

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Жилкин Федор Игоревич

Анализ методов поиска лиц в области цифровой криминалистики

Курсовая работа

Научный руководитель:
к. т. н., доц. Литвинов Ю. В.

Санкт-Петербург
2019

Оглавление

| | |
|--|-----------|
| Введение | 3 |
| 1. Постановка задачи | 5 |
| 2. Обзор | 6 |
| 2.1. Haar's cascades | 6 |
| 2.1.1. Выбор характеристик Хаара | 7 |
| 2.1.2. Интегральное изображение | 9 |
| 2.1.3. Adaboost Training | 10 |
| 2.1.4. Каскадный классификатор | 11 |
| 3. Итоги | 13 |
| Список литературы | 14 |

Введение

С продвижением технического прогресса связана безопасность людей, ведь благодаря новейшим технологиям стало куда легче и быстрее пресекать преступления, изолировать людей, совершивших злодеяния. Одно из важнейших обстоятельств является появление камер видеонаблюдения на улицах, в метро и общественных местах. Таким образом службы безопасности разных стран могут отслеживать преступников для их дальнейшей изоляции. Но количество информации, поступающей с этих камер растет с каждым днем и становится необходимым автоматизация обработки полученных видео и фотографий. Решить эту проблему помогает автоматическое распознавание лиц для дальнейшей их верификации (определение, что два лица на разных фотографиях принадлежат одному и тому же человеку), кластеризации (разбиение лиц на фотографиях по группам, каждая из которых соответствует одному человеку) и классификация (выяснение, принадлежит ли данное лицо человеку, находящемуся в базе данных).

Любая задача, связанная с лицами, начинается с их обнаружением на фотографиях или видео. Данная работа посвящена как раз распознаванию лиц (face detection). В ней будут рассмотрены основные методы распознавания лиц, проведена сравнительная характеристика и будут выяснены лучшие решения для работы в сфере криминалистики на основе двух ключевых факторов:

- эффективность решения в условиях слабой освещенности, наличия угла поворота лица и наличия вещей, частично закрывающих лицо;
- так как нет гарантии, что криминалисты будут использовать графические ускорители при работе с программой, а тяжелые решения (например сверточные глубокие нейронные сети) требуют большой вычислительной мощности, что ведет к долгой работе на CPU, появляется немаловажный аспект быстродействия готового продукта;

Данная работа проводится совместно с компанией Belkasoft, специализирующейся на создании программного обеспечения в сфере киберкриминалистики. Также в данной работе принимает участие Чернявский Олег, который занимается кластеризацией лиц (face clustering).

1. Постановка задачи

Целью работы является обзор и сравнение существующих методов распознавания лиц людей на фотографиях и видеозаписях на основе факторов, приведенных в введении. Также по результатам исследования, совместно с компанией BelkaSoft, будет создана программа кластеризации лиц. Для успешного выполнения данной цели были поставлены следующие задачи:

- изучить предметную область задачи и теоретическую часть;
- проанализировать архитектуры систем нахождения лиц;
- исследовать алгоритмы нахождения лиц на фотографиях (с возможной их модификацией);
- провести анализ работы алгоритмов на данных выше критериях;
- составить тестовый набор данных (датасет) фотографий, представляющих интерес для криминалистов;

2. Обзор

На данный момент существует много различных решений обнаружения лиц на фотографиях. На данном этапе работы проанализирован каскадный классификатор Хаара (Haar's cascades)

2.1. Haar's cascades

Каскадный классификатор Хаара, впервые описанный в 2001 году в оригинальной статье [1] (иногда называемый как каскад ускоренных классификаторов, работающих с хаароподобными функциями), представляет собой особый случай ансамблированного обучения [3], называемый ускорением (boosting). Как правило он основан на классификаторах Adaboost [2] (и других моделях, таких как Real Adaboost, Gentle Adaboost или Logitboost). Каскадные классификаторы обучаются на нескольких сотнях изображений, которые содержат объект, который мы хотим обнаружить, и других изображениях, которые не содержат этих объектов. Как мы можем определить, есть ли на изображении лицо или нет? Существует алгоритм, называемый «средой обнаружения объектов Viola – Jones» (Viola–Jones object detection framework), который включает в себя все этапы, необходимые для обнаружения лица:

- выбор характеристик (признаков) Хаара, особенности, полученные из вейвлетов¹ Хаара.
- интегральное представление изображения
- Adaboost Training
- каскадный классификатор

¹Вейвлет — математическая функция, позволяющая анализировать различные частотные компоненты данных. Источник: Свободная энциклопедия, url: <https://ru.wikipedia.org/wiki/Вейвлет>, дата обращения: 11.12.2019

2.1.1. Выбор характеристик Хаара

Есть некоторые общие черты, которые мы находим на человеческих лицах:

- область темных глаз по сравнению с щеками
- яркая область переносицы по сравнению с глазами
- какое-то конкретное расположение глаз, рта, носа и т.д.

Процесс извлечения этих характеристик продемонстрирован на Рис. 1.

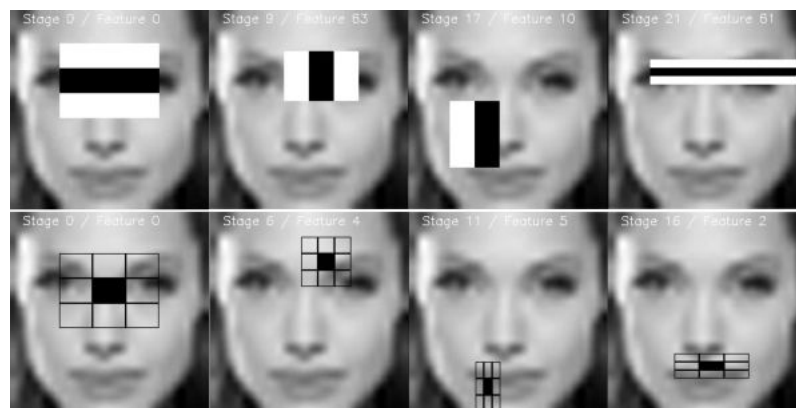


Рис. 1: Извлечение характеристик (feature extraction)

В этом примере первая характеристика (прямоугольник/признак) измеряет разницу в интенсивности между областью глаз и областью щек. Значение характеристики просто вычисляется путем суммирования пикселей в черной области и вычитания пикселей в белой области.

$$RectangleFeature = \sum pixels_{blackarea} - \sum pixels_{whitearea}$$

Затем мы применяем этот прямоугольник как сверточное ядро по всему нашему изображению. Мы должны применить каждое ядро со всеми возможными размерами и положениями (поворотами, наклонами). Простые изображения размером 24*24 пикселей обычно дают более 160 000 характеристик, каждая из которых состоит из суммы/вычитания значений пикселей. В вычислительном отношении это было бы

невозможно для обнаружения лица в реальном времени. Итак, как мы можем ускорить этот процесс?

- как только хорошая область была идентифицирована прямоугольником, бесполезно запускать функцию по совершенно другой области изображения. Это может быть достигнуто Adaboost'ом.
- вычислить прямоугольные характеристики, используя принцип интегрального изображения, который намного быстрее. Мы рассмотрим это ниже.

Существует несколько типов прямоугольников (изображены на Рис.2), которые можно применять для извлечения характеристик Хаара. Согласно оригинальной статье:

- двухугольная характеристика – это разница между суммой пикселей в двух прямоугольных областях, используемая в основном для обнаружения краев (a, b)
- трехугольная характеристика – вычисляется как сумма в двух внешних прямоугольниках, вычтенных из суммы в центральном прямоугольнике, используется в основном для обнаружения линий (c, d)
- четырехугольная характеристика – это разница между диагональными парами прямоугольника (e)

