

## Разработка методов распознавания лиц для систем видеонаблюдения

Луцик Юрий Александрович, кандидат технических наук;

Жуковец Александр Николаевич, студент магистратуры

Белорусский государственный университет информатики и радиоэлектроники (г. Минск, Беларусь)

*Статья посвящена исследованию методов биометрической идентификации человека и описанию разработки системы видеонаблюдения.*

**Ключевые слова:** биометрическая идентификация человека, методики распознавания образов, разработка веб-приложения, разработка мобильного приложения.

Одной из актуальных тем в современном мире является задача распознавания лиц. Она актуальна как для области интеллектуальных сред, так и в области систем безопасности. Крупные компании разрабатывают свои собственные системы распознавания лиц, например, система FaceNet от Google, которая распознаёт людей с точностью 99,63%, искусственная нейронная сеть Facebook, показывающая результат в 97,5%. Среди продуктов с открытым исходным кодом можно выделить OpenCV. Это библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения. Может свободно использоваться в академических и коммерческих целях, распространяется в условиях лицензии BSD.

Для реализации программы распознавания лиц была выбрана одна из наиболее эффективных комбинация алгоритмов распознавания лиц — алгоритма оценки ориентиров (англ. Face Landmark Estimation, FLE) и гистограммы направленных градиентов (англ. Histogram of Oriented Gradients, HOG). Описывая общий алгоритм работы программы с использованием данной комбинации алгоритмов можно выделить следующие этапы:

1. Поиск всех лиц на фото.
2. Распознавание каждого лица, даже если оно странным образом повернуто, или если освещение плохое.
3. Определять уникальные черты лица, которые отличают одного человека от других, например, размер глаз, форма лица и так далее.
4. Сравнить выявленные уникальные особенности этого лица со всеми людьми, которых система уже знает, чтобы понять, кто изображен на фото.

Для поиска лиц на фотографиях используется алгоритм гистограммы направленных градиентов. Основной идеей алгоритма является допущение, что внешний вид и форма объекта на участке изображения могут быть описаны распределением градиентов интенсивности или направлением краев. Их реализация может быть произведена путём раз-

деления изображения на маленькие связанные области, именуемые ячейками, и расчетом для каждой ячейки гистограммы направлений градиентов или направлений краев для пикселей, находящихся внутри ячейки. Комбинация этих гистограмм называется дескриптором. Для увеличения точности локальные гистограммы подвергаются нормализации по контрасту. С этой целью вычисляется мера интенсивности на большем фрагменте изображения, который называется блоком, и полученное значение используется для нормализации. Нормализованные дескрипторы обладают лучшей инвариантностью по отношению к освещению.

Дескриптор гистограммы направленных градиентов имеет несколько преимуществ над другими дескрипторами. Поскольку он работает локально, метод поддерживает инвариантность геометрических и фотометрических преобразований, за исключением ориентации объекта. Подобные изменения появятся только в больших фрагментах изображения. Более того, как обнаружили Далал и Триггс, грубое разбиение пространства, точное вычисление направлений и сильная локальная фотометрическая нормализация позволяют игнорировать движения пешеходов, если они поддерживают вертикальное положение тела. Таким образом, дескриптор гистограммы направленных градиентов, является хорошим средством нахождения людей на изображениях. [10]

Первым шагом вычислений во многих детекторах особых точек является нормализация цвета и гамма-коррекция. По итогам которого получается черно-белое изображение более удобное для преобразования в карту градиентов.

Преобразование к карте градиента происходит путём анализа каждого пикселя изображения, на то, насколько тёмным он является по сравнению с окружающими его пикселями. После чего рисуется стрелка, показывающая в какую сторону изображение становится темнее. Такая стрелка называется градиентом. На рисунке 1 представлен визуальный пример работы алгоритма.

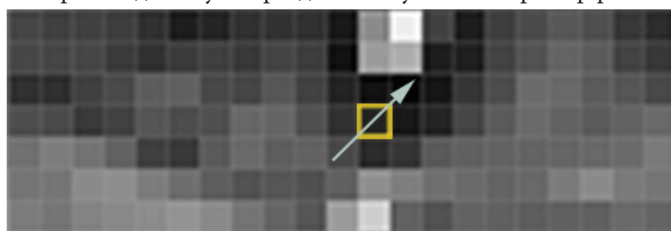


Рис. 1. Градиент пикселя

Преимущество метода направленных градиентов перед обычным анализом пикселей заключается в том, что вне зависимости от яркости исходного изображения будет получаться одинаковая карта градиента.

