

Санкт-Петербургский политехнический университет Петра Великого
Институт Информационных Технологий и Управления
Кафедра компьютерных систем и программных технологий

Отчёт по самостоятельно работе
по предмету «Теория распознавания образов»

ЛОКАЛИЗАЦИЯ И ОТСЛЕЖИВНИЕ ЛИЦ НА ИЗОБРАЖЕНИИ

Работу выполнил студент гр. 53501/3 _____ Мартынов С. А.

Работу принял преподаватель _____ Никитин К. В.

Санкт-Петербург
2015

Содержание

Введение	3
1 Признаки изображений	5
1.1 Признаки Хаара	5
1.2 Локальные бинарные шаблоны	11
1.3 Двухмерное косинусное преобразование	15
2 Методы бинарной классификации признаков	19
2.1 Композиции классификаторов	19
2.2 Адаптивный бустинг и метод Виолы-Джонса	21
2.3 Смеси гауссовых распределений	25
3 Нейронные сети	29
4 Экспериментальные исследования	32
Заключение	35
Список литературы	36

Введение

Локализация человеческого лица с последующей идентификацией давно находится в списке наиболее важных задач для исследователей в области систем машинного зрения и искусственного интеллекта. Множество исследований, проводимых ведущими научными центрами, так и не позволило создать универсальную систему компьютерного зрения, способную к локализации и распознаванию лица человека при различных условиях.

Причиной тому явилось сразу множество факторов, среди которых:

- высокая вариативностью лиц, обусловленной анатомическими особенностями людей;
- различный уровень освещенности объектов, зависящих от типа, количества и характеристик направленности источников света;
- необходимость обнаружения лиц, имеющих различное пространственное положение.

Зачастую особенности системы, использующей локализацию и распознавание лиц может накладывать дополнительные ограничения на скорость работы (близкой к реальному времени), аппаратным ресурсам (процессор, объём памяти) и экономичности (аккумулятора).

Кроме скоростных характеристик, от алгоритма требуется обеспечение малого (порядка 5%) количества ложных распознаваний. В системах, реализующих существующие методы распознавания, при увеличении уровня распознаваний свыше 90% наблюдается существенный рост числа ложных решений, что затрудняет их практическое использование. Прежде чем распознавать лицо, необходимо убедиться в его присутствии на изображении. Для чего применяются известные методы обнаружения и распознавания лиц на изображениях (метод главных компонент, нейронные сети, метод опорных векторов). Результативность применения метода определяется спецификой решаемой задачи. Поэтому построение метода распознавания лиц, обеспечивающего высокий

уровень достоверности решения при отсутствии ограничений на исходные изображения, является весьма актуальной задачей.

Целью данной работы является обзор основных методов обнаружения лиц (под обнаружением лиц на изображении будем понимать процесс локализации областей изображения, содержащих лица людей; границы искомых областей в общем случае размыты, однако чаще всего подразумевается минимальный описывающий прямоугольник), обеспечивающих повышение достоверности распознавания объектов анализа, снижение уровня ложных распознаваний, уменьшение времени обучения классификатора и времени предварительной обработки изображения.

1 Признаки изображений

Вычисление признаков изображения является одним из важнейших аспектов при распознавании визуальных образов. Признак в данном контексте это произвольные дескрипторы изображения, полученные в результате обработки исходных данных. Качество детектирования определяется информативностью выделенных признаков, а вычислительная сложность алгоритмов выделения – скоростью работы системы распознавания. В самом простом случае, признаком может являться яркость или цвета пикселей, составляющих исследуемое изображение.

Имеется множество различных примеров классификации признаков по типу, но обычно выделяют три основные категории:

- яркостные – базируются на перепадах яркости в пикселях;
- текстурные – описываются с помощью повторяющихся базовых элементов текстур;
- геометрические – описывают фрагменты фигур и контуров, присутствующих на изображении.

Тип вычисляемых признаков определяется содержимым изображения, которое надо описать. Обобщенное человеческое лицо, как графический паттерн, из-за своей сложности может быть описано сразу всеми типами признаков.

1.1 Признаки Хаара

Вейвлеты Хаара – это семейство базисных функций, открытое в начале XX века Альфредом Хааром. Вейвлеты Хаара ортогональны, обладают компактным носителем, хорошо локализованы в пространстве, являются кусочно-постоянными функциями с разрывами 1-го рода. Из-за своей простой и быстрой формулы расчета они повсеместно применяются в задачах анализа нестационарных сигналов.

Рассмотрим множество кусочно-постоянных функций, определенных на интервале $[0, 1]$, состоящем из 2^l подинтервалов равной длины. Пусть функции, определенные на всем единичном интервале, образуют пространство V^0 ; кусочно постоянные функции, одна часть каждой из которых определена на интервале $[0, \frac{1}{2}]$, а другая на $[\frac{1}{2}, 1)$ образуют пространство V^1 и т.д. Таким образом, пространство V^l содержит все кусочно-постоянные функции, определенные на всех 2^l подинтервалах единичного интервала. Базис Хаара представляет собой множество функций, определенных для каждого пространства V^l . Базисные функции называются масштабирующими функциями и порождаются сдвигами и растяжениями прямоугольного импульса.

$$\varphi(x) = \begin{cases} 1, & 0 \leq x \leq 1, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$\varphi_{l,t}(x) = 2^{l/2}\varphi(2^lx - t) \quad (2)$$

Масштаб определяется множителем $2^{l/2}$, а сдвиг задается параметром $t = 0 \dots 2^{l-1} - 1$. Интеграл от масштабирующей функции на интервале $[-\infty, +\infty]$ равен единице. Соответствующий набор вейвлет-функций Хаара, ортогональный масштабирующим функциям, определяется посредством смещения и растяжения своего материнского вейвлета.

$$\psi(x) = \begin{cases} 1, & 0 \leq x \leq \frac{1}{2}, \\ -1, & \frac{1}{2} \leq x \leq 1, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

$$\psi_{l,t}(x) = 2^{l/2}\psi(2^lx - t) \quad (4)$$

Интеграл от вейвлет-функции на интервале $[-\infty, +\infty]$ принимает нулевое значение.

