

Засыпкин Денис Сергеевич, студент-магистр, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»

Белов Юрий Сергеевич, к.ф. -м.н., доцент, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»

ОБЗОР АЛГОРИТМОВ РАСПОЗНАВАНИЯ ЛИЦА ЧЕЛОВЕКА В БИБЛИОТЕКЕ OPENCV

Аннотация: Обнаружение и распознавание лица человека часто применяется в системах идентификации личности. В статье описаны алгоритмы, используемые для этих целей в библиотеке OpenCV. В частности, описаны алгоритмы распознавания лица: Eigenfaces, Fisherfaces и LBPH.

Ключевые слова: Обнаружение лиц, распознавание лиц, Fisherfaces, OpenCV, Eigenfaces.

Annotation: Human face detection and recognition is often used in personal identification systems. The article describes the algorithms used for these purposes in the OpenCV library. In particular, face recognition algorithms are described: Eigenfaces, Fisherfaces and LBPH.

Keywords: Face detection, face recognition, Fisherfaces, OpenCV, Eigenfaces.

Процедура распознавания. Основная цель распознавания - сопоставить лицо на картинке с информацией о личности человека. Эта процедура обычно используется после обнаружения и работает с изображениями лиц, которые уже обнаружены, извлечены, обрезаны, выровнены и изменены в размере, поэтому алгоритм может сосредоточиться на поиске важных характеристик на

изображении лица, которые лучше всего описывают личность человека и позволяют распознать его на максимально возможном количестве изображений.

Распознавание всегда состоит из двух основных частей, независимо от используемого метода или алгоритма, обучения распознавателя и самого распознавания [3]. В первой части распознаватель извлекает полезные характеристики из набора данных изображений, определенного для обучения. Этот набор данных очень важен для точности распознавания, на каждом изображении должно быть только одно лицо, чтобы впоследствии сопоставить лицо с личностью. Обычно набор данных содержит подмножество изображений, на которых изображен один и тот же человек, но с разных ракурсов и с разной степенью освещения. Точная реализация зависит от используемого инструмента или библиотеки. Несмотря на то, что распознавание может быть на видеозаписи (даже в реальном времени), распознавание лиц выполняется на изображениях, в видеозаписях выбираются определенные кадры (частота кадров зависит от реализации) и только на этих кадрах выполняется распознавание.

Распознаватели человеческих лиц. Непосредственно в библиотеке OpenCV реализованы три основных алгоритма распознавания лиц, которые могут использоваться для идентификации лица по известным и обученным наборам данных, алгоритмы сильно различаются по точности, условиям ввода и скорости.

Для распознавания лиц распознаватель должен иметь в своем распоряжении набор подготовленных соответствующим образом обучающих изображений. Каждое изображение в наборе должно иметь идентификатор, который может быть номером или именем в виде строки. Изображения одного и того же человека должны иметь одинаковый идентификатор, чтобы алгоритм мог объединить несколько представлений. Уже обрезанные и выровненные изображения помещаются в функции как тренировочные.

Распознаватели в OpenCV. Распознаватели лиц OpenCV имеют общий абстрактный базовый класс `cv::face::FaceRecognizer`, от которого происходят классы `cv::face::BasicFaceRecognizer`, обеспечивающие как алгоритм Eigenfaces

(созданный с помощью `createEigenFaceRecognizer ()`), так и `Fisherfaces` (созданный с помощью `createFisherFaceRecognizer ()`) и `cv::face::LBPHFaceRecognizer`, отвечающий за распознавание лиц LBPH с конструктором `createLBPHFaceRecognizer ()`, [6]. Все три метода имеют общий интерфейс для изучения лиц `train ()` и распознавания их `predic ()`. Обнаружение лиц хорошо подходит для фронтального обнаружения, смешивание фронтальных изображений с изображениями профиля усложняет идентификацию и снижает общую точность.

Алгоритм Eigenfaces. Алгоритм Eigenfaces для распознавания лиц был одним из первых алгоритмов, описанных для распознавания объектов, и, как следует из названия, основан на Eigenfaces. Eigenfaces - составляющие черты лица, и комбинируя определенное количество черт лица с весом, можно реконструировать внешний вид человека. Собственные черты создаются из обучающего изображения с помощью анализа главных компонент (PCA), или, точнее, на ковариационной матрице обучающих данных с помощью этого метода получается большое количество пар собственных векторов и собственных значений, а собственный вектор с наибольшими собственными значениями называется собственными чертами. Этот алгоритм предлагает отличное сжатие данных, хранящая обучающая библиотека меньше, чем у других алгоритмов, потому что он представляет каждое лицо как линейную комбинацию взвешенных собственных векторов, которая требует небольшого объема памяти. Однако большим недостатком является высокая восприимчивость к различным условиям освещения или изменения фона.