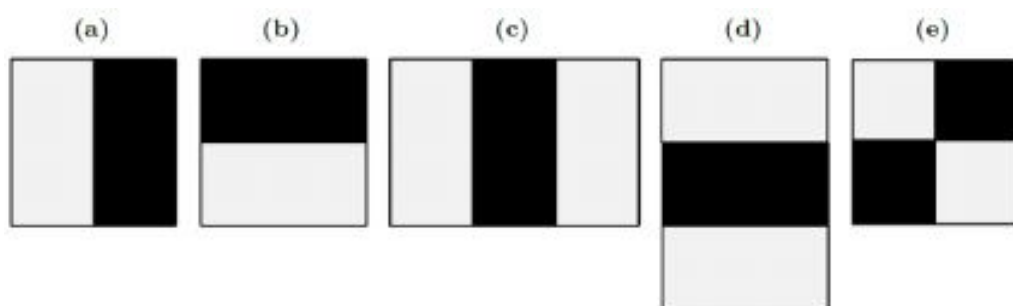


Рис. 2: Прямоугольники Хаара (Haar's rectangles)

Теперь, когда характеристики выбраны, мы применяем их к набору обучающих изображений, используя классификацию Adaboost, которая представляет собой набор слабых классификаторов для создания точной модели ансамбля (ensemble model). С 200 функциями достигается точность 95%. Авторы статьи выбирали 6 000 функций.

2.1.2. Интегральное изображение

Вычисление прямоугольных характеристик в стиле сверточного ядра может быть очень долгим. По этой причине авторы, Виола и Джонс, предложили промежуточное представление для изображения – интегральное изображение. Роль интегрального изображения заключается в том, чтобы просто вычислить любую прямоугольную сумму, используя только четыре значения. Предположим, мы хотим определить прямоугольные характеристики в данном пикселе с координатами (x, y) . Затем пиксельное интегральное изображение вычисляется как сумма пикселей выше и слева от данного пикселя.

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

где $ii(x, y)$ интегральное изображение и $i(x, y)$ обычное изображение.

Когда мы вычислим целое интегральное изображение, возникает рецидив формы, который требует только одного прохода по исходному изображению. Действительно, мы можем определить следующую пару повторений:

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

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

где $s(x, y)$ сумма накопленных строк и $s(x - 1) = 0, ii(-1, y) = 0$.

Чем это может быть полезно? Рассмотрим область D (см. Рис. 3), для которой мы хотели бы оценить сумму пикселей. Мы определили 3 других региона: A, B и C.

- Значение интегрального изображения в точке 1 – это сумма пикселей в прямоугольнике A, значение в точке 2 – сумма A + B

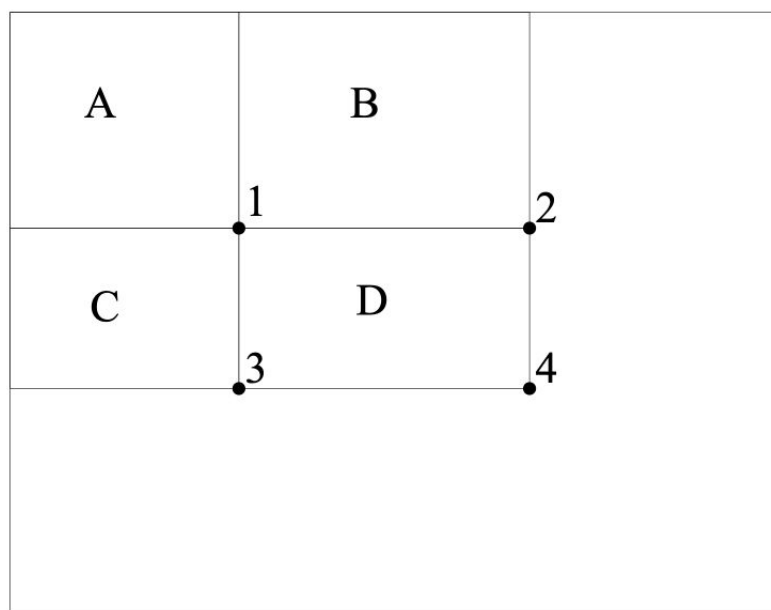


Рис. 3: Регионы в области D

- Значение в точке 3 – $A + C$
- Значение в точке 4 – $A + B + C + D$

Следовательно, сумма пикселей в области D может быть просто вычислена как: $4 + 1 - (2 + 3)$, и за один проход мы вычислили значение внутри прямоугольника, используя только 4 ссылки на массив.