Однако сохранение градиента для каждого пикселя — это слишком много информации, которую сложно обработать. Эффективным способом решения такой проблемы является обработка основных направлений света на более высоком уровне.

Для этого необходимо разделить изображение на маленькие квадраты выбранного размера, например, 16

на 16 пикселей каждый. В каждом квадрате подсчитать, сколько точек градиента повернуто в каждом из основных направлений. Затем заменить этот квадрат на изображения стрелочками, направленными туда же, куда и большинство.

В конечном итоге исходное изображение будет преобразовано в карту градиентов, в котором ясно проглядывается базовая структура лица. На рисунке 2 представлен результат разбиения фотографии на карту градиентов с помощью сетки в 16 на 16 пикселей.



Рис 2. Результат разбиения фото на карту градиентов

Чтобы найти лицо на карте градиентов достаточно найти часть изображения, которая наиболее похожа на известную карту градиентов, полученную из мно-

жества других лиц в ходе обучения. На рисунке 3 представлен пример сравнения общей маски лица и обработанной.

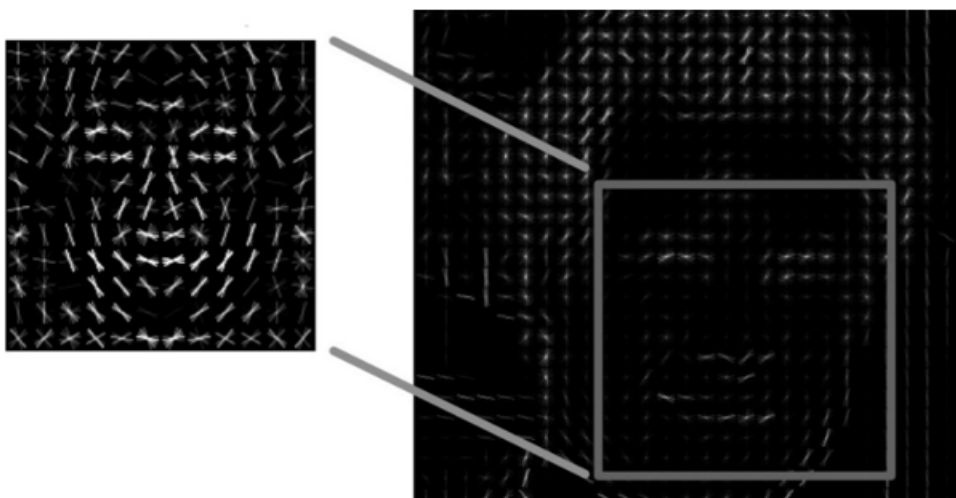


Рис. 3. Поиск лица на карте градиентов

После нахождения лица необходимо определить, в какую сторону оно повёрнуто, и привести его к общему виду для последующего анализа, ведь лицо, повёрнутое в разные стороны, это лицо одного и того же человека.

Самым распространённым методом приведения лица к общему виду является алгоритм оценки ориентиров.

Основная идея состоит в том, что мы отметим 68 особых точек (называемых ориентирами), которые существуют на каждом лице — верхняя часть подбородка, внешняя точка каждого глаза, внутренняя точка каждой брови и т.д. На рисунке 4 представлен общий вид расположения точек.



Рис. 4. Общий вид расположения точек

После определения точек необходимо изменить изображение так, чтобы глаза и рот были как можно лучше центрированы. Для таких преобразований используются аффинные преобразования, то есть такие преобразо-

вания, при которых все линии остаются параллельными вне зависимости от способа преобразования (искажение, поворот, масштабирование). На рисунке 5 представлен пример работы алгоритма.



Рис. 5. Пример работы преобразований

По итогам преобразования, вне зависимости от того, как повёрнуто лицо на исходном изображении, нейронная сеть получит на вход изображение с одинаковым положением основных частей лица человека.

Самый простой подход к распознаванию лица заключается в непосредственном сравнении неизвестного лица, обнаруженного на шаге 2, со всеми изображениями людей, которые уже были отмечены ранее. Если будет найдено ранее отмеченное лицо, которое очень похоже на распознаваемое лицо, то это наверняка один и тот же человек. Но у этого подхода есть большая проблема. Сравнение всех отмеченных ранее лиц с каждым новым загруженным изображением занимает слишком много времени, в то время как лица должны распознаваться за миллисекунды, а не за часы. Эффективным способом оптимизации сравнения является переход от сравнения всего изображения к сравнению отдельных измерений, например, размер ушей, расстояние между глазами и так далее.