Одномерный базис Хаара, заданный в V^2 , может быть распространен на плоскость двумя способами. Стандартная композиция получается, когда одномерное вейвлет-преобразование применяется для каждого измерения поочередно: новым базисом служат всевозможные тензорные произведения функций $\psi_{l,t}(\bullet)$. Нестандартный базис выводится путем смещений и растяжений двумерной масштабирующей функции (5) и трех двумерных материнских вейвлетов.

$$\varphi(x, y) = \varphi(x)\varphi(y) \quad (5)$$

$$\psi^m(x, y) = \begin{cases} \psi(x)\varphi(y), & m = 1, \\ \varphi(x)\psi(y), & m = 2, \\ \psi(x)\psi(y), & m = 3. \end{cases} \quad (6)$$

Такой подход имеет преимущества перед стандартной композицией, поскольку обеспечивает лучшее пространственное разрешение за счет некоторой избыточности базиса (рис. 1). При одномерном преобразовании расстояние между соседними вейвлетами $\psi_{l,t}(x)$ составляет 2^l , в стандартном двумерном базисе – 2^{l-1} , а в нестандартном – 2^{l-2} .

На рис. 2 схематично представлены материнские вейвлеты, полученные по формуле (6). Темная область соответствует отрицательным значениям вейвлета, светлая – положительным.

Признаки Хаара – признаки цифрового изображения, вычисляемые с помощью функций, схожих по своей структуре с двумерными вейвлетами Хаара. Специалистами было предложено использовать материнские вейвлеты Хаара в качестве ядер для свертки исходного изображения. Каждый вычисляемый признак H представлен комбинацией прямоугольников h_i . При подсчете признака окно h скользит по изображению, и вычисляется среднее значение

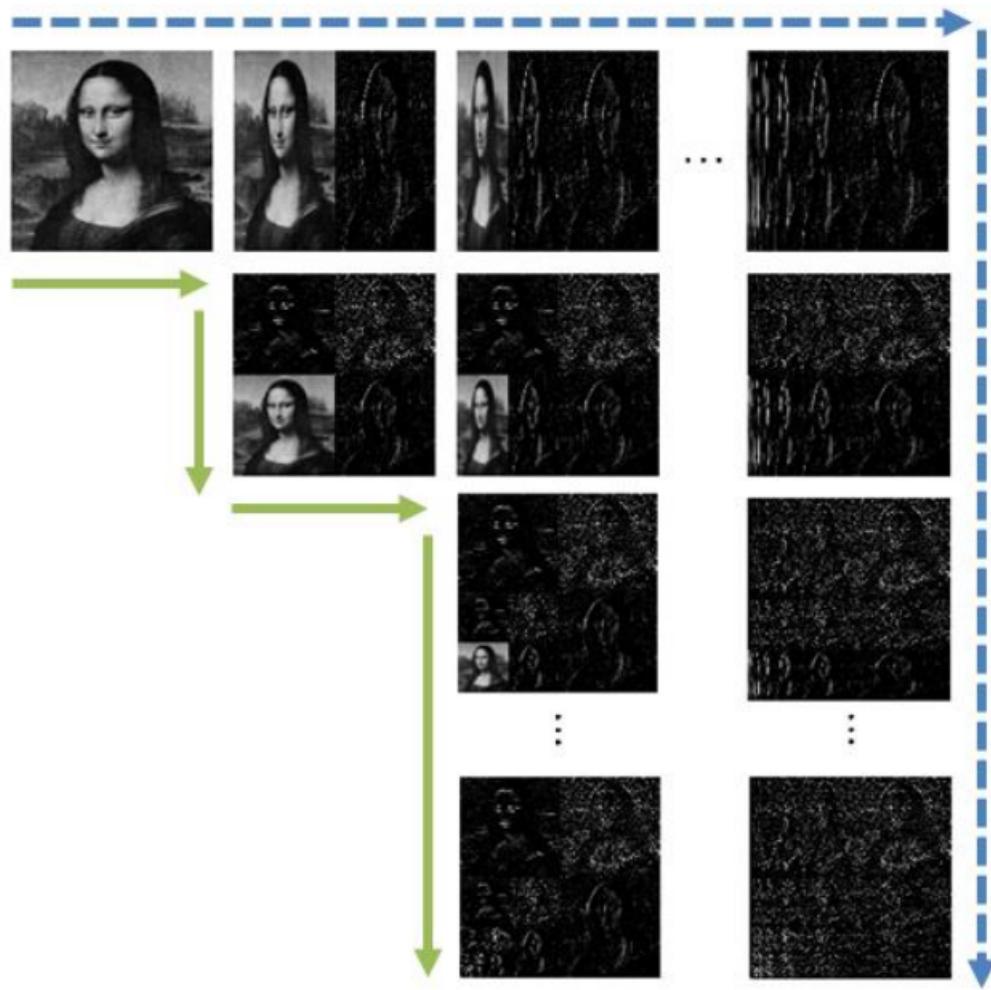


Рис. 1: Пример вейвлет-декомпозиции двумя способами: стандартным (пунктирная стрелка) и нестандартным (сплошная стрелка)

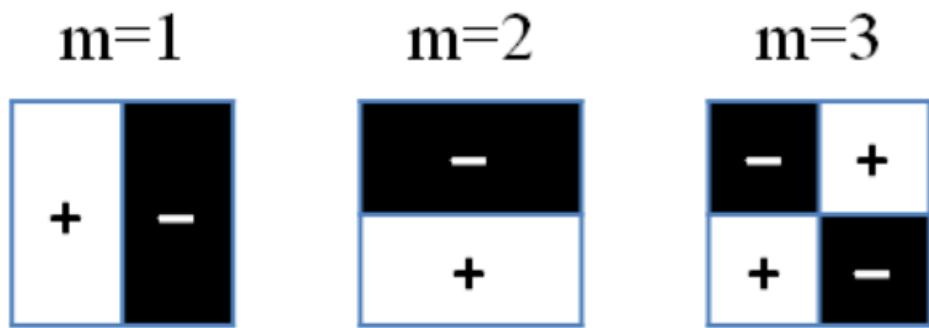


Рис. 2: Три двумерных материнских вейвлета

пикселей, покрываемых каждым прямоугольником – $\mu(i)$. Прямоугольникам соответствуют веса $w_i \in \{-1, +1\}$, определяющие с каким знаком войдет среднее значение пикселей в общую сумму. Согласно свойству интеграла от масштабирующих функций, веса удовлетворяют условию нулевой суммы:

$$\sum_{k=1}^N w_i = 0 \quad (7)$$

Выражение для вычисления признака имеет следующий вид:

$$H = \sum_{k=1}^N w_i \mu_i \quad (8)$$

На рисунке 3 изображены примеры расширенных признаков Хаара, предложенных Виолой и Джонсом (верхний ряд) и Ленхард (нижний ряд).

