы–Джонса используется файл с параметрами классификатора для детектирования лиц в кадре видеопотока. Используется стандартный метод скользящего окна, но на каждом шаге выбирается область изображения, на котором производится классификация. При достижении конца изображения его размер уменьшается на определенное число, которое определяется параметром scale_factor. По умолчанию данный параметр равен 1.1. После уменьшения поиск продолжается с новым масштабом.

При детектировании возникают ложные срабатывания, поэтому для улучшения качества распознавания используют принцип соседства. Если в данной области данный объект обнаружен определенное количество раз, считается, что объект найден. Параметр называется min_neighbors и по умолчанию равен 3.

Параметры min_size и max_size определяют минимальный и максимальный размеры детектируемого объекта. По умолчанию ограничений нет.

С помощью класса-обертки ViolaJonesFaceFounder реализован поиск лиц на изображении. Рисование боксов на изображении реализуется также с помощью библиотеки OpenCV в класс

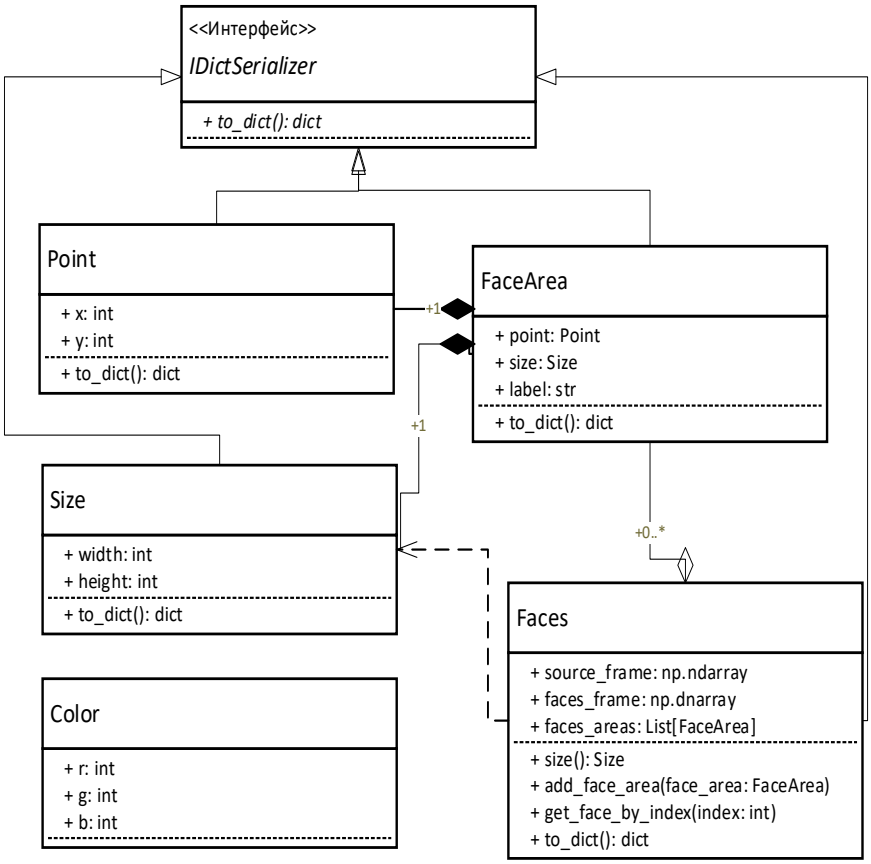

Рис. 2. UML-диаграмма классов в FaceDataTypes

Fig. 2. The UML class diagram in FaceDataTypes

OpenCVFaceAreaPainter. Входными параметрами являются цвет бокса FaceDataTypes.Color и ширина линии line_width.

Классификация лиц на изображении реализуется с помощью класса FaceSNNDetector.

Компонент детектирования лиц

Клиенты подключаются к серверу FaceFounderServer детектирования лиц и получают ответ в виде сериализованного объекта Faces. Для упрощения работы клиентов с сервером разработан класс-интерфейс IFaceFounderClient. Клиент работает с ним как с объектом, но в классе-наследнике происходят подключение к серверу детектирования лиц, выполнение запроса к серверу, десериализация необходимых данных. В результате клиент не зависит от реализации формы запроса к серверу и получает готовый объект Faces.

Разработанная реализация HTTPFaceFounderClient выполняет синхронный http-запрос к серверу, в качестве ответа получает сериализованный библиотекой pickle объект Faces. Данный класс выполняет синхронные запросы, поэтому скорость работы сервера детектирования лиц фиксированная и клиенту выдаются уже подготовленные к передаче бинарные данные. В этом случае нецелесообразно выполнять асинхронные запросы.

