

*Салех Лабиб Омар Абдулла*¹,
аспирант высшей школы киберфизических систем и управления;
*Хлопин Сергей Владимирович*²,
доцент высшей школы киберфизических систем и управления,
канд. техн. наук, доцент

МЕТОДЫ РАСПОЗНАВАНИЯ ЛИЦ НА ОСНОВЕ АНАЛИЗА ВИДЕОИЗОБРАЖЕНИЙ

^{1,2} Россия, Санкт-Петербург, Санкт-Петербургский политехнический
университет Петра Великого,
¹ labib_saleh_92@yahoo.com, ² x@spbstu.ru

Аннотация. Объектом исследования являются алгоритмы и методы распознавания лиц на видеоизображениях, библиотеки OpenCV, face_recognition. Объектом разработки является создание системы регистраций событий на основе анализа видеоизображений. Для этого будут использованы видеоизображения, получаемые с камеры. На основе анализа изображений с камеры системой регистрации событий будет детектироваться, распознаваться и проводится фиксация числа людей и время их прохождения. В качестве средств разработки были использованы язык программирования Python, библиотека NumPy, библиотека PIL, библиотека OpenCV, база данных SQLite, библиотека Face_Recognition.

Ключевые слова: компьютерное зрение, детектирование объектов, анализ изображения, распознавание лиц, библиотека opencv, библиотека face_recognition.

*Labib Omar Abdullah Saleh*¹,
Postgraduate Student, Master of Science;
*Sergey V. Khlopin*²,
Associate Professor, Candidate of Technical Sciences;

FACE RECOGNITION METHODS BASED ON VIDEO IMAGES ANALYSIS

^{1,2} Peter the Great St.Petersburg Polytechnic University, St. Petersburg, Russia,
¹ labib_saleh_92@yahoo.com, ² x@spbstu.ru

Abstract. The object of research are algorithms and methods for recognizing faces in video images, OpenCV libraries, face_recognition. The object of development is the creation of an event registration system based on the analysis of video images. For this, video images received from the camera will be used. Based on the analysis of images from the camera, an event registration system will detect, recognize, and record the number of people and their transit time. As development tools, we used the Python programming language, NumPy library, PIL library, OpenCV library, SQLite database, Face_Recognition library.

Keywords: computer vision, detection of objects, image analysis, face recognition, opencv library, face_recognition library.

Введение

Существует несколько причин, по которым в последнее время возрос интерес к распознаванию лиц, в том числе растущая обеспокоенность общественности по поводу безопасности, необходимость проверки личности в цифровом мире и необходимость анализа лица и методов моделирования в управлении мультимедийными данными и компьютерных развлечениях. В последние годы наблюдается значительный прогресс в этой области благодаря достижениям в области моделирования лица и развитие методов анализа изображений. Надежное распознавание лиц позволяет решать более сложные задачи для компьютерного зрения и использования информации о распознанных образах для поведенческого анализа объектов. Предполагается, что разрабатываемая система будет распознавать лица, присутствующие в изображениях и видео последовательностях в автоматическом режиме и создавать отчеты о регистрации объектов.

1. Анализ задачи распознавания лиц в видеопотоках

Для решения задачи детектирования событий необходимо выполнить следующие действия:

- Преобразование исходной видеопоследовательности (изображения) с применением различных фильтров для подготовки к этапу сегментирования;
- Сегментирование изображения и поиск сегментов, содержащих лица;
- Детектирование лица и объекта принадлежащего лицу;
- Анализ поведения детектированного объекта.