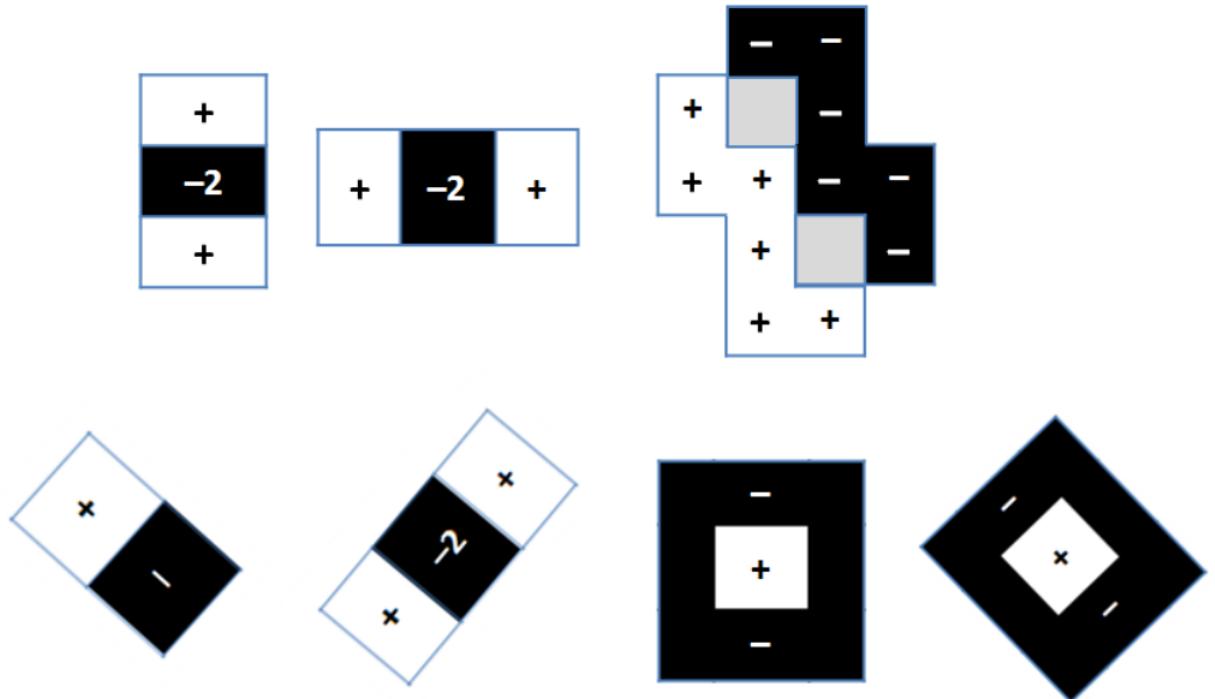


Рис. 3: Расширенный набор признаков Хаара

В пользу признаков Хаара говорит наличие для них эффективных вычислительных схем. Виола и Джонс предлагают использовать интегральное изображение для ускорения процедуры подсчета признаков. Интегральное изображение \bar{I} это матрица той же размерности, что и исходное изображение I , элементы которой определяются следующим образом:

$$\bar{I}(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} I(x', y') \quad (9)$$

Тогда среднее значение μ_i для прямоугольника h_i с вершинами в точках A_i, B_i, C_i, D_i с площадью S_i (рис 1) может быть посчитано за четыре арифметические операции над пикселями интегрального изображения:

$$\mu_i = (\bar{I}(A_i) + \bar{I}(B_i) - \bar{I}(C_i) - \bar{I}(D_i))/S_i \quad (10)$$

Для признаков Ленхард, повернутых на 45° , также существует быстрая схема вычислений. Для этого используется понятие таблицы суммированных областей, аналогичное определению интегрального изображения, выраженного формулой (9). Для вычисления таких признаков Ленхард вводится понятие таблицы суммированных областей, которые повернуты на угол, вычисляемой по следующей формуле:

$$\bar{\bar{I}}(x, y) = \sum_{\substack{x' \leq x \\ x' \leq x - |y' - y|}} I(x', y') \quad (11)$$

Среднее значение μ_i для прямоугольника h_i с вершинами в точках A_i, B_i, C_i, D_i с площадью S_i и углом поворота $\alpha_i = 45^\circ$ может быть также вычислено по формуле (10) для $\bar{I} = \bar{\bar{I}}$.

1.2 Локальные бинарные шаблоны

Порядковое измерение контраста (ordinal contrast measure) является порождением простого концепта: относительное важнее абсолютного. При анализе различных изображений мы можем отмечать значительные изменения абсолютных величин: цветов, яркости, текстурных элементов. Однако, взаимные порядковые отношения между соседними величинами, отражающие структуру изображенных объектов, подвержены меньшим изменениям. Порядковое кодирование контраста (ordinal contrast encoding) используется для вычисления существенных контрастных различий (contrast polarity) между двумя пикселями или регионами изображений. Оператор ставит в соответствие паре пикселей 1, если первый пиксель ярче второго или 0, в противном случае. Такой код удобен для вычислений, а информационная энтропия измерений максимальна, поскольку для случайных паттернов практически равновероятны значения 0 и 1.

