

СИСТЕМА РАСПОЗНАВАНИЯ ЛИЦ НА ОСНОВЕ БИБЛИОТЕКИ АЛГОРИТМОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ OPENCV

FACE RECOGNITION SYSTEM BASED ON THE LIBRARY OF COMPUTER VISION ALGORITHMS OPENCV

H. Zhou
G. Afanasyev
A. Afanasyev
A. Filatova

Summary. The article presents the results of creating a face detection system that was developed using the AdaBoost face detection algorithm based on haar-like features, a cascade classifier trained by this algorithm, and the OpenCV computer vision algorithm library. The program code of the face recognition system was developed in the VC++12 language. Experimental results show that the development cycle of the detection system is short, the detection speed is fast, the real-time performance is strong, the detection speed is high, which can be used as the basis for the development of face recognition and face tracking systems.

Keywords: computer vision, face recognition, AdaBoost algorithm, Haar-like signs, OpenCV.

Чжоу Хан

*Магистрант, Московский государственный
технический университет им. Н.Э. Баумана
(национальный исследовательский университет)*
825588535@qq.com

Афанасьев Геннадий Иванович

*К.т.н., доцент, Московский государственный
технический университет им. Н.Э. Баумана
(национальный исследовательский университет)*
gaipcs@bmstu.ru

Афанасьев Арсений Геннадьевич

*Ассистент, Московский государственный технический
университет им. Н.Э. Баумана (национальный
исследовательский университет)*
afanasievag@bmstu.ru

Филатова Анастасия Евгеньевна

*Магистрант, Московский государственный
технический университет им. Н.Э. Баумана
(национальный исследовательский университет)*
hummuschan19@gmail.com

Аннотация. В статье приводятся результаты создания системы обнаружения лиц, которая была разработана с использованием алгоритма обнаружения лиц AdaBoost, основанного на хаароподобных признаках, каскадного классификатора, обученного этим алгоритмом, и библиотеки алгоритмов компьютерного зрения OpenCV. Программный код системы распознавания лиц был разработан на языке VC++12. Экспериментальные результаты показывают, что цикл разработки системы обнаружения короткий, скорость обнаружения быстрая, производительность в реальном времени сильная, скорость обнаружения высокая, что может быть использовано в качестве основы для разработки систем распознавания лиц и слежения за лицами.

Ключевые слова: компьютерное зрение, распознавание лиц, алгоритм AdaBoost, Хаароподобные признаки, OpenCV.

Введение

С быстрым развитием компьютерных технологий и технологий цифровой обработки сигналов, технология компьютерного зрения постепенно получила широкое распространение. OpenCV [1, 2] (Open Source Computer Vision Library) — это пакет алгоритмов компьютерного зрения с открытым исходным кодом, предоставленный компанией Intel, состоящий из ряда функций языка C и небольшого количества классов C++, с сотнями API среднего и высокого уровня для обработки изображений и компьютерного зрения. Он может быть использован как идеальный инструмент для вторичного развития. Алгоритм обнаружения лиц AdaBoost, основанный на Хаароподобных признаках, обладает такими преимуществами, как высокая скорость обнаружения, высокая производительность в реальном времени

и хорошая устойчивость. В данной работе на платформе разработки программного обеспечения VC++12 описывается разработанная система обнаружения лиц на статических изображениях, в видео потоке с видео камеры и на видео в формате AVI.

Алгоритм обнаружения лиц AdaBoost

Цель обнаружения лица — отличить области лица от областей без лица на неподвижном изображении или видео. Метод обнаружения лиц, предложенный Viola-Jones и др., представляет собой метод, основанный на интегральной карте, каскадном классификаторе и алгоритме AdaBoost, который может быть реализован в следующие 3 шага [1,3].

1. Использование Хаароподобных характеристик для представления лиц и нового представления

изображения, «интегральной карты», для быстрого вычисления их собственных значений.

2. Алгоритм машинного обучения AdaBoost используется для выбора прямоугольных признаков (слабых классификаторов), которые наилучшим образом представляют лицо, и построения слабых классификаторов в сильный классификатор путем взвешенного голосования.
3. Несколько сильных классификаторов обучаются последовательно, чтобы сформировать каскадный структурированный классификатор для улучшения скорости обнаружения классификатора.