Признаки Хаара (примитивы) являются прямоугольниками, состоящими из смежных областей. Они позиционируются (располагаются) на изображении, далее происходит суммирование значений интенсивности пикселей в каждой области [12]. Алгоритм библиотеки OpenCV Eigenfaces основан на методе главных компонент PCA (Principle Component Analysis) [5]. Метод главных компонент (PCA) это довольно мощный способ представления данных, но он не учитывает классы объектов, и может быть потеряно много отличительной информации из-за отбрасывания компонент, которые не вошли в базис. Эту проблему решает алгоритм LDA (Линейный дискриминантный анализ), который является основой метода FisherFaces [5]. Гистограмма локальные бинарные шаблоны (англ. Local Binary Patterns Histogram, LBPН) этот метод работает следующим образом сначала изображение делится на одинаковые блоки, которые образуют сетку. Далее для каждого блока строится гистограмма кодов, которые вычисляются таким образом: берется пиксель, который

сравнивается с соседями, если интенсивность центрального пикселя больше или равна интенсивности соседа, то он обозначается 1, иначе — 0. В итоге каждому пикселю будет соответствовать двоичное число, состоящее из результатов сравнений. Полученные гистограммы объединяются в одну общую, которая является итоговым дескриптором, который используется для классификации лица [3].

Основным методом для детектирования лиц библиотеки `face_recognition` является метод направленного градиента (histogram of oriented gradients — HOG). В основе данного метода лежит предположение, что распределение градиентов интенсивности яркости точек изображения позволяет достаточно точно определить наличие и форму присутствующих на нем объектов. При описании фрагмента изображения оно разбивается на несколько небольших участков (16x16 пикселей), далее называемых ячейками. В каждой ячейке вычисляются гистограммы h_i для направления градиентов яркости внутренних точек. На следующем этапе полученные гистограммы объединяют в одну гистограмму $h = f(h_1, \dots, h_k)$, после чего производится операция нормализации по яркости (L_2 или L_1 норма) и получения нормированного множителя по формуле (1) [6]:

$$h_{L_2} = \frac{h}{\sqrt{|h|_2^2 + \varepsilon}}, h_{L_1} = \frac{h}{|h|_1 + \varepsilon}, h_{\sqrt{L_1}} = \sqrt{h_{L_1}}, \quad (1)$$

где ε — некоторая малая константа.

Вычислении градиентов в ячейках изображения проводится путем фильтрации [10] ячеек ядрами $[-1, 0, 1]$ и $[-1, 0, 1]^T$. В результате фильтрации из исходного изображения получаются значения матриц D_x и D_y . Далее, используя D_x и D_y , вычисляется величина интенсивности D и угол θ направления градиента яркости исходного изображения по формулам:

$$D = \sqrt{D_x^2 + D_y^2}, \quad \theta = \arctg\left(\frac{D_y}{D_x}\right) \quad (2)$$

Получаемые матрицы используются для вычисления углов и величин (модулей) градиентов в каждой точке изображения [6]. В дальнейшем на основе получаемых гистограмм производится поиск лиц в изображениях и обозначение областей присутствия лиц для дальнейшего их детектирования (выделение черт лиц). Для выделения черт лиц человека используется алгоритм “face landmark estimation”. Основной идеей алгоритма является то, что было выбрано 68 специфических точек (меток), присутствующих на лице человека (выступающая часть подбородка, внешние края глаз, внутренний край бровей и т. д. Затем производится настройка алгоритма обучения на поиск этих специфических точек на исходном изображении [7]. Алгоритм “face landmark estimation” основанный на ансамбле деревьев регрессии (regression trees) [7].

Сверточная нейронная сеть (англ. convolutional neural network — CNN) — это искусственная нейронная сеть, имеющая особую архитектуру, нацеленную на эффективное распознавание изображений. Главной особенностью сверточных нейронных сетей считается использование операции свертки для анализа обрабатываемых образов. Операция свертки производится следующим образом: ядро свертки «скользит» по обрабатываемой области и вычисляет взвешенную сумму элементов этой области относительно значений коэффициентов ядра [9]:

$$(f * g)[m, n] = \sum_{k,l} f[m - k, n - l] * g[k, l] \quad (3)$$

где — исходная матрица изображения, g — ядро свертки.