Операция порядкового кодирования контраста $T \equiv T(X)$ обладает следующими свойствами инвариантности:

$$T(X) \equiv T(X + a) \equiv T(X \times a) \equiv T(X^a), \quad (12)$$
$$a \equiv const > 0$$

Census-преобразование (рис. 4) проецирует локальное окружение заданного пикселя в битовую строку. Полученная битовая строка может быть использована в качестве дескриптора области изображения. Обозначим через (x_c, y_c) координаты центрального пикселя, (x_p, y_p) – координаты смежного пикселя, $f(\bullet)$ – значение яркости пикселя в точке с заданными координатами. Тогда census-преобразование может быть представлено следующим образом:

$$T(X) = \otimes_{p=0}^{p-1} s(f(x_p, y_p) - f(x_c, y_c)) \quad (13)$$

$$s(x) = \begin{cases} 1, & x < 0, \\ 0, & x \geq 0. \end{cases} \quad (14)$$

$$T \left(\begin{array}{|c|c|c|} \hline 125 & 243 & 189 \\ \hline 45 & \textbf{196} & 98 \\ \hline 211 & 231 & 27 \\ \hline \end{array} \right) = \begin{array}{|c|c|c|c|} \hline & & & 2 \\ \hline & 3 & 0 & 1 & 0 & 1 \\ \hline & 4 & 0 & & 0 & 0 \\ \hline & 5 & 1 & 1 & 0 & 7 \\ \hline & & & 6 & & \\ \hline \end{array} = 01000110$$

Рис. 4: Census-преобразование

В качестве метрики схожести между census-образами пикселей изображений, используется расстояние Хэмминга, то есть рассчитывается число различающихся битов в соответствующих битовых строках. Основное ограничение census-преобразования в том, что оно охватывает малый регион анализируемого изображения вблизи центрального пикселя – этого может оказаться недостаточно для выделения ключевых текстурных характеристик.

Обобщенный случай census-преобразования – метод локальных бинарных шаблонов (ЛБШ), который был предложен Пиетикайненом. Количество соседних точек, участвующих в построении дескриптора, и их удаленность от центрального пикселя определяются параметрами P и R соответственно:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(f(x_p, y_p) - f(x_c, y_c))2^P \quad (15)$$

Каждому анализируемому множеству пикселей оператор ЛБШ ставит в соответствие натуральное число, определяющееся из равенства (15). Принципиальным отличием ЛБШ от Census-преобразования является то, что

ЛБШ-преобразование может применяться к соседним пикселям, лежащим на некоторой окружности радиуса $R \geq 1$ с центром в пикселе x_c (рисунок 5). Координаты пикселей x_p могут быть рассчитаны исходя из соотношений:

$$s(x) = \begin{cases} x_p = x_c + R \cos \frac{2\pi p}{P} \\ y_p = y_c - R \sin \frac{2\pi p}{P} \end{cases} \quad (16)$$

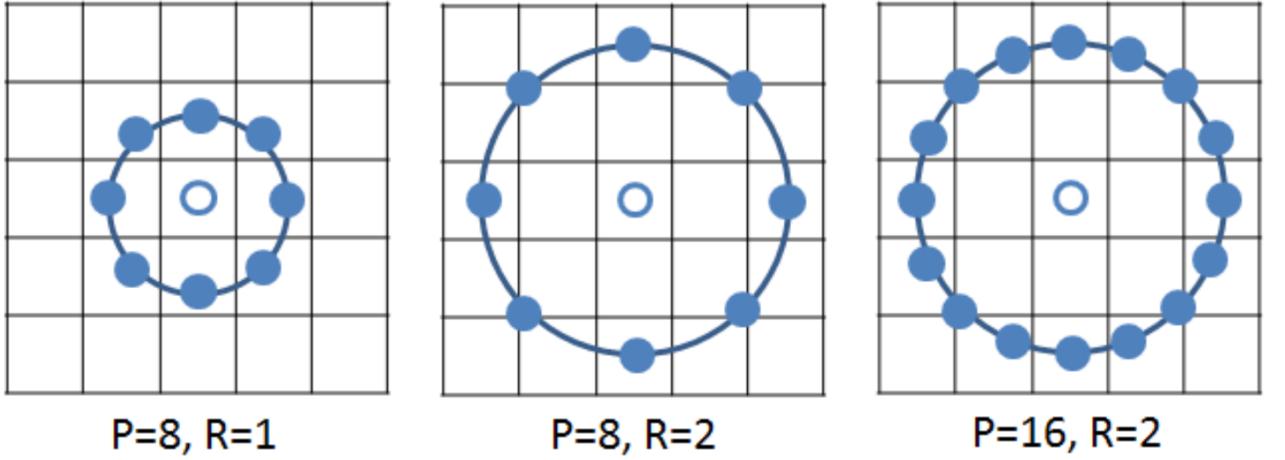


Рис. 5: LBP-преобразование с различным радиусом и количеством соседей

Если координаты точки (x_p, y_p) не соответствуют центру какого-либо пикселя изображения, то значение яркости в данной точке может быть получено с помощью интерполяции. Одним из самых простых и распространенных способов является билинейная интерполяция. Если положить, что четыре ближайших к (x_p, y_p) пикселя лежат на вершинах единичного квадрата, $x = x_p - |x_p|$, $y = y_p - |y_p|$, то значение яркости в точке (x_p, y_p) может быть вычислено по значениям пикселей в вершинах квадрата:

$$f(x_p, y_p) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix} \quad (17)$$

При $P=8, R=1$ формула ЛБШ принимает классическую форму. Многомасштабный анализ, основанный на ЛБШ, может производиться двумя способами.

В первом случае меняется радиус R оператора ЛБШ-преобразования. Во втором случае уменьшается размерность исходного изображения с последующим применением ЛБШ-оператора фиксированного радиуса. Оператор (15), строго говоря, не является инвариантным к поворотам исходного изображения. Предлагается проецировать область пикселя в битовую строку, записанную как циклический код. Случайно совершая циклические перестановки, ищется совпадение с одним из наперед заданных бинарных шаблонов. Оператор $LBD_{P,R}^{ri}$ возвращает номер совпавшего с битовой строкой бинарного шаблона. Для случая, когда $P = 8$ имеется ряд унифицированных шаблонов (uniform patterns), которые получаются из циклических сдвигов нескольких категорий битовых строк (рисунок 6):

1. Линии (lines): 01111111, 00000001.
2. Углы (corners): 00111111, 00011111, 00001111, 00000011.
3. Край (edge): 00000111.