Для представления черт лица используются простые прямоугольные признаки, названные так потому, что они напоминают вейвлеты типа Хаара, предложенные Виолой и другими [4]. Обычно используются такие признаки типа Хаара, как краевые признаки, линейные признаки и центральные признаки. В них собственное значение — это разница между суммой значений серого цвета всех пикселей внутри двух или более прямоугольников одинаковой формы и размера на изображении. Сумма значений серого цвета всех пикселей в белой прямоугольной области минус сумма значений серого цвета всех пикселей в черной прямоугольной области равномерно используется в системе [4].

Для того чтобы повысить скорость обучения выборки и обнаружения лиц, Виола и др. подходы предложили интегральное представление изображения, которое позволяет быстро вычислять прямоугольные собственные значения. Интегральная карта изображения рассчитывается, как показано в уравнении (1) [5].

$$ii(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} i(x', y') \quad (1)$$

Где $i(x', y')$ — пиксельное значение изображения в точке (x', y') , а $ii(x, y)$ — вычисленное интегральное значение изображения относительно точки (x, y) .

При использовании рекурсивных формул [5] (2) и (3) интегральная карта может быть вычислена путем обхода всех точек исходного изображения за один раз, где $s(x, y)$ — значение интеграла строки в точке (x, y) , $s(x, -1) = 0$, $ii(-1, y) = 0$.

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (2)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (3)$$

Алгоритм AdaBoost [1,6] — это итерационный алгоритм выбора признаков и обучения классификаторов. Основная идея заключается в обучении одного и того же классификатора (слабого классификатора) на разных обучающих наборах, а затем сбор этих слабых классификаторов на разных обучающих наборах определенными

методами для формирования окончательного сильного классификатора. В этом алгоритме различные обучающие наборы используются для настройки весов каждой выборки. В начале каждой выборке присваивается одинаковый вес, и слабый классификатор $h1(x)$ обучается на этом распределении выборок. Образцам, которые неправильно классифицированы по $h1(x)$, придается больший вес, а образцам, которые правильно классифицированы по $h1(x)$, — меньший, что позволяет выделить неправильно классифицированные образцы и объединить их в новое распределение образцов. В то же время, $h1(x)$ присваивается вес в зависимости от количества ошибок классификации, указывающий на важность этого слабого классификатора, причем чем меньше ошибок классификации, тем больше вес. С новым распределением выборки он обучается снова, чтобы получить слабый классификатор $h2(x)$ и его веса. После T циклов обучения получается T слабых классификаторов и T соответствующих весов, а затем эти T слабых классификаторов складываются вместе с определенными весами для получения окончательного сильного классификатора.

Каскадный классификатор — это каскад из нескольких сильных классификаторов, также известный как каскадный детектор или водопадный детектор. Каждый слой представляет собой сильный классификатор, обученный алгоритмом AdaBoost. Правильный результат первого слоя вызывает классификацию второго слоя, правильный результат второго слоя вызывает классификацию третьего слоя, и так далее. Таким образом, окно с подозреваемым лицом поочередно проходит через каждый слой классификатора и подтверждается как лицо. И наоборот, если на одном из слоев обнаружено окно признается нелицевым, обнаружение этого окна немедленно прекращается и начинается обнаружение следующего окна [6]. Такая каскадная структура классификатора разработана с использованием принципа пошагового усложнения, при этом начальные слои сильных классификаторов относительно просты, и обычно один слой состоит из одного или нескольких слабых классификаторов. Это позволяет быстро отсеивать подокна, которые явно не являются лицами, позволяя впоследствии обнаруживать все меньше и меньше целей, что значительно увеличивает скорость обнаружения. Кроме того, классификаторы разработаны с возможностью изменения размера для поиска целевых областей разного размера, что позволяет избежать прямых преобразований масштаба изображения, уменьшить вычислительные усилия и увеличить скорость обнаружения.

Разработка системы распознавания лиц

Каскадный классификатор был создан с помощью Haartraining OpenCV. Программа сначала подготавливает большое количество положительных образцов (образцы с лицом) и отрицательных образцов (образцы

без лица), затем создает `examples.exe` и `haartraining.exe` в подкаталоге `bin` каталога установки OpenCV для создания образцов и обучения до сходимости, и, наконец, использует `performance.exe` в подкаталоге `bin` для обучения образцов для тестирования. Создается окончательный XML-файл каскадного классификатора [7]. Для того чтобы быстро разработать систему обнаружения лиц, в данной работе для экспериментов были выбраны каскадные классификаторы хаароподобных признаков `haarcascade_frontalface_alt_tree.xml` и `haarcascade_profileface.xml` из OpenCV.