На этапе обучения изображение из набора обучающих данных преобразуется в плоский вектор - из матрицы изображения с размером $[m \times n]$ в вектор изображения с размером $[(m \times n) \times 1]$. Последовательно это означает μ всех векторов (изображений из набора обучающих образов) и $x_i \in R_d$, который является вектором изображения с размерностью.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

Матрица ковариаций измеряет отклонения значений во всех измерениях, в которых вычисляется. Полезно определить, какие измерения различаются больше всего, что означает, что они содержат много данных для распознавания из измерений, которые не сильно различаются и для распознавания не важны [8]. Чтобы лучше понять вычисления, мы объясняем их шаг за шагом. Сначала вычитая средний вектор μ из каждого вектора, получаем разницу i между конкретным изображением i и средним изображением. Складывая все матрицы разностей вместе, получаем матрицу $A = \{1, \dots, \text{Phin}\} \in \mathbb{R}^{D \times M}$ и ковариационная матрица просто вычисляется из этой матрицы A как $C = A * \text{transpose}(A)$. Весь процесс за один шаг можно представить следующим образом:

$$S = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$$

Из ковариационной матрицы C вычисляются собственные значения λ_i и собственные векторы $Sv_i = \lambda_i v_i$ $i = 1, 2, \dots, n$. На следующем шаге собственные векторы v_i упорядочиваются в соответствии с их собственным значением λ_i в порядке убывания. Берется k количество собственных векторов с наивысшими собственными значениями, количество собственных векторов k затем решается, насколько точным может быть алгоритм, потому что каждый собственный вектор v_i представляет определенную черту или особенность. Заключительный шаг обучения состоит в том, чтобы выяснить, насколько каждая из k черт покрыта для каждого обучающего изображения, которое позже будет использовано для распознавания.

На этапе распознавания первые шаги аналогичны этапу обучения: преобразование изображения в плоский вектор и его нормализация по среднему вектору, который был вычислен на этапе обучения. Затем находим вес изображения и вычисляем ошибку между весами изображения и всех обучающих изображений. Ошибка может быть суммой простых или евклидовых расстояний каждой пары весов. Если наименьшая ошибка ниже заданного порога, изображение идентифицируется как изображение из обучающего набора

с наименьшей ошибкой, в противном случае человек на изображении помечается как неизвестный. В OpenCv алгоритм реализован в классе EigenFaceRecognizer.

Алгоритм Fisherfaces. Алгоритм Fisherface был создан для повышения точности метода Eigenface. Анализ основных компонентов, составляющий ядро алгоритма Eigenface, фокусируется на максимальной дисперсии изображения, которое хорошо представляет данные и не требует много памяти. Когда некоторые компоненты выбираются, а некоторые отбрасываются, теряется много различающей информации. С другой стороны, метод Fisherface применяет линейный дискриминантный анализ (LDA) для уменьшения размерности и получения характеристик, которые лучше всего разделяют людей. Цель LDA состоит в том, чтобы максимизировать соотношение между индивидуальным разбросом и внутри индивидуальным разбросом, тем самым получая наиболее полезные функции для различения.

Для достижения независимости от условий освещения важно предоставить несколько изображений в обучающую библиотеку со всеми возможными состояниями и выражениями лица каждого человека [1]. Этот метод улучшает точность Eigenfaces в основном для ситуаций, когда Eigenfaces не справлялся чаще всего: изображения с разными условиями освещения [2].

Алгоритм работает с множеством C классов / индивидов $X = \{X_1, \dots, X_C\}$, класс X_i - это вектор длины N , созданный сглаживающим изображением i из обучающей выборки: $X_i = \{x_1, \dots, x_n\}$. Как и в методе Eigenfaces, вычисляется среднее всех векторов μ , а также μ_i среднее значение класса i : $i \in \{1, \dots, C\}$.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$
$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j$$

Использование μ и μ_i позволяет вычислить разброс между классами S_B и разброс внутри классов S_W , где μ - общее среднее значение, μ_i - среднее значение класса i , вычисленное на предыдущем шаге. Переменная c - это количество

классов, N - количество выборок для класса i , X_i - это одна выборка из класса i , x_j показывает пример этого X_i [7].