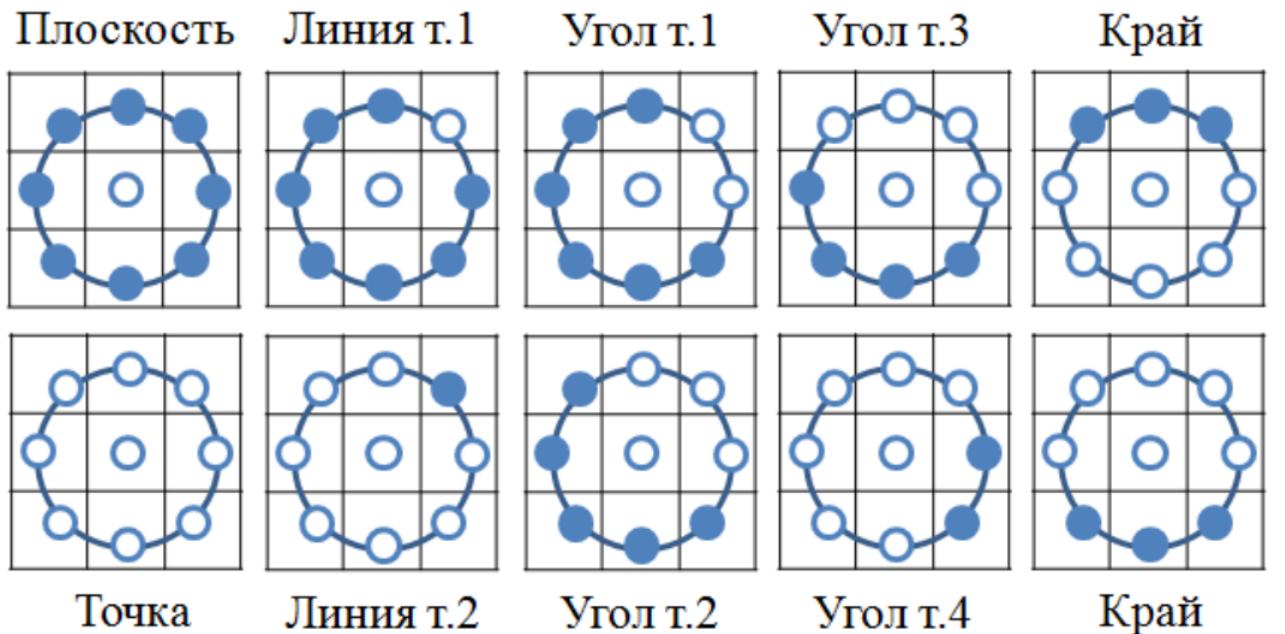


Рис. 6: Примеры унифицированных шаблонов

Перечисленные категории образуют 56 паттернов, к ним добавляется еще два шаблона, которые не варьируются циклическим смещением: 11111111 – плоскость (flat), 00000000 – точка (spot). Оператор, использующий 58 унифи-

цированных шаблонов для описания области изображения, обозначается как $LBD_{P,R}^{ri2u}$.

В качестве дескриптора для анализа изображений используется гистограмма ЛБШ-образов пикселей. Пусть область значений A оператора $LBD_{P,R}^{ri2u}$ разбита на равные промежутки: $A = \cup_i (a_{i-1}, a_i]$

$$n_{P,R}(i) = \sum_{j=1}^N \mathbf{1}_{\{LBD_{P,R}^{ri2u}(x_{c_j}, y_{c_j}) \in (a_{i-1}, a_i]\}} \quad (18)$$

$n_{P,R}(i)$ – число значений, попавших в i -ый интервал. Пусть $N = |A|$. Тогда нормализованная гистограмма ЛБШ-признаков принимает следующий вид:

$$h_{P,R}(i) = \frac{n_{P,R}(i)}{N \times (a_i - a_{i-1})} \quad (19)$$

1.3 Двухмерное косинусное преобразование

При выборе признаков необходимо учитывать следующие характеристики: устойчивость к условиям освещения, чувствительность к цифровому сжатию, вычислительная сложность. Помимо прочего, важным фактором является размер выборки признаков, т.е. количество однотипных признаков, выделяемых на одном статическом изображении. Косинусное преобразование – подходящий метод для выделения признаков изображения с точки зрения вышеперечисленных факторов, поскольку обладает высокой устойчивостью к условиям освещения высокочастотным шумам, является вычислительно простым алгоритмом.

$$f(u, v) = \frac{2}{N} C(u) C(v) \left(\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \right) \quad (20)$$

$$C(z) = \begin{cases} \frac{1}{\sqrt{2}}, & z = 0 \\ 1, & z \neq 0 \end{cases} \quad (21)$$

где $f(x, y)$ – квадратная область $N \times N$ исходного изображения, подвергающегося косинусному преобразованию, u, v – координаты вычисляемого признака.

Применение двухмерного дискретного косинусного преобразования (ДКП-2) ко всей области изображения не позволяет извлекать информацию о частотных характеристиках отдельных областей изображения (рисунок 7). Кроме того, для некоторых вероятностных алгоритмов требуется большое число признаков, отражающих локальную информацию, для эффективного обучения. Рассмотрим применение ДКП-2 блочно, с окном фиксированного размера и постоянной величиной сдвига S (рисунок 8), как в алгоритме JPEG.