Данный компонент состоит из двух частей: детектирования лиц с камеры и сервиса, предоставляющего клиентам текущее обнаруженное лицо. Предполагается, что скорость работы камеры и клиентов разная. Клиент может в любой момент запросить объект Faces, а камера и детектор лиц работают с постоянной частотой кадров в секунду. Развязка скорости работы клиента и камеры выполняется с помощью технологии двойной буферизации. В момент перерисовки кадра и поиска лиц на новом кадре клиенту отправляется предыдущий кадр. Запуск камеры занимает некоторое время, которое может быть больше времени запуска сервера. В случае успешного запуска сервера и неготовности камеры клиенту отправляется

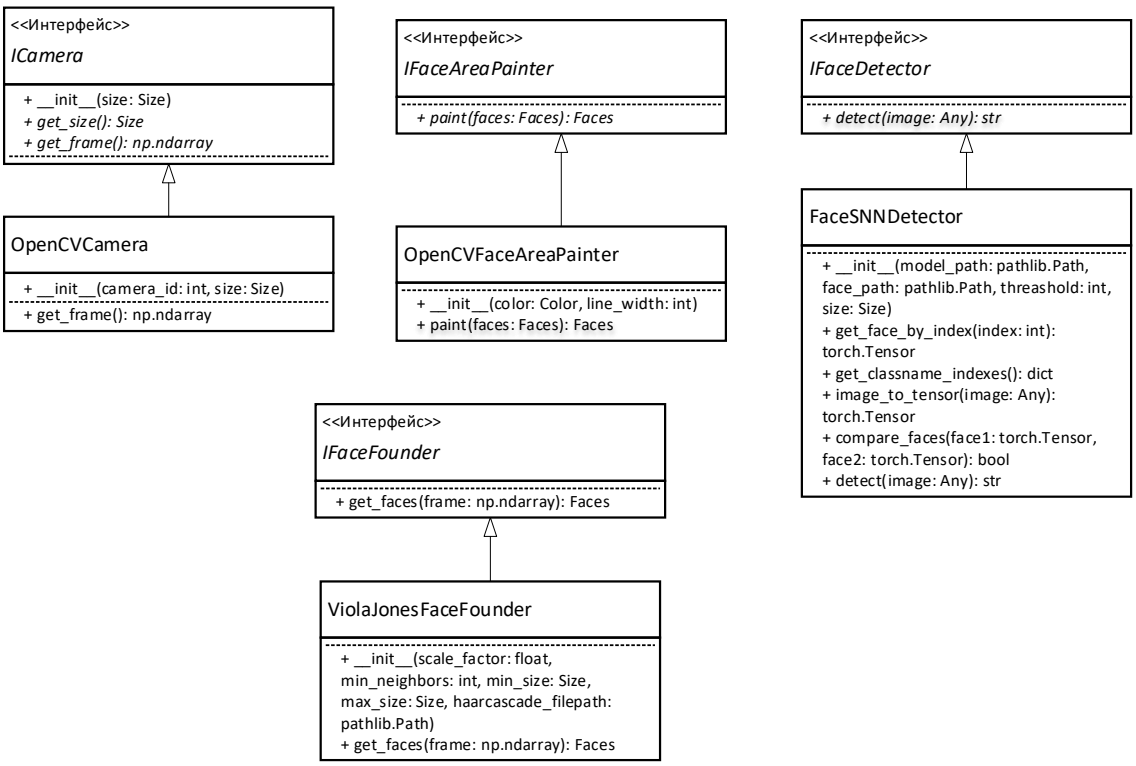

Рис. 3. Диаграмма классов в компоненте Realization

Fig. 3. The class diagram in the Realization component

пустой объект Faces с кадром по умолчанию, который загружается с диска. Данная двойная буферизация также позволяет быстро передавать данные к нескольким клиентам, если это требуется. Клиенты получают готовый сериализованный объект Faces.

Первоначальное тестирование данной реализации сервера детектирования лиц показало некорректный поиск лиц в видеопотоке. Было обнаружено, что даже если лицо находится перед камерой, алгоритм поиска лиц на каждом третьем, четвертом или пятом кадрах мог выдавать, что лицо отсутствует. Данную проблему удалось решить добавлением счетчика, у которого есть минимальное и максимальное значения. Если количество лиц на кадре не изменилось, счетчик увеличивается на единицу. Если количество лиц изменилось (или стало равным нулю) в текущем кадре, счетчик уменьшается на единицу. Если значение счетчика стало равным нулю или достигло максимального значения, записывается новое значение объекта Faces в двойном буфере и происходит смена текущего значения буфера. Минимальное значение счетчика равно нулю, максимальное значение можно конфигурировать. Максимальное значение счетчика необходимо подбирать таким образом, чтобы не возникало ложных срабатываний при приемлемой скорости смены объекта Faces. В результате было установлено максимальное значение счетчика, равное четырем.