Нужно просто помнить, что прямоугольники – довольно простые характеристики на практике, но достаточные для обнаружения лица. Управляемые фильтры имеют тенденцию быть более гибкими, когда речь идет о сложных проблемах.

2.1.3. Adaboost Training

Обучение классификатора с помощью Adaboost. Учитывая набор размеченных тренировочных образов (положительных или отрицательных), Adaboost используется для:

1. выбора небольшого множества характеристик
2. обучения классификатора (обучение с учителем) [4]

Поскольку предполагается, что большинство функций из 160000 не имеют никакого значения, слабый алгоритм обучения, на основе которого мы строим модель (boosting model), предназначен для выбора одной характеристики, которая разделяет лучше всего отрицательные и положительные примеры.

2.1.4. Каскадный классификатор

Хотя процесс, описанный выше, довольно эффективен, основная проблема остается. На изображении большая часть изображения – это область без лица. Придавать одинаковую важность каждой области изображения не имеет смысла, поскольку мы должны сосредоточиться в основном на областях, которые, скорее всего, содержат изображение лица. Виола и Джонс достигли повышенной скорости обнаружения, сократив при этом время вычислений с помощью каскадных классификаторов.

Ключевая идея состоит в том, чтобы не рассматривать области изображения, которые не содержат граней (областей резкой смены интенсивности цвета). Поскольку задача состоит в том, чтобы правильно идентифицировать лицо, мы хотим минимизировать количество ложных отрицательных результатов, то есть области, которые содержат лицо и не были идентифицированы как с лицом.

Ряд классификаторов применяется к каждой области изображения. Эти классификаторы являются простыми деревьями решений:

- если первый классификатор положительный, мы переходим ко второму
- если второй классификатор положительный, мы переходим к третьему
- ...

Любой отрицательный результат в некоторой точке приводит к отклонению области как потенциально содержащей лицо. Первоначальный классификатор исключает большинство отрицательных примеров при

низких вычислительных затратах, а следующие классификаторы устраняют дополнительные отрицательные примеры, но требуют больших вычислительных усилий. Схематично этот процесс изображен на Рис. 4.

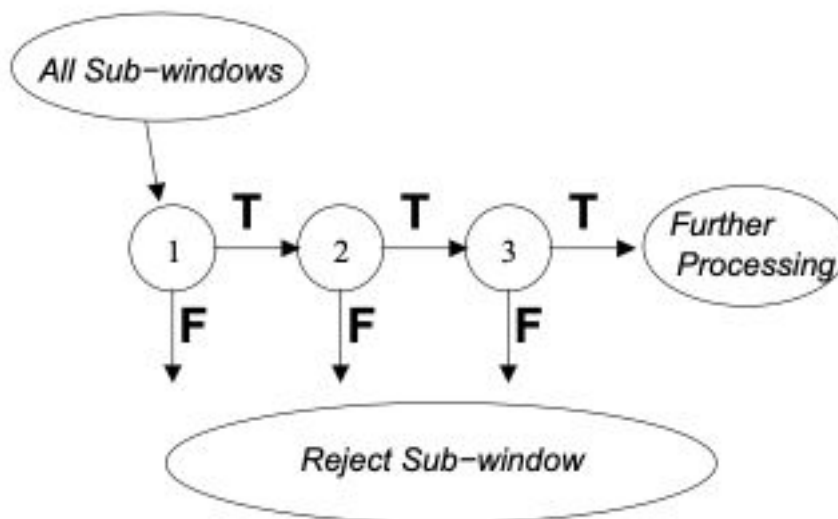


Рис. 4: Каскад (Haar's cascade)