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu) (\mu_i - \mu)^T$$

$$S_W = \sum_{i=1}^c \sum_{x_j \in X_i} (x_j - \mu_i) (x_j - \mu_i)^T$$

На следующем этапе вычислений он ищет оптимальную Fisherface W_{opt} , которая максимизирует отношение детерминанта между разбросом внутри класса S_W и разбросом между классами S_B и, таким образом, разделением классов. Это невозможно решить, потому что S_W имеет размер не более $(N - c)$, N - это количество выборок, а c - количество классов, что делает матрицу разброса S_W сингулярной. Образцы проецируются в $(N - c)$ -мерное пространство, чтобы получить обобщенные собственные векторы W_{fld} . Окончательная матрица W , которая проецирует образец в $(c-1)$ -мерное пространство, получается в уравнении 2.10 [4].

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

$$W_{pca} = \arg \max_W |W^T S_T W|$$

$$W_{fld} = \arg \max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|}$$

$$W = W_{fld}^T W_{pca}^T$$

Подобно методу Eigenfaces, Fisherface вычисляет линейный дискриминантный анализ, который позволяет сравнивать результаты тестируемого изображения и изображений, сохраненных во время фазы обучения. Чем больше разница при прогнозировании двух векторов, содержащих признаки, которые разделяют индивидов W , тем ниже уверенность в результатах. В классе OpenCV, реализующий метод Fisherface, называется FisherFaceRecognizer.

Алгоритм LBPН. Гистограммы локальных бинарных шаблонов (LBPН) - это алгоритм для распознавания лиц, который представляет локальные объекты, такие как края на лицах, и поэтому не так подвержен ошибкам, вызванным различными условиями освещения. Предыдущие два алгоритма, Eigenfaces и Fisherfaces, рассматривают данные как вектор в пространстве изображений высокой размерности, эта многомерность привела к тому, что пришлось искать подпространства низкой размерности, чтобы избежать работы с подавляющими структурами. LBPН предназначен для распознавания людей в различных условиях освещения и на заднем фоне, и в отличие от Fisherfaces ему не нужно много изображений одного и того же человека со всеми возможными условиями, которые могут возникнуть. Конкретные черты человека в этом алгоритме представлены с помощью простого вектора данных.

Для обучения алгоритма нам понадобится набор данных с несколькими изображениями лиц человека, которых мы хотим распознать, возможно, с разным расположением и выражением. Объем изображений для каждого человека существенно влияет на точность [5]. Алгоритм преобразовывает изображение в оттенки серого, где каждое изображение представлено значением интенсивности в диапазоне от 0 до 255. Затем для каждого окна размером 3x3 пикселя он сравнивает центральное значение с другими 8 значениями, если значение на краю ниже, то устанавливает 0, иначе 1. Затем удаляется центральное значение. Окно теперь содержит только двоичные значения 0 и 1, поэтому можно объединить эти 8 значений в одно новое значение, которое можно рассматривать как двоичное число, и это число помещается как значения для пикселя в центре. Весь процесс показан на рисунке 1.

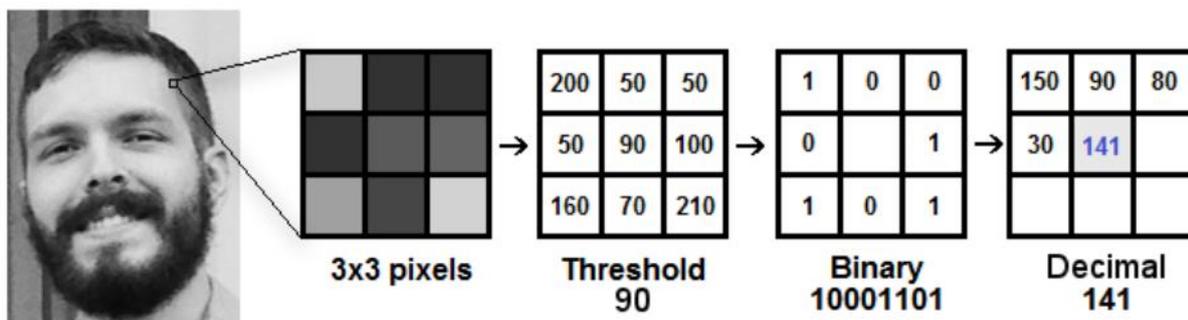


Рисунок 1: Принцип алгоритма LBPН, из изображения берется блок 3x3 пикселей в оттенках серого, показанный в диапазоне 0-255, если значение выше, чем центральный пиксель, результат на пикселе приравнивается 1, в противном случае 0. В конце двоичное значение объединяется и помещается как в центральный пиксель.