Имеется детектор размером ядра g , который проходит изображение f с определенным шагом (обычно 1). На каждой итерации компоненты детектора умножаются на ядро g . После чего значения складываются и фиксируются в итоговой матрице. Оператор свертки составляет основу сверточного слоя (convolutional layer). Слой состоит из определенного количества ядер g и вычисляет свертку выходного изображения предыдущего слоя с помощью каждого из ядер, каждый раз прибавляя составляющую смещения. Затем ко всему выходному изображению применяется функция активации, например, ReLU (rectified linear unit). Математически операция, выполняемая в сверточном слое (convolutional layer), будет выглядеть следующим образом [9]:

$$x^l = f(x^{l-1} * k^l + b^l) \quad (4)$$

где x^l — выход слоя l , $f()$ — функция активации, b^l — коэффициент сдвига слоя l , $*$ — операция свертки входа x с ядром k .

При этом размер исходных матриц уменьшается [9]:

$$x_j^l = f\left(\sum_i x_i^{l-1} * k_{ij}^l + b_j^l\right) \quad (5)$$

где x_j^l — карта признаков j (выход слоя l), $f()$ — функция активации; b_j^l — коэффициент сдвига слоя l для карты признаков j , k_j^l — ядро свертки j карты, слоя l , $*$ — операция свертки входа x с ядром k .

Субдискретизирующий (subsampling) слой. Слои этого типа выполняют уменьшение размера входной карты признаков (обычно в 2 раза). Это можно делать разными способами, в нашем случае мы выбрали метод выбора максимального элемента (max-pooling) [9] — вся карта признаков разделяется на ячейки 2×2 элемента, из которых выбираются максимальные по значению.

Формально слой может быть описан следующим образом [9]:

$$x^l = f(a^l * \text{subsample}(x^{l-1}) + b^l) \quad (6)$$

где x^l — выход слоя l , $f()$ — функция активации; a^l и b^l — коэффициенты сдвига слоя l ;

subsample — операция выборки локальных максимальных значений [8]. Полносвязный слой (fully connected layer) решает задачу классификации. Данный слой моделирует сложную нелинейную функцию, оптимизируя параметры которой, улучшается качество распознавания. Число нейронов скрытого слоя равно числу карт слоя субдискретизации, так как нейроны каждой карты предыдущего слоя субдискретизации связаны с одним нейроном скрытого слоя [9]. Вычисление значений нейрона:

$$x_i^l = f\left(\sum_j x_j^{l-1} * w_{i,j}^{l-1} + b_j^{l-1}\right) \quad (7)$$

где x_j^l — карта признаков j (выход слоя l), $f()$ — функция активации, b^l — коэффициент сдвига слоя l , $w_{i,j}^{l-1}$ — матрицы весовых коэффициентов слоя l .

Процесс обучения свёрточной нейронной сети с целью вывода наборов характеристик лица требует большого объёма данных и большой производительности компьютера. Разработчики OpenFace уже сделали это [11] и предоставили доступ к нескольким прошедшим обучение сетям. В результате всё, что требуется от нас самих, это провести наши изображения лиц через их предварительно обученную сеть и получить 128 характеристик для каждого лица. Последний шагом является поиск образа человека в созданной базе данных известных лиц, имеющего характеристики, наиболее близкие к характеристикам нашего тестового изображения. Это можно сделать, используя любой базовый алгоритм классификации обучения машин. Какие-либо особые приёмы глубинного обучения не требуются. Мы будем использовать простой линейный SVM-классификатор, но могут быть применены и многие другие алгоритмы классификации.

2. Разработка системы регистрации на основе анализа видеоизображений

В этом разделе мы будем проводить анализ двух библиотек: `opencv` и `face_recognition`. Первый тест будет проходить для библиотеки `opencv`. Мы будем тестировать точность распознавания лиц при использовании алгоритмов LBP, Eigenfaces и Fisherfaces которые предоставляет нам библиотека `opencv`. Мы будем тестировать точность распознавания лиц используя исходные изображения, которые были обученными алгоритмами LBP, Eigenfaces и Fisherfaces и у нас получается следующие результаты как указаны на таблицы 1.