VC++12 был использован для создания внешнего интерфейса, который разделен на 3 части, а именно: область отображения обнаружения, область отображения позиционирования и область меню. Область отображения обнаружения в основном реализует отображение результатов обнаружения лиц на основе неподвижных изображений, видео с камеры и видео в формате AVI. Область отображения позиционирования реализует отображение в реальном времени помеченных областей лица. Область меню реализует операции обнаружения и маркировки лиц на основе неподвижных изображений, видео с камеры и видео в формате AVI, а также отображение и хранение изображений помеченных областей лица.

Процесс обнаружения лиц на основе статических изображений можно условно разделить на 4 процесса:

1. Загрузка обученного каскадного классификатора.
2. Преобразование его во внутренний формат для обработки компьютером распознавания.
3. Загрузка статического изображения для обнаружения.
4. Обнаружение области лица на изображении и маркировка местоположения лица.

Основной процесс обнаружения лица на основе видеокамеры:

1. Обнаружить и запустить камеру, если камера не установлена, будет выдано предупреждение и обнаружение закончится, из-за разнообразия устройств захвата видео и требований к обработке видео в реальном времени, в данной работе используется технология обработки видео в реальном времени `DirectX`.
2. Загрузить обученный каскадный классификатор и преобразовать его во внутренний формат.
3. Оценить бит флага обнаружения, если он истинный. Если флаг обнаружения истинный, изображение области лица будет сохранено, и следующий кадр будет обнаружен, в противном случае обнаружение будет завершено, если флаг обнаружения ложный.

Видео можно рассматривать как временное расположение изображений кадр за кадром, так что видеообнаружение лица можно рассматривать как непрерывное

обнаружение набора последовательностей изображений, поэтому процесс обнаружения лица на основе видео AVI и обнаружение лица на основе видеокамеры в основном одинаковы, разница в том, что ему не нужно определять начало шага камеры, только загрузить видео AVI, поэтому процесс не повторяется.

Основные функции для распознавания лиц в библиотеке компьютерного зрения OpenCV являются:

1. `cascade = (CvHaarClassifierCascade*) cvLoad(cascade_name, 0, 0, 0)` — эта функция реализует функцию загрузки каскадного классификатора, где `cascade_name` — путь к каскадному классификатору.
2. `cvGrabFrame(capture)` и `frame = cvRetrieveFrame(capture)` — эти две функции вызываются вместе для быстрого получения кадров видео в формате AVI, где `capture = cvCaptureFromAVI(path)` считывает захваченное видео в формате AVI.
3. `CvMemStorage* storage = cvCreateMemStorage(0)` — эта функция создает блок памяти с размером по умолчанию 64 КБ и возвращает указатель на головку блока.
4. `CvSeq* faces = cvHaarDetectObjects(small_img, cascade, storage, 1.1, 2, 0, cvSize(30, 30))` — функция использует обученный каскадный классификатор для поиска прямоугольных областей на изображении, содержащих лица, и возвращает эти области в виде последовательности прямоугольных коробок для возврата.
5. `cvRectangle(img, LeftPoint, RightPoint, color, 3, 8, 0)` — эта функция реализует рамку для выделения области обнаруженного лица.
6. `cvSaveImage(savepath+strcat(strCount, «.bmp»), OutImage)` — функция для достижения разметки изображения области лица, сохраненного в формате `bmp`, которое `OutImage = cvCreateImage(cvSize(120, 120), 8, 3)`. Создает заголовок и выделяет пространство данных изображения размером 120 x 120.

После получения обнаруженного изображения необходимо выполнить обработку, включая преобразование цвета изображения, преобразование размера изображения и выравнивание гистограммы, прежде чем впоследствии можно будет выполнить обнаружение лица. Соответствующие функции обработки:

1. `cvCvtColor(img, gray, CV_BGR2GRAY)` — преобразование цвета.
2. `cvResize(gray, small_img, CV_INTER_LINEAR)` — преобразование размера изображения.
3. `cvEqualizeHist(small_img, small_img)` — выравнивание гистограммы серого изображения.