Выбор частей необходимых для измерения можно возложить на нейронную сеть. Глубокое обучение, определяет, какие части лица нужно измерять, лучше, чем люди.

Решение заключается в создании сверточной нейронной сети глубокого обучения которая должна быть обучена создавать 128 измерений для каждого лица.

Во время обучения сети анализируется одновременно три лица:

1. Обучающее изображение лица известного человека.
2. Другая фотография того же известного человека.
3. Изображение совершенно другого человека.

Алгоритм просматривает измерения, которые он делает для каждого из этих трех изображений. Затем он немного настраивает нейронную сеть, чтобы удостовериться, что измерения, созданные для изображений 1 и 2, будут более похожи, а измерения для 2 и 3 — менее похожи:

Повторив этот этап миллионы раз для миллионов изображений тысяч разных людей, нейронная сеть учится надежно создавать 128 измерений для каждого человека. Любые десять разных изображений одного и того же человека должны давать примерно одинаковые измерения.

Полученные 128 измерений каждого лица называют картой. Идея преобразования массива необработанных

данных, например, изображения, в список генерируемых компьютером чисел крайне важна для машинного обучения.

Процесс обучения сверточной нейронной сети для получения карты лиц требует большого количества данных и мощного компьютера. Но как только сеть будет обучена, она сможет генерировать измерения для любого лица, даже для встреченного впервые.

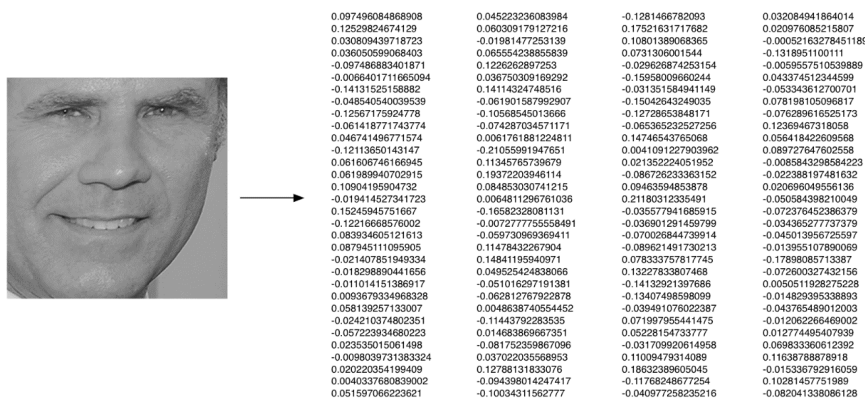


Рис. 6. Измерения для изображения

Последний этап заключается в поиске человека в базе данных, который ближе всего подходит под полученные 128 измерений исходного изображения.

Для поиска изображения необходимо обучить небольшую опорную сеть используя алгоритм классификации, например, метод опорных векторов.

Основная идея метода заключается в переводе исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с наибольшим зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей гиперплоскостью будет гиперплоскость, создающая наибольшее расстояние до двух параллельных гиперплоскостей. Алгоритм основан на допущении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора.

Проектируемая систем — система видеонаблюдения с возможностью распознавания лиц. Данная система представляет из себя мобильное и веб-приложение, которое позволяет в режиме реального времени осуществлять просмотр видеотрансляции из записывающего устройства. Проектируемая система видеонаблюдения позволяет пользователям вести трансляцию из видеоприбора, выполнять распознавание лиц, отправлять уведомления при обнаружении движения, сохранять изображение при фиксации движения, а также обучать модель по средствам добавления новых лиц через пользовательский интерфейс.

Проектируемая система разделена на три группы актеров: «Видеокамера», «Пользователь», «Сервер» (см. Рисунок 7). Каждая группа обладает определенным набором разрешенных действий.

Таким образом, все полученные изображения лиц необходимо пропустить через готовую обученную сеть, чтобы получить 128 измерений для каждого лица. На рисунке 6 представлены результаты работы сети для получения 128 измерений тестового изображения.

Актер «Видеокамера» обладает следующими возможностями:

1. Обнаружение движения;
2. Передача изображения на сервер;
3. Трансляция видеопотока.