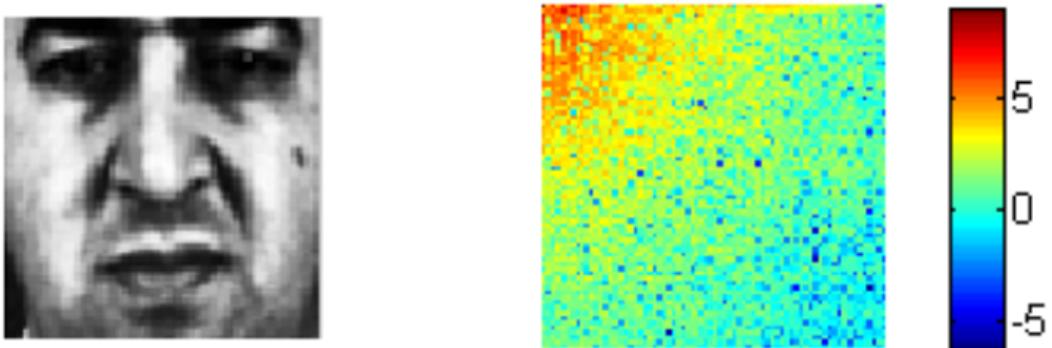


Рис. 7: Пример вычисления ДКП-2: оригинальное изображение (слева), абсолютные значения коэффициентов ДКП-2 (справа)

Так, для изображения размером 96×96 пикселей оконное ДКП-2 при $N = 8$ и $S = 4$ производит 484 64-мерных вектора коэффициентов.

Исследуя зависимость качества работы системы распознавания, построенной на признаках ДКП-2, от размера окна преобразования N и базы лиц, оказывается, что оптимальный размер окна преобразования сильно изменяется при тестировании на различных базах, от 8×8 (база MOBIO) до 20×20 (база

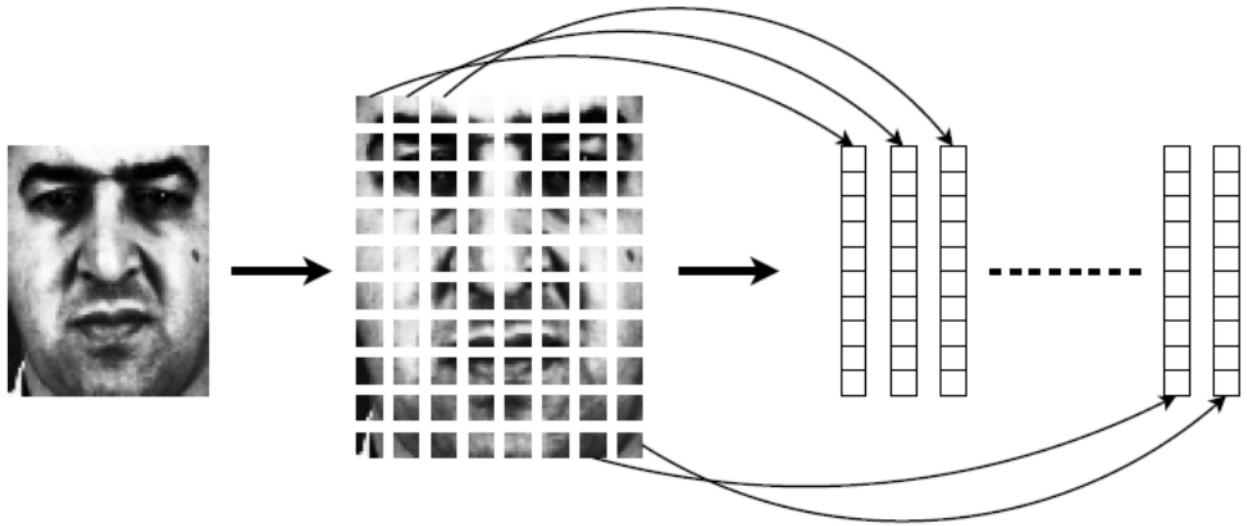


Рис. 8: Формирование матрицы признаков посредством оконного ДКП-2

SCface), и зависит от условий, в которых была собрана конкретная база, в том числе и от размера изображений лиц.

Большая часть полезной информации сосредоточена в низкочастотных коэффициентах изображения, в то время как шумовые составляющие обычно выражаются высокочастотными. Поэтому в ряде случаев, связанных со сжатием изображения или извлечением информативных данных, размерность вектора признаков F экспериментально ограничиваются сверху.

На этапе формирования векторов признаков оконного ДКП-2 (рисунок 8) каждая матрица признаков, получаемая путем применения ДКП-2 к локальной области, разворачивается в многомерный вектор коэффициентов. Существует множество способов формирования такого вектора, наиболее распространенным является способ "zig-zag" (рисунок 9).

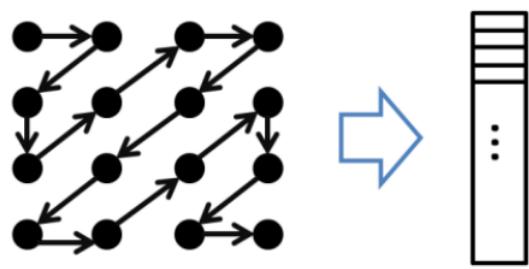


Рис. 9: Способ "zig-zag" развертки матрицы признаков в вектор

Особенность данного подхода заключается в том, что не отдается предпочтение какому-либо из двух направлений, в которых разворачивается набор коэффициентов ДКП-2.

2 Методы бинарной классификации признаков

Детектирование лиц на изображении относится к задачам бинарной классификации. Пусть $X = \{x_i\}$ – множество произвольных областей изображения, $Y = \{-1, 1\}$ – множество меток классов, $k \in K$ – алгоритмы бинарной классификации, тогда процесс классификации областей может быть записан следующим образом:

$$X \xrightarrow{k} Y \quad (22)$$

Предположим, что для некоторых $\hat{x}_j \in \hat{X} \subset X$ известны ответы $\hat{y}_j \in \hat{Y} \subset Y$, $j = 1, \dots, L$. Отображение вида

$$(\hat{X} \times \hat{Y})^L \rightarrow k \quad (23)$$

называется алгоритмом обучения классификатора k .

Для оценки производительности классификатора вводится функция потерь $L(k, x)$ – величина ошибки алгоритма k на элементе x из множества X . При решении задач бинарной классификации часто используется функция потерь от одного аргумента:

$$L(k, x) = L(y * k(x)) \quad (24)$$

где y – метка класса элемента x .