Компонент распознавания лиц

Распознавание лиц выполняется на отдельном сервере. Клиент отправляет запрос в виде изображения по протоколу HTTP. На этом изображении должно быть только лицо человека, которого необходимо обнаружить. Сервер возвращает уникальный идентификатор пользователя (UserId).

Клиент взаимодействует с сервером через интерфейс IFaceRecognitionClient. Основные типы данных клиента представлены на рисунке 4. Интерфейс этого класса разработан таким образом, чтобы класс-наследник мог выполняться в отдельном потоке с сервером распознавания лица.

Процедура взаимодействия клиента с сервером происходит следующим образом. Клиент выполняет запрос к серверу с помощью метода

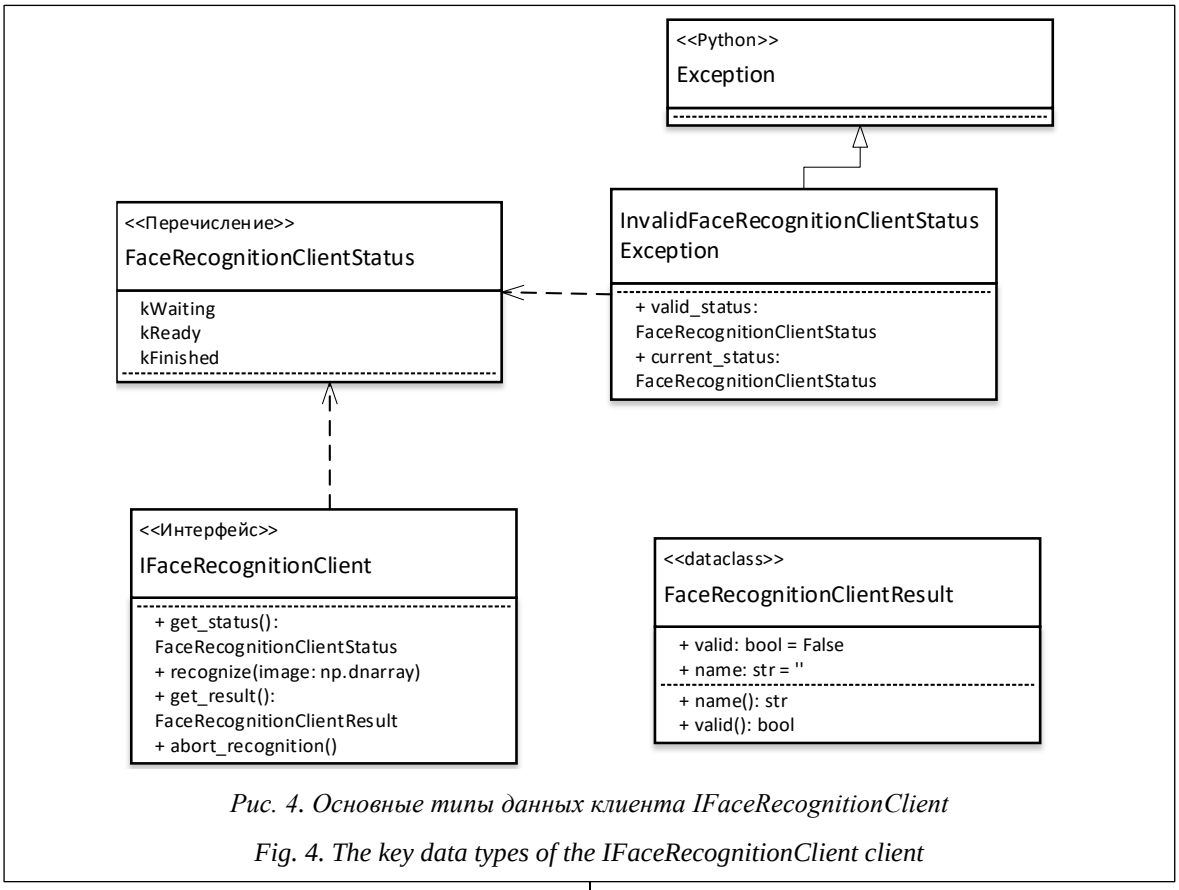

Рис. 4. Основные типы данных клиента IFaceRecognitionClient

Fig. 4. The key data types of the IFaceRecognitionClient client

IFaceRecognitionClient.recognize, а затем проверяет текущее состояние запроса с помощью метода IFaceRecognitionClient.get_status. Данный метод возвращает перечисление FaceRecognitionClientStatus. Если текущее состояние не FaceRecognitionClientStatus.kReady, метод IFaceRecognitionClient.recognize вызывает исключение IFaceRecognitionClient.InvalidFaceRecognitionClientStatusException. Вызывая метод IFaceRecognitionClient.get_result, клиент получает результат распознавания лица на переданном изображении в виде FaceRecognitionClient.FaceRecognitionClientResult. В случае изменения формата запроса или протокола к серверу пользователю предоставляется новый клиент, поэтому процедура взаимодействия объекта с клиентом унифицирована.

За последние годы глубокие нейронные сети доказали свою высочайшую эффективность для решения задач визуального распознавания образов и существенно развились. Немаловажную роль в развитии этих технологий также сыграли существенное увеличение количества обучающих данных изображений и разработка новых методов обучения и архитектур для нейронных сетей. В частности, прогресс в области глубоких нейронных сетей затронул и алгоритмы распознавания лиц. Технологии распознавания лиц базируются на очень глубоких сверточных нейронных сетях, которые для каждого изображения лица вычисляют уникальный биометрический шаблон, представляющий собой вектор чисел, обычно называемый дескриптором лица. Сравнивая биометрические шаблоны, полученные из двух изображений лиц, компьютер может вынести предположение о том, принадлежат ли они одному человеку или нет, тем самым решая задачу биометрической верификации на основе изображения лица.