Мультимедийный программный интерфейс `DirectX`, предоставляемый компанией Microsoft, реализован на языке программирования Visual C++12 и с COM, что

позволяет лучше строить платформы машинного зрения для систем Windows. Интерфейс напрямую предоставляет ресурсы, необходимые для разработки высококачественных приложений, работающих в режиме реального времени. Преимущества технологии DirectX в том, что она позволяет обеспечить аппаратную независимость для разработчиков программного обеспечения, т.е. система не зависит от конкретного устройства видеозахвата.

Компонент обнаружения лиц на основе камеры использует компонент DirectShow SDK DirectX для захвата и обработки видео в реальном времени, с помощью модели под названием Filter Graph для управления всем потоком данных, поддерживая совместимость с VFW (Video For Windows) и обеспечивая превосходный интерфейс. Сравнение захвата видео в реальном времени компонентами OpenCV и DirectShow показывает, что компонент DirectShow гораздо более плавно обрабатывает видео с камеры. Основные функции компонента DirectShow для открытия камеры и обработки видео:

1. CCameraDS camera
2. Camera.OpenCamera()
3. IplImage *pFrame=camera.QueryFrame()

Анализ результатов испытаний

Эксперименты проводились на портативном компьютере DELL G15 5511, 15.6", WVA, Intel Core i7 11800H 2.3ГГц, 8-ядерный, 8ГБ DDR4, 512ГБ SSD, NVIDIA GeForce RTX 3050. Для обнаружения лиц на неподвижных изображениях экспериментальная выборка состояла из 46 цветных изображений, собранных из Интернета, с минимальным размером 469 x 447 и максимальным размером 1936 x 1288. Средние Precision (точность) составила 96 %, Recall (полнота) составила 93 %.

Видео обрабатывалось плавно, а результаты обнаружения полностью соответствовали требованиям реального времени. Распознавание лиц было проведено на видеофайлах из Интернета, подобных U-go-girl.

AVI, kissingyou.AVI. Средняя скорость обнаружения лиц на видео составила порядка 0,07 секунды. Средние Precision (точность) и Recall (полнота) практически совпали с аналогичными показателями для неподвижных изображений.

В ходе экспериментов было установлено, что основной причиной промахов в обнаружении является большой угол отклонения позы лица, поэтому правила обновления обучающих весов классификатора нуждаются в дальнейшем улучшении и совершенствовании. При обнаружении лиц важно отметить, что пиксели неподвижного изображения и резкость видео не должны быть слишком высокими, так как при слишком высоких значениях производительность обнаружения лиц в реальном времени значительно снижается, и скорость обнаружения также уменьшается.

Заключение

В данной работе, основываясь на принципе алгоритма обнаружения лиц AdaBoost, основанного на хараподобных признаках, каскадный классификатор лиц, обученный алгоритмом AdaBoost, использовался для разработки системы обнаружения лиц с помощью библиотеки компьютерного зрения OpenCV и VC++12. Было реализовано обнаружение и маркировка лиц на основе неподвижных изображений, видео с камеры и видео в формате AVI, соответственно, а также отображение и хранение в реальном времени изображений маркированных областей лица. Экспериментальные результаты показывают, что обнаружение лица на основе AdaBoost с использованием библиотеки компьютерного зрения OpenCV имеет преимущество в скорости обнаружения и высокой производительности в реальном времени, а метод реализации прост и практичен, что может быть использовано для распознавания и отслеживания лиц, анализа движения и мониторинга окружающей среды в реальном времени.

ЛИТЕРАТУРА

1. Маркова А.В., Афанасьев А.Г., Афанасьев Г.И. Классификация на основе алгоритма дерева решений для машинного обучения // E-Scio. 2022. № 11 (74). С. 106–118.
2. Chen, S.Y., Liu S. OpenCV-based implementation of computer vision technology // Beijing: Science Press. 2008.
3. Su J., Lin T., Wang K., et al. Face detection and tracking in video // Applied Science and Technology. 2011. 38(3). P. 5–11.
4. Hua Y., Peng H., Gu J. Face detection method by boosting covariance feature // Computer Engineering and Applications. 2010. 46(18). P. 186–189.
5. Guo L., Wang Q. Research of face detection based on AdaBoost algorithm and OpenCV implementation // Journal of Harbin University of Science and Technology. 2009. 14(5). P. 123–126.
6. Cui X. Research and implementation of face detection system based on AdaBoost algorithm // Changchun: School of Computer Science and Technology, Jilin University. 2008.
7. Xu Q., Wang J. Real-time face detection and statistics based on video image // Computer and Modernization. 2010(1). P. 120–123.