Классификаторы обучаются с использованием Adaboost и настройкой порога (threshold), чтобы минимизировать ложную оценку. При обучении такой модели переменными являются:

- количество ступеней классификатора
- количество функций на каждом этапе
- порог каждого этапа

В библиотеке OpenCV² эта модель уже обучена на распознавание лиц. Рис. 5 демонстрирует работу алгоритма на изображении 600*600 пикселей, на котором содержится 25 лиц. Алгоритм обнаружил 23 лица и отработал за 0.68 секунд. Скорость работы алгоритма позволяет использовать его в системах непрерывного обнаружения лиц, например, на записях с камер видеонаблюдения.

²URL: <https://opencv.org/>, дата обращения: 10.12.2019

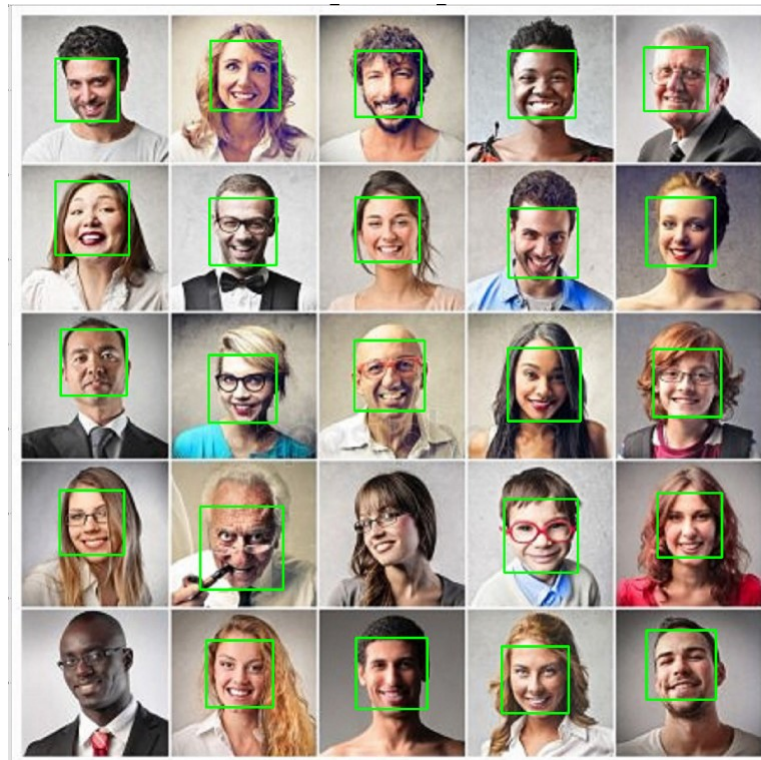


Рис. 5: Результат работы алгоритма

3. Итоги

На данном этапе работы была изучена теоретическая часть каскадных классификаторов и произведен тест этого алгоритма на фотографиях с лицами.

Планируется сделать:

- подготовить тестовый набор фотографий
- построить тестовое окружение
- протестировать алгоритмы и разобраться в теоретической части (HOG, MMOD и др.)
- сравнить эффективность работы алгоритмов на основе факторов, приведенных в введении
- подготовить обзор алгоритмов и решений

Список литературы

- [1] Paul Viola Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features // <https://www.cs.cmu.edu/>. — 2001. — URL: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf> (online; accessed: 11.12.2019).
- [2] Wikipedia. AdaBoost // Википедия, свободная энциклопедия. — 2019. — URL: <https://en.wikipedia.org/wiki/AdaBoost> (online; accessed: 11.12.2019).
- [3] Wikipedia. Ensemble learning // Википедия, свободная энциклопедия. — 2019. — URL: https://en.wikipedia.org/wiki/Ensemble_learning (online; accessed: 11.12.2019).
- [4] Wikipedia. Supervised learning // Википедия, свободная энциклопедия. — 2019. — URL: https://en.wikipedia.org/wiki/Supervised_learning (online; accessed: 11.12.2019).