На основании сравнения биометрических шаблонов компьютер выдает некоторую меру схожести, позволяя искать фотографии человека по базе, имея некоторую фотографию-запрос, выбирая фото из базы по максимальной мере схожести с запросом, тем самым решая задачу биометрической идентификации по изображению лица.

Современные системы распознавания лиц, как правило, включают в себя следующие этапы обработки входного изображения:

– локализация лица на фотографии (детектирование);

– нахождение ключевых точек лица;

– выравнивание лица;

– извлечение биометрического шаблона (дескриптора);

– сравнение биометрических шаблонов.

Зачастую некоторые этапы могут быть выполнены в рамках одной системы. Например, в одном из первых эффективных нейросетевых детекторов MTCNN (Multi-task Cascaded Convolutional Neural Network) [13] этапы детектирования лица и нахождения его ключевых точек объединены в одну сеть. В основе лежит каскад из трех нейронных сетей, последовательно применяемых к изображению, приведенному к разным масштабам (пирамиде изображений). Первая сеть каскада (P-Net) генерирует множество регионов, в которых потенциально может находиться лицо. Вторая сеть (R-Net) необходима для корректировки предсказаний регионов первой сети. Третья сеть (O-Net) окончательно корректирует предсказания координат области, в которой находится лицо, формируя итоговое предсказание координат лица и одновременно с этим предсказывая положение его пяти ключевых точек.

Для обеспечения достаточной полноты детектирования лиц малого размера в архитектуре MTCNN используется пирамида изображений, что существенно увеличивает вычислительные затраты. Это может быть критично, например, при развертывании системы распознавания лиц на камерах низкого разрешения в реальном времени. Кроме того, каскад из недостаточно глубоких и широких нейронных сетей формирует признаки, недостаточно информативные для нахождения сложных лиц с перекрытиями и большой вариацией в освещении.

Один из способов решения проблемы детектирования сложных лиц реализован в архитектуре S^3FD (Single Shot Scale-invariant Face Detector) [14]. В отличие от MTCNN, который работает с пирамидой изображений, S^3FD принимает на вход одно изображение одного масштаба, эффективно обрабатывая пространственное смещение между областями за счет использования глубокой сверточной нейронной сети на входном изображении. Основная идея этой архитектуры заключается в том, что предсказания координат лица строятся сразу на разных масштабах карт признаков, позволяя таким образом находить лица даже самого малого размера. Кроме того, за счет большей информативности признаков S^3FD лучше работает на лицах, сложных для детектирования.