2.1 Композиции классификаторов

Рассмотрим изображение X и функцию $f_i(x)$, где x – некоторая произвольная область изображения, а f_i – скалярная функция вычисления признака от заданной области. Согласно представленным в предыдущих разделах описаниям

видов признаков, f_i может вычисляться как свертка изображения с одним из вейвлетов Хаара по формуле (10)(22) или являться i -ым бином гистограммы признаков ЛБШ (19). Применительно к ДКП-2 f_i можно представить как i -ый коэффициент преобразования $F(u, v)$ при фиксированном положении окна.

Набор всевозможных функций вида (25) будем называть множеством слабых (элементарных) классификаторов

$$k_i(x, a_i) = k_i(f_i, x, p_i, \theta_i) = \begin{cases} 1, & p_i f_i(x) < p_i \theta_i \\ -1, & \text{иначе} \end{cases} \quad (25)$$

где θ – порог принятия решения, $p \in \{-1, 1\}$ определяет направление знака неравенства, $k_i \in K$, $a_i = \{f_i, p_i, \theta_i\}$ – набор указанных параметров. Применительно к задаче детектирования лиц на изображениях, множество K можно представить как совокупность независимых детекторов, анализирующих каждую область изображения. Поскольку заранее неизвестно, какие из признаков окажутся наиболее информативными для описания конкретного паттерна лица, то на этапе обучения алгоритма детектирования целесообразно проверить все доступные элементарные детекторы и попытаться сформировать один "сильный" детектор.

Процедура последовательного построения композиции элементарных классификаторов, когда каждый последующий классификатор стремится компенсировать недостатки композиции всех предыдущих, получила название бустинга (boosting). Алгоритм бустинга является жадным алгоритмом – на каждом шаге, компенсируя ошибку предыдущей композиции, он принимает локально оптимальное решение, будучи уверенным, что итоговое решение будет также оптимальным. Цель бустинга – построение композиции s_D слабых алгоритмов k_i :

$$s_D(x) = s(k_1(x), k_2(x), \dots, k_L(x)) \quad (26)$$

Процесс последовательного обучения композиции может иметь следующие критерии останова:

- построено заданное количество элементарных классификаторов L ,
- достигнута заданная точность на обучающей выборке,
- достигнутую точность не удается улучшить на протяжении T шагов.

Закономерным результатом развития данного подхода стал алгоритм адаптивного бустинга. Количество классификаторов в композиции стало произвольным, а каждый последующий классификатор строится на ошибочных ответах предыдущих (адаптивность), используя одну и ту же базу для обучения. Для алгоритма адаптивного бустинга впервые была доказана теорема бустинга, которая задавала оптимальный способ формирования композиции слабых классификаторов в виде взвешенной суммы.

2.2 Адаптивный бустинг и метод Виолы-Джонса

Рассмотрим три модификации алгоритма адаптивного бустинга, применяющиеся для построения детектора лиц:

- Дискретный адаптивный бустинг (Discrete Adaptive Boosting, DAB).
- Вещественный адаптивный бустинг (Real Adaptive Boosting, RAB).
- Плавный адаптивный бустинг (Gentle Adaptive Boosting, GAB).

Алгоритм DAB был предложен Фройндом и Шапире. При построении композиции классификаторов алгоритмом DAB используются два типа весов: коэффициенты взвешенного голосования классификаторов $c_m \in C$ и веса обучающих сэмплов $w_i \in W$. На каждой итерации алгоритм DAB увеличивает веса тех наблюдений x_i , на которых классификаторы чаще ошибаются. Доля решения классификатора в общем голосовании c_m определяется точностью его решений на выборке X . Формула взвешенного голосования DAB выглядит следующим образом:

$$s_D(x) = \sum_{m=1}^M c_m k_m(x) \quad (27)$$

Перед началом итеративного процесса веса объектов полагаются одинаковыми (28). На каждой итерации выполняется перенормировка весов с целью выполнения условия (29).

$$w_i = \frac{1}{N}, \quad i = 1, \dots, N \quad (28)$$

$$\sum_{i=1}^N w_i = 1 \quad (29)$$

Для каждого классификатора из множества К выполняется три шага:

1. Сначала на выборке X обучается классификатор $k_m(x, a_m)$ (в более общем случае $k_m = k_m(x, a_m, W)$):

$$k_m = \operatorname{argmax}_{a_m} \sum_{i=1}^N L(y_i w_i k(x_i, a_m)) \quad (30)$$

2. Затем, рассчитывается ошибка m-го классификатора на всем обучающем множестве.

$$\mathcal{E}_m = \frac{1}{N} \sum_{i=1}^N 1_{\{y_i \neq k_m(x_i)\}} \quad (31)$$

Согласно основной теореме бустинга, определяется вес классификатора k_m в финальной композиции:

$$c_m = \frac{1}{2} \ln \frac{1 - \mathcal{E}_m}{\mathcal{E}_m} \quad (32)$$

3. Наконец, обновляются веса для каждого сэмпла из базы обучения. При изменении весов используется экспоненциальная функция потерь:
 $L(k, x) = e^{-y*k(x)}.$

$$w_i = w_i e^{c_m 1_{\{y_i \neq k_m(x_i)\}}} \quad (33)$$

$$w_i = \frac{w_i}{\sum_{j=1}^N w_j} \quad (34)$$

RAB является вероятностным обобщением алгоритма DAB. Дискретное множество ответов $Y = \{-1, 1\}$ заменяется на непрерывную оценку $f_m(x)$, знак которой определяет класс распознанного объекта, а модуль – степень "уверенности" классификатора. Голосование классификаторов задается суммой вкладов f_m :

$$s_R(x) = \sum_{m=1}^M f_m(x) \quad (35)$$

На втором шаге m -ой итерации, после обучения классификатора на выборке X , рассчитывается оценка вероятности положительного класса:

$$p_m(X) = \hat{P}_w(y = 1|x) \quad (36)$$

Вклад m -го классификатора в общее решение определяется логит-функцией от вероятности положительного класса:

$$f_m(X) = \frac{1}{2} \ln \frac{p_m(x)}{1 - p_m(x)} \quad (37)$$

На третьем шаге формула обновления весов принимает следующий вид:

$$w_i = w_i e^{-y_i f_m(x_i)} \quad (38)$$

Главное отличие алгоритма GAB от RAB – это способ оценки вероятностей классов, определяющих функции слабых классификаторов. В RAB для этого используется логистическая регрессия (37), а в алгоритме GAB рассчитывается разность между вероятностями двух классов (39).

$$f_m(X) = \hat{P}_w(y = 1|x) - \hat{P}_w(y = -1|x) \quad (39)$$

В то время как оценка (37) является численно неустойчивой, функция, задаваемая формулой (39), всегда лежит в пределах $[-1, 1]$. Другим отличием является аппроксимация функции потерь: алгоритм RAB использует экспоненциальную оценку, GAB – квадратичную.