Таблица 1

Результаты точность распознавании лиц используя исходное изображения

Faces	Names	OpenCV		
		LBPH	Eigenfaces	Fisherfaces
	The Rock	65%	56%	28%
	Angelina	68%	20%	53%
	Jason	67%	17%	42%
	Young thug	72%	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Bruce Lee	69%	26%	23%
	Face	37%	40%	30%
	Timati	46%	детектирует, но не распознает лицо (0%)	56%
	The Joker	70%	50%	42%
	50 Cent	68%	37%	39%
	Megan Fox	64%	детектирует, но не распознает лицо (0%)	32%
Минимальная точность распознавания (min)		37%	17%	23%
Максимальная точность распознавания (max)		72%	56%	56%
Средняя точность распознавания		62.6%	35.14%	38.3%

Мы будем использовать 10 разных изображений одного и того человека для распознавания лиц с исходной изображением, которое сохранено и обучено алгоритмом LBPH, Eigenfaces и Fisherfaces. При тестировании точность распознавание лиц алгоритмом LBPH, Eigenfaces и Fisherfaces у нас получается следующие результаты, которые указаны в таблицы 2.

Таблица 2

Сравнения распознавание лиц разные изображениях одно и того человека с исходной изображения

Faces	OpenCV		
	LBPH	Eigenfaces	Fisherfaces
	Распознает как The Rock	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как The Rock	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как Angelina	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как The Rock	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как The Rock	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как The Rock	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как Angelina	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как The Rock	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как Angelina	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)
	Распознает как The Rock	детектирует, но не распознает лицо (0%)	детектирует, но не распознает лицо (0%)

Результаты таблицы 2 показывают не корректность работы алгоритмов LBPH, Eigenfaces и Fisherfaces при распознавании лиц одного и того же человека используя 10 не исходных изображений.

Будем тестировать библиотеку face_recognition первый тест — это тест средний процент успеха распознавания (кадров в 10 тестах) известного лица человека. Результат тестирования указаны в таблицы 3.

Таблица 3

Тестирования библиотеку face_recognition

Faces		Names	face_recognition		
Now	Then		Точность распознавания (кадров в 10 тестах) известного лица	Точность распознавания объекта по фотографии, сделанной в прошлом (10 лет назад)	Точность детектирования неизвестного лица человека в 10 тестах по фотографии, сделанной в прошлом
		The Rock	98%	63%	36%
		Angelina	84%	94%	6%
		Jason	98%	43%	55%
		Young thug	Распознает как 50 Cent (0%)	Распознает как 50 Cent (0%)	0%
		Bruce Lee	100%	99%	0%
		Face	90%	97%	0%
		Timati	98%	97%	2%
		The Joker	98%	98%	2%
		50 Cent	66%	52%	0%
		Megan Fox	49%	58%	39%
Минимальная точность распознавания (min)			49%	43%	2%
Максимальная точность распознавания (max)			100%	99%	55%
Средняя точность распознавания			86.7%	77.8%	23.3%

Из данных таблицы 3 следует, что все лица были распознаны, кроме изображения лица Young Thug. Лицо Young Thug было признано другим лицом (50 Cent). Распознавание лица Young Thug было неправильно распознано библиотекой face_recognition. В таблице 3 мы также можем заметить, что лицо Bruce Lee было распознано со 100 % точностью относительно исходного изображения. В таблице 3 лицо Jason на основе фотографии, сделанной в прошлом, было распознано как Неизвестное с точностью 55 %. В таблице 3 лицо Jason, основанное на фотографии, сделанной в прошлом, имеет самую высокую точность распознавания его лица как Неизвестного. В таблице 3 лицо Megan Fox имеет минимальную точность распознавания.