Вместе с тем основная задача при проектировании системы заключалась в том, чтобы иметь возможность обучать нейросеть с учетом только одной исходной фотографии на каждого сотрудника, хранящейся в БД, без использования сложных алгоритмов. Таким образом, сравниваются два изображения и определяется степень их схожести. Отсюда сиамские сети, или обучение одним выстрелом [15]: строится непрямой классификатор, затем оцениваются сходства. Для этого используется архитектура сиамской сети (рис. 5), когда входные изображения проходят через две сети, но сети фактически одинаковые: та же архитектура, тот же вес, на самом деле это та же сеть, но использу-

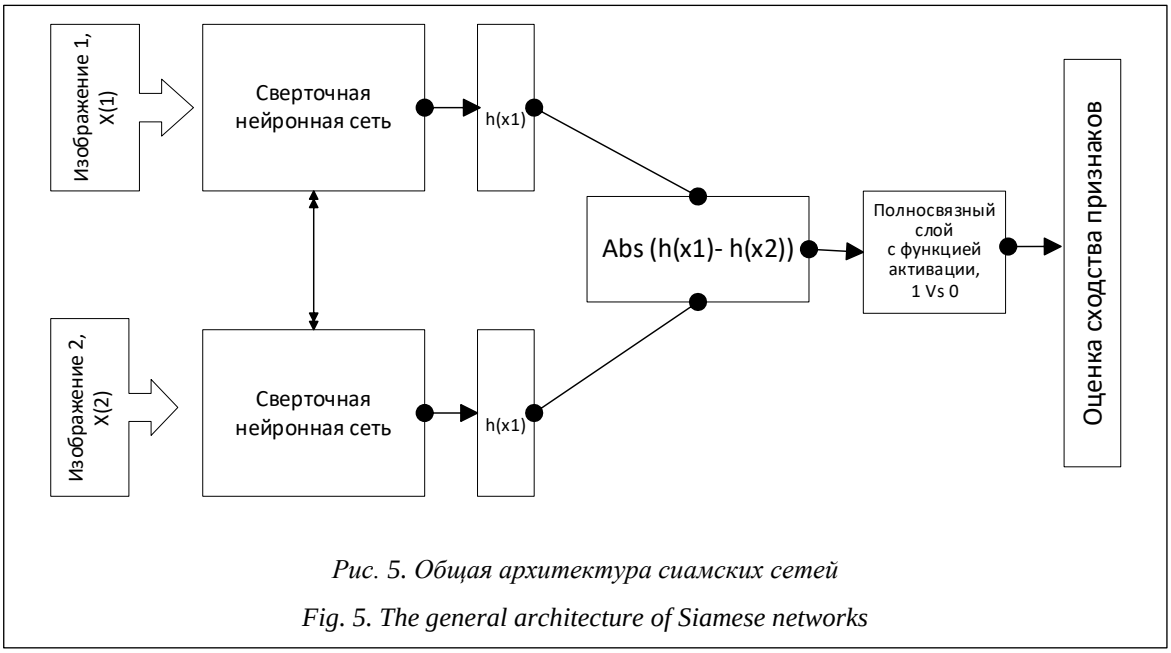

Рис. 5. Общая архитектура сиамских сетей

Fig. 5. The general architecture of Siamese networks

ется для двух разных входов. Выходы затем сравниваются, чтобы решить, являются ли изображения подобными.

Компонент распознавания конкретного лица получает необходимую область лица на кадре в формате двумерного массива, а возвращает специальный тип, содержащий результат успешного или неудачного распознавания. В случае успешного распознавания конкретного лица также можно получить идентификатор пользователя. Клиентский веб-сервис предоставляет видеопоток с определением лиц в этом видеопотоке в виде нарисованных прямоугольников. В зависимости от количества обнаруженных лиц в видеопотоке клиенту выводятся разные сообщения. Если никто не обнаружен, постоянно выводится «Подойдите к камере». Если обнаружено одно лицо, выводится сообщение «Лицо обнаружено, идет распознавание» (рис. 6). Если обнаружены два и более лиц, выводится сообщение «Перед камерой должен находиться только один человек». При обнаружении только одного лица запускается процедура распознавания конкретного лица с точностью не менее 90 %. Охранник на посту после успешного определения конкретного сотрудника на веб-странице нажимает на кнопку «Открыть дверь». Сервису управления дверью передается команда на открывание двери или турникета.

В целях уменьшения временных затрат и повышения качества исследований для создания модели глубокого обучения применялась также и предварительно обученная нейронная сеть (MobilenetV3) [16, 17] с использованием метода трансферного обучения (Transfer Learning) на основе двух приемов: выделение признаков (feature extractor) из новых образцов, которые затем пропускаются через новый классификатор, обучаемый с нуля, а также дообучение (fine-tuning, «тонкая настройка»), суть которого заключается в размораживании нескольких верхних слоев захардкоженной модели, использованной для выделения признаков, и в совместном обучении вновь добавленной части модели (в данном случае полносвязного классификатора) и этих верхних слоев. Проще говоря, у сети сбрасываются веса последнего слоя, сеть дообучается на своих данных, веса остальных слоев либо замораживаются, либо совсем немного обновляются. Этот прием называется дообучением, поскольку немного корректирует наиболее абстрактные представления в повторно используемой модели, чтобы сделать их более актуальными для данной задачи, и позволяет увеличить точность обучения.