Используя вейвлеты Хаара в качестве признаков для слабых классификаторов, Пол Виола и Майкл Джонс предложили обучать методом адаптивного бустинга сильные алгоритмы и затем выстраивать их в однобокое дерево решений, которое называется каскадом. На каждом уровне каскада сильный классификатор выносит решение, является ли образец искомым паттерном (в нашем случае лицом) или нет (рис. 10).

Решение каскадного классификатора D может быть записано, как произведение ответов r сильных классификаторов H_i :

$$D = \prod_{i=1}^r H_i \quad (40)$$

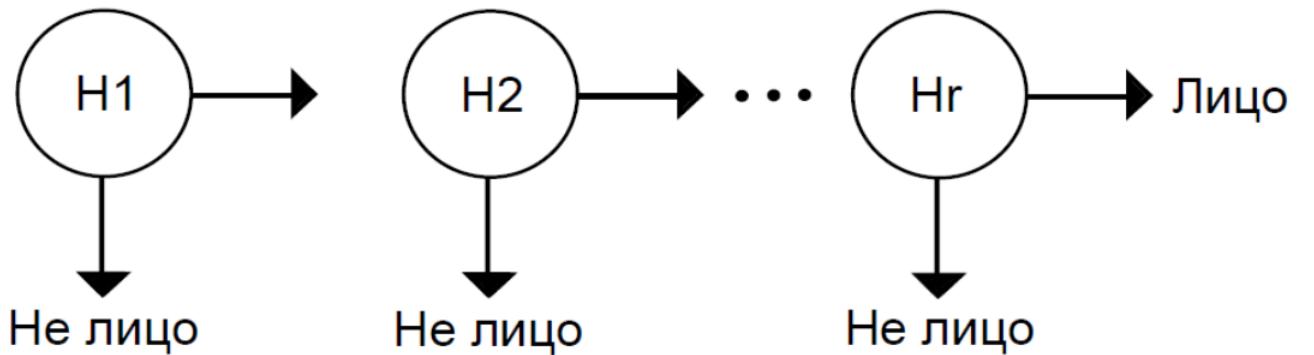


Рис. 10: Пример вырожденного дерева каскадов

Преимущество такого подхода перед одним сильным классификатором объясняется скоростью принятия решений. В качестве эксперимента сильный классификатор, обученный по методу AdaBoost и содержащий 200 признаков, сравнивался с каскадным классификатором, состоящим из 10 уровней по 20 признаков на каждом. В то время как TPR отличался на 2-3% в пользу сильного классификатора, производительность каскада была почти в 10 раз выше.

2.3 Смеси гауссовых распределений

Рассмотрим матрицу X , по столбцам которой расположены вектора признаков изображения. Предполагается, что распределение каждого элемента вектора признаков подчиняется нормальному, поскольку при построении тот подвержен влиянию огромного числа случайных факторов. В качестве генеративного метода, моделирующего характеристики наблюдаемого объекта и среды наблюдения, хорошо зарекомендовали себя смеси гауссовых распределений (СГР). Модель гауссовых смесей может быть записана следующим образом:

$$P(X|\theta) = \sum_{i=1}^K a_i \text{Norm}_x[\mu_i, \Sigma_i] \quad (41)$$

$$\text{Norm}_x[\mu_i, \Sigma_i] = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(X - \bar{\mu}_i)^T \Sigma_i^{-1} (X - \bar{\mu}_i)} \quad (42)$$

где X – D -мерный вектор случайных величин, $\bar{\mu}_i$ – вектор математического ожидания, Σ_i – ковариационная матрица, α_i – веса смеси: $\sum_{i=1}^N \alpha_i = 1$, $\theta = [\alpha_i, \bar{\mu}_i, \Sigma_i]$ – параметры гауссовой смеси, K – количество гауссоид.

Для построения СГР используется классический ЕМ-алгоритм (Expectation-maximization algorithm), максимизирующий правдоподобие модели на заданных обучающих данных. На Е-шаге вычисляется ожидаемое значение функции правдоподобия, при этом скрытые переменные рассматриваются как наблюдаемые (43). На М-шаге рассчитывается оценка максимального правдоподобия, таким образом, увеличивается ожидаемое правдоподобие, вычисляемое на Е-шаге. Производится переоценка вектора параметров, используя текущее значение вектора скрытых переменных (44)-(46).

$$\gamma_{nk} = \frac{\alpha_k \text{Norm}_{X_n}[\mu_k, \Sigma_k]}{\sum_{j=1}^K \alpha_j \text{Norm}_{X_n}[\mu_k, \Sigma_k]} \quad (43)$$

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} X_n \quad (44)$$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma_{nk} (X_n - \hat{\mu}_k)(X_n - \hat{\mu}_k)^T \quad (45)$$

$$\hat{\alpha}_k = \frac{N}{N_k} \quad (46)$$

где $N_k = \sum_{n=1}^N \gamma_{nk}$, N – количество элементов обучающей выборки. Пример итеративной сходимости алгоритма показан на рисунке 11.

При обучении модели гауссовых смесей необходимо провести инициализацию параметров модели перед первой итерацией. Не гарантируется нахождение глобального максимума в пространстве данных обучения, таким образом, результат обучения системы в значительной степени зависит от начальных значений. Предположим, что число гауссоид K заранее задано, тогда для инициализации параметров $[\alpha_i, \