Сравнение таблицы 1 с таблицей 3. Таблица 3 имеет самую высокую среднюю точность распознавания. Сравнение таблицы 2 с таблицей 3 показывает, что использование разных изображений фотографий может быть распознано библиотекой face_recognition, но не распознано библиотекой OpenCV. Сравняя таблицу 3, таблицу 2 и таблицу 1, мы видим, что библиотека face_recognition имеет лучший результат, чем библиотека OpenCV, в обнаружении и распознавании лиц. По результатам, библиотека face_recognition является лучшим вариантом для разработки системы регистрации событий. Как показывают результаты, при использовании библиотеки face_recognition вероятность и точность распознавания высоки, а изменение условий освещения не влияет на результаты распознавания. Сравняем результаты тестирования библиотеки OpenCV и face_recognition и выбираем лучший. Поэтому для реализации системы регистрации событий мы будем использовать библиотеку face_recognition.

На рисунке 1 показан пример работы библиотеки face_recognition для распознавания лиц.



Рис. 1. Пример работы библиотека face_recognition

Как мы видим на рисунке 1, библиотека face_recognition может правильно распознавать лица двух разных людей одновременно, определять лица, которые не включены в базу данных как неизвестные лица, и, наконец, распознавать лица человека в любом возрасте.

Заключение

В рамках выполнения работы были исследованы методы распознавания лиц на видеоизображениях, модель сверточной нейронной сети (CNN). Был пройден ряд этапов, в ходе которых были проведены обзор методов поиска и распознавания лиц, проанализирована задача распознавания лиц, был выбран инструментарий, изучены особенности его применения, для выбранных методов выполнена их реализация и тестирование, проведён сравнительный анализ между методами и библиотеками Open CV и Face_Recognition.

Для дальнейшего решения поставленной задачи и разработки системы регистраций была выбрана библиотека face_recognition, так как результаты показали, что вероятность и точность распознавания лиц у нее более высокое и изменения условий освещения не влияют на результаты распознавания.

Список литературы

1. Шапиро Л. Компьютерное зрение. – М.: БИНОМ. Лаборатория знаний, 2013. – 752 с.
2. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход / Пер. с англ. – М.: Издательский дом Вильямс, 2004. – 928 с.
3. Face Recognition with Local Binary Patterns. – URL: https://link.springer.com/chapter/10.1007/978-3-540-24670-1_36 (дата обращения: 02.11.2022).
4. Le Cun Y., Bengio Y., Hinton G. Deep learning // Nature. – 2015. – Vol. 521 – № 7553. – P. 436.
5. Face Recognition with OpenCV. – URL: http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html (дата обращения: 02.11.2022).
6. Южаков Г.Б. Алгоритм быстрого построения дескрипторов изображения, основанных на технике гистограмм ориентированных градиентов // Труды Московского физико-технического института. – 2013. – Vol. 5, no. 3 (19). – Pp. 084–091.
7. Kazemi V., Sullivan J. One Millisecond Face Alignment with an Ensemble of Regression Trees // KTH, Royal Institute of Technology. – Stockholm, Sweden, 2014.
8. Le Cun Y., Bengio Y., Haffner P., Bottuo L. Gradient-based learning applied to document recognition. – 1998.
9. Борисов Е.С. Классификатор изображений на основе сверточной нейронной сети [Электронный ресурс] // Дом-страница Евгения Сергеевича Борисова. [Россия], 2016. – URL: <http://mechanoid.kiev.ua/ml-lenet.html> (дата обращения: 02.11.2022).
10. Асташенкова Л. К., Рощупкина С. Н., Кудринская К. В. Распознавание ключевых точек лица на изображении человека // Молодой ученый. – 2019.
11. OpenFace. – URL: <https://cmusatyalab.github.io/openface/> (дата обращения: 02.11.2022).
12. Жарких А. А. Обработка изображений на основе вейвлет-преобразования в базисе Хаара над конечным полем нечетной характеристики // Вестник МГТУ. – 2009. – Том 12, № 2.