Исходный код системы находится в приватном GitHub-репозитории https://github.com/mto-in-progress/facedetection, доступ к которому в данный момент возможен по запросу, но в ближайшее время будет открыт для просмотра и использования.

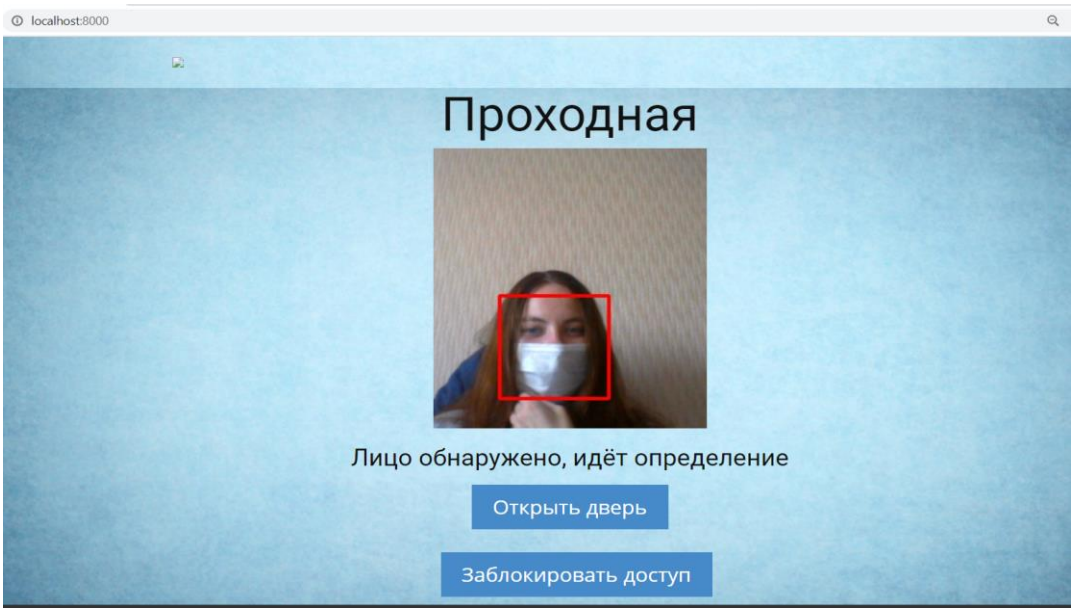

Рис. 6. Общий вид веб-страницы оператора (охранника)

Fig. 6. General view of the operator's (security guard's) web page

Заключение

Разработана система контроля управления доступом на основе распознавания лиц, состоящая из четырех независимых компонентов: детекции лиц, распознавания конкретного лица, контроля открытия двери и клиентского веб-сервиса. Использованная слабосвязанная архитектура позволяет масштабировать систему аутентификации горизонтально и вертикально. При необходимости компоненты системы физически можно размещать на разных серверах, увеличивая пропускную способность системы в целом. Взаимозаменяемые компоненты программы можно независимо друг от друга улучшать при дальнейшей модернизации проекта. Разработан сервис для обнаружения и идентификации лица человека в видеопотоке, а также реализованы сервис для идентификации конкретного человека по лицу и клиентский сервис для взаимодействия с пользователем (охранник, администратор).

Проанализирован быстрый и алгоритмически эффективный подход к построению сверточной нейронной сети, а также реализована адаптация уже предобученных сетей под задачу распознавания (MobileNetV3). Ключевой особенностью сиамских сетей является отсутствие необходимости формирования огромного датасета с данными, что особенно важно для распознавания лиц в масштабе большой организации.

С целью уменьшения временных затрат и повышения качества исследований для создания модели глубокого обучения использовались, в том числе, предварительно обученные нейронные сети с применением метода трансферного обучения на основе двух приемов: выделения признаков, а также дообучения. Точность идентификации лиц для нейросети достигла 90 % после 30 эпох.

Преимущество данной системы заключается в простоте разработки и внедрения. Система является альтернативным вариантом коммерческих систем контроля и управления доступом на объекты с использованием технологии распознавания лиц.

Литература

1. Михайлов А., Колосков А., Дронов Ю. Комплексный подход при идентификации личности // Системы безопасности. 2015. Т. 4. С. 62–71.