Актер «Пользователь» обладает следующими возможностями:

1. Войти в систему;
2. Выйти из системы;
3. Зарегистрироваться в системе;
4. Удаление изображений в базе данных;
5. Просмотр изображений из базы данных;
6. Добавление записывающих устройств;
7. Удаление записывающих устройств;
8. Загрузка изображений для обучения системы.

Актер «Сервер» обладает следующими возможностями:

1. Обнаружение и распознавание лиц;
2. Отправка сообщения в Telegram-бот;
3. Изменение/просмотр базы данных;
4. Обучение системы.

Проектируемая система состоит из четырех компонентов (см. Рисунок 8):

1. Записывающее устройство;
2. Серверная часть приложения;
3. База данных;
4. Пользовательский интерфейс.

Модуль «устройство видеонаблюдения» транслирует видеопоток в режиме реального времени. В случае, если на изображении обнаружено движение формируется HTTP запрос к серверной части приложения. В серверной части на уровне контроллеров обрабатывается HTTP запрос, то есть выбирается нужная функция для работы с поступившим запросом. Затем слой контроллеров вызывает не-





Рис. 7. Характеристики системы

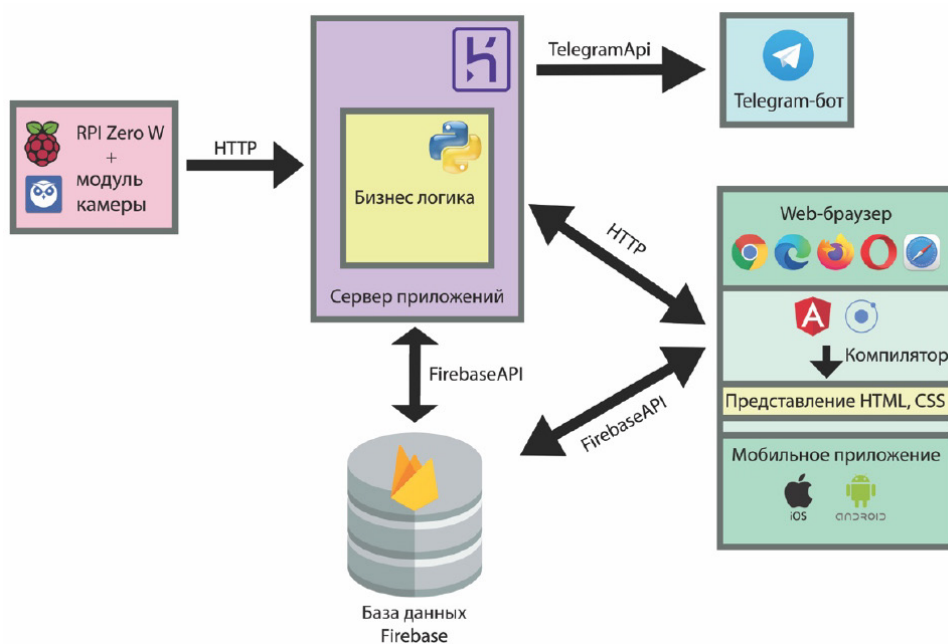


Рис. 8. Проектирование компонентов

обходимые функции на уровне сервисов. После этого слой сервисов вызывает соответствующие функции слоя данных. Слой данных делает запись в базу данных. После чего отправляется сообщение в соответствующий телеграмм-бот.

После взаимодействия пользователя с пользовательским интерфейсом системы, вызываются соответствующие функции в клиентской части приложения для обработки действия пользователя. Затем клиентская часть приложения обращается к Firebase-API для взаимодействия с данными приложения.

В качестве записывающего устройства используется одноплатный компьютер Raspberry Pi Zero W с модулем камеры. На устройстве установлена операционная система MotioneyeOS.

Разработка серверной части системы производится на языке программирования Python с использованием фреймворка Flask. При разработке данной части приложения использовалась система контроля версий Git, а также консоль Heroku для развертывания приложения. Для удобства разработки рекомендуется использовать

среду разработки программного обеспечения Visual Studio Code.

Для развертывания документо-ориентированной базы данных используется Firebase Database. Для хранения медиа-данных должна использоваться Firebase Storage. Для удобства работы с базой данных рекомендуется использовать Firebase Console.

Схема разрабатываемой базы данных представлена на рисунке 9. В сущности user\_account содержится информация о пользователе, а также массивы со списком

устройств, используемых пользователем, а также списком команд выполняемых пользователем в процессе работы устройства.