На этом этапе кадр выглядит как изображение в градациях серого, на котором выделяются быстрые изменения интенсивности, такие как края или углы. На следующем шаге изображение с последнего шага делится на несколько сеток, количество сеток зависит от требуемой точности, скорости вычислений или требований к памяти. Для каждой сетки мы создаем гистограмму (значения от 0 до 255, поскольку мы работаем в градациях серого), показывающую, как часто данное значение / интенсивность пикселей встречается в сетке. Объединяя каждую гистограмму, мы получаем новую большую гистограмму с $256 * GridX * GridY$ позициями, которая представляет конкретную особенность человека. Гистограммы из нескольких обучающих изображений одних и тех же людей сохраняются и сравниваются отдельно для достижения более высокой точности.

После того, как мы обучили алгоритм с изображениями людей, которых мы хотим распознать, он может перейти к фактическому распознаванию. Тот же процесс, что описан в предыдущих параграфах, выполняется для получения гистограммы. Сравнение этой гистограммы с гистограммами из обучения выполняется по сумме евклидовых расстояний или разнице в абсолютных значениях между каждой отдельной позицией между двумя гистограммами. В частности, в OpenCV это делается с помощью евклидовых расстояний. Метод выбирает наименьшее различие из всех гистограмм и возвращает личность человека из обучающего набора, которому принадлежала гистограмма с наименьшим различием и уровнем точности, как числовая разница между двумя гистограммами.

В OpenCV класс, специфичный для распознавания лиц LBPН, называется LBPНFaceRecognizer. Этот класс предоставляет все необходимые функции для обучения и запуска распознавания лиц и настройки распознавания LBPН. В этом классе можно установить максимальную разницу между диаграммами, на

которых распознавание возвращает положительное значение, значение *GridnX* и *GridY* или радиус, поэтому окно с обнаружением краев изменяется с 3x3 пикселей на большее окно.

Выводы: В данной статье были подробно рассмотрены алгоритмы обнаружения и распознавания лица человека в библиотеке OpenCV. В числе рассмотренных алгоритмов такие, как: обнаружение объектов с помощью классификатора Виолы-Джонса, распознавание лиц с использованием алгоритмов Eigenfaces, Fisherfaces и LBPН.

Библиографический список:

1. Бацева Д.А., Черепков Е.А., Белов Ю.С. ОПРЕДЕЛЕНИЕ ХАРАКТЕРНЫХ ТОЧЕК ЛИЦА ДЛЯ ПОСТРОЕНИЯ ОБЪЕМНОЙ МОДЕЛИ ГОЛОВЫ ЧЕЛОВЕКА // Электронный журнал: наука, техника и образование. 2018. № 1 (17). С. 70-75.

2. Кучер М.Ю., Белов Ю.С. ПОДХОДЫ К РАСПОЗНАВАНИЮ ЛИЦ И ИХ МЕТОДЫ // В сборнике: Технические и естественные науки. сборник избранных статей по материалам Международной научной конференции. Санкт-Петербург, 2020. С. 38-40.

3. Кучер М.Ю., Белов Ю.С. ЭТАПЫ ФУНКЦИОНИРОВАНИЯ СИСТЕМ РАСПОЗНАВАНИЯ ЛИЦ // В сборнике: Наука. Исследования. Практика. сборник избранных статей по материалам Международной научной конференции. Санкт-Петербург, 2020. С. 91-93.

4. Face Recognition with OpenCV: Fisherfaces. - dev team, URL: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#fisherfaces (дата обращения: 22.06.2021).

5. Face Recognition with OpenCV: Local Binary Patterns Histograms. - dev team, URL: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms (дата обращения: 27.06.2021).

6. OpenCV documentations [Electronic resource]. – opencv dev team, 2011 – 2014 – URL: <https://docs.opencv.org> (дата обращения: 25.06.2021).

7. Брилюк, Д.В., Распознавание человека по изображению лица и нейросетевые методы / Минск: Институт Технической Кибернетики Национальной Академии Наук Беларуси / 2004 г. // [Электронный ресурс]. – Режим доступа: URL: https://www.researchgate.net/publication/236605826_ (дата обращения 24.06.2021).

8. Gary, Bradski Learning OpenCV: Computer Vision with the OpenCV Library / Gary Bradski, Adrian Kaehler. - М.: O'Reilly Media, 2008. - 556 с.