2. Система контроля и управления доступом «Интегра-СКД». URL: https://www.integra-s.com/sistema-kontrolia-dostupa/ (дата обращения: 10.10.2020).

3. Познакомьтесь с алгоритмом. URL: https://findface.pro/technology/ (дата обращения: 10.10.2020).

4. Система контроля и управления доступом – IP СКД (СКУД) IDmatic. URL: http://idmatic.ru/idmatic-technologii (дата обращения: 10.10.2020).

5. Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z. et al. ImageNet large scale visual recognition challenge. IJCV, 2015, vol. 115, no. 3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

6. Франсуа Ш. Глубокое обучение на Python; [пер. с англ.]. СПб: Питер, 2018. 400 с.

7. Maras M.H., Alexandrou A. Determining authenticity of video evidence in the age of artificial intelligence and in the wake of deepfake videos. The Intern. Journal of Evidence and Proof, 2019, vol. 23, no. 3, pp. 255–262. DOI: 10.1177/1365712718807226.

8. Никитин М.Ю., Конушин В.С., Конушин А.С. Нейросетевая модель распознавания человека по лицу в видеопоследовательности с оценкой полезности кадров // Компьютерная оптика. 2017. Т. 41. № 5. С. 732–742. DOI: 10.18287/2412-6179-2017-41-5-732-742.

9. Ranftl A., Alonso-Fernandez F., Karlsson S., Bigun J. Real-time AdaBoost cascade face tracker based on likelihood map and optical flow. Biometrics IET, 2017, vol. 6, no. 6, pp. 468–477. DOI: 10.1049/iet-bmt.2016.0202.

10. Визильтер Ю.В., Горбацевич В.С., Моисеенко А.С. Одноэтапный детектор лиц и особых точек на цифровых изображениях // Компьютерная оптика. 2020. Т. 44. № 4. С. 589–595. DOI: 10.18287/2412-6179-CO-674.

11. Guobo X., Wen L. Image edge detection based on opencv. IJEEE, 2013, vol. 1, no. 2, pp. 104–106. DOI: 10.12720/IJEEE.1.2.104-106.

12. Грамотная клиент-серверная архитектура: как правильно проектировать и разрабатывать web API. URL: https://tproger.ru/articles/web-api/ (дата обращения: 10.10.2020).

13. Zhang K., Zhang Z., Li Z. et al. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett., 2016, vol. 23, no. 10, pp. 1499–1503. DOI: 10.1109/LSP.2016.2603342.

14. Zhang S., Zhu X., Lei Z., Shi H., Wang X., Li S.Z. S3FD: Single shot scale-invariant face detector. Proc. IEEE ICCV, 2017. URL: https://arxiv.org/pdf/1708.05237.pdf (дата обращения: 10.10.2020). DOI: 10.1109/iccv.2017.30.

15. Dey S., Dutta A., Toledo J.I., Ghosh S., Lladós J., Pal U. SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification. URL: https://arxiv.org/pdf/1707.02131v2.pdf (дата обращения: 10.10.2020).

16. Howard A., Sandler M., Chen B., Wang W., Chen L.-C., Tan M., Chu G. et al. Searching for MobileNetV3. Proc. IEEE ICCV, 2019. URL: https://arxiv.org/pdf/1905.02244.pdf (дата обращения: 10.10.2020). DOI: 10.1109/iccv.2019.00140.

17. Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017. URL: https://arxiv.org/pdf/1704.04861.pdf (дата обращения: 10.10.2020).

Software & Systems Received 26.10.20
DOI: 10.15827/0236-235X.134.245-256 2021, vol. 34, no. 2, pp. 245–256

The access control system development based on face recognition

S.A. Antipova [1], *Ph.D. (Physics and Mathematics), Senior Researcher, samiraspb11@gmail.com*

[1] *General of the Army A.V. Khrulyov Military Academy of Logistics, St. Petersburg, 199034, Russian Federation*

Abstract. The basis of this work is the developed access control system based on face recognition. The program comprises four independent components: face detection in a video stream, face recognition based on a convolutional neural network, door/turnstile opening control, and a client web service. Each module works in its own process, so during the development of the project, you can transfer each module to a separate server. The agile development methodology (agile approach) ensures high speed and system quality.

The key feature of the work is the substantiated application of deep learning technology on the example of the created model of the software tool using the technology of ″one-shot″ learning or siamese networks, implemented using the PyTorch framework. MobileNetV3 was used as a pre-trained neural network. Siamese networks are convenient for use from the point of view of the absence of the need to form a huge dataset with data, which is especially important for face recognition. The architecture of such networks comprises two identical neural networks with the same weight and structure, and the working results are transferred to one activation function - thus, the similarity of the input data (similarity assessment) is determined based on the comparison of the values of two vectors.