Каждая из записей user\_device\_list содержит:

1. Основную информацию (параметры и статус девайса);
2. Данные о местоположении девайса;
3. Массив со списком данных, получаемых из девайса;
4. Массив, содержащий сообщения, отправленные пользователю из записывающего устройства.

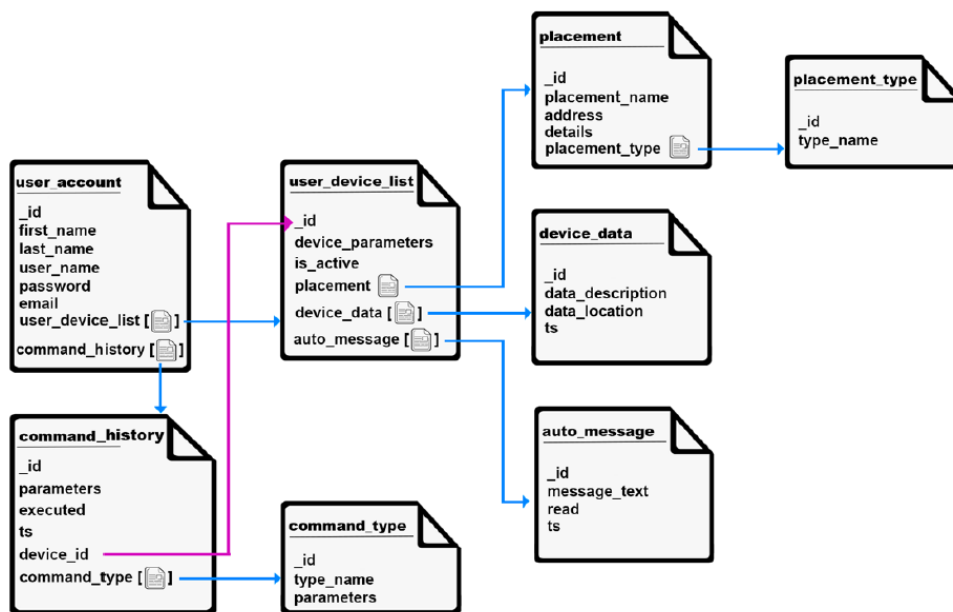


Рис. 9. Документо-ориентированная модель данных

Разработка клиентской части приложения выполнялась при помощи JavaScript-фреймворка Angular и Ionic. Также при разработке данной части приложения использовался HTML5 и CSS, а также препроцессор CSS — SCSS. Для удобства разработки рекомендуется использо-

вать среду разработки программного обеспечения Visual Studio Code.

Пользовательский интерфейс состоит из четырех страниц:

1. Страница регистрации;

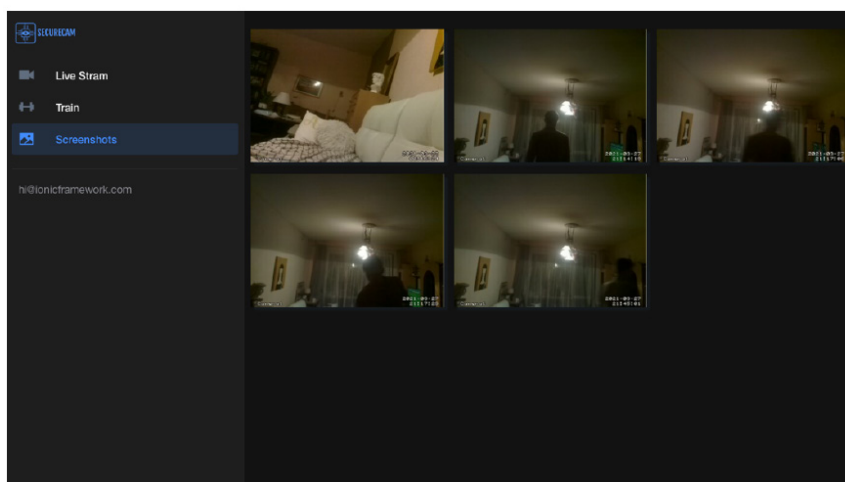


Рис. 10. Пользовательский интерфейс

2. Страница онлайн трансляции;
3. Страница скриншотов;
4. Страница добавления новых лиц, для распознавания.

Пример пользовательского интерфейса представлен на рисунке 10.