The enterprise's point of entry tested the system, considering the biometric data of employees, on which the neural network was trained. The system showed high accuracy in identifying individuals.

The proposed service-oriented architecture allows scaling the authentication and verification system horizontally and vertically. If necessary, system components can be physically placed on different servers, increasing the throughput of the system as a whole.

Keywords: access control system, deep learning, convolutional neural networks, face recognition.

References

1. Mikhaylov A., Koloskov A., Dronov Yu. An integrated approach to personal identification. *Security and Safety*, 2015, vol. 4, pp. 62–71 (in Russ.).

2. *Access Control System Integra-ACS*. Available at: https://www.integra-s.com/sistema-kontrolia-dostupa/ (accessed October 10, 2020) (in Russ.).

3. *Meet the Algorithm*. Available at: https://findface.pro/technology/ (accessed October 10, 2020) (in Russ.).

4. *Biometric Technologies IDmatic IP-SCD.* Available at: http://idmatic.ru/idmatic-technologii (accessed October 10, 2020) (in Russ.).

5. Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z. et al. ImageNet large scale visual recognition challenge. *IJCV*, 2015, vol. 115, no. 3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

6. Chollet F. *Deep Learning with Python*. 2018, 361 p. (Russ. ed.: St. Petersburg, 2018, 400 p.).

7. Maras M.H., Alexandrou A. Determining authenticity of video evidence in the age of artificial intelligence and in the wake of deepfake videos. *The Intern. Journal of Evidence and Proof*, 2019, vol. 23, no. 3, pp. 255–262. DOI: 10.1177/1365712718807226.

8. Nikitin M.Yu., Konushin V.S., Konushin A.S. Neural network model of human face recognition in a video sequence with an assessment of the usefulness of frames. *Computer Optics*, 2017, vol. 41, no. 5, pp. 732–742. DOI: 10.18287/2412-6179-2017-41-5-732-742 (in Russ.).

9. Ranftl A., Alonso-Fernandez F., Karlsson S., Bigun J. Real-time AdaBoost cascade face tracker based on likelihood map and optical flow. *Biometrics IET*, 2017, vol. 6, no. 6, pp. 468–477. DOI: 10.1049/iet-bmt. 2016.0202.

10. Vizilter Yu.V., Gorbatsevich V.S., Moiseenko A.S. Single-shot face and landmarks detector. *Computer Optics*, 2020, vol. 44, no. 4, pp. 589–595. DOI: 10.18287/2412-6179-CO-674 (in Russ.).

11. Guobo X., Wen L. Image edge detection based on opencv. *IJEEE,* 2013, vol. 1, no. 2, pp. 104–106. DOI: 10.12720/IJEEE.1.2.104-106.

12. *Competent Client-Server Architecture: How to Properly Design and Develop a Web API*. Available at: https://tproger.ru/articles/web-api/ (accessed October 10, 2020).

13. Zhang K., Zhang Z., Li Z. et al. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.*, 2016, vol. 23, no. 10, pp. 1499–1503. DOI: 10.1109/LSP.2016. 2603342.

14. Zhang S., Zhu X., Lei Z., Shi H., Wang X., Li S.Z. S3FD: Single shot scale-invariant face detector. *Proc. IEEE ICCV*, 2017. Available at: https://arxiv.org/pdf/1708.05237.pdf (accessed October 10, 2020). DOI: 10.1109/iccv.2017.30.

15. Dey S., Dutta A., Toledo J.I., Ghosh S., Lladós J., Pal U. *SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification*. Available at: https://arxiv.org/pdf/1707.02131v2.pdf (accessed October 10, 2020).

16. Howard A., Sandler M., Chen B., Wang W., Chen L.-C., Tan M., Chu G. et al. Searching for MobileNetV3. *Proc. IEEE ICCV*, 2019. Available at: https://arxiv.org/pdf/1905.02244.pdf (accessed October 10, 2020). DOI: 10.1109/iccv.2019.00140.

17. Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M. et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. Available at: https:// arxiv.org/pdf/1704.04861.pdf (accessed October 10, 2020).

Для цитирования

Антипова С.А. Разработка системы контроля доступа на основе распознавания лиц // Программные продукты и системы. 2021. Т. 34. № 2. С. 245–256. DOI: 10.15827/0236-235X.134. 245-256.

For citation

Antipova S.A. The access control system development based on face recognition. *Software & Systems*, 2021, vol. 34, no. 2, pp. 245–256 (in Russ.). DOI: 10.15827/0236-235X.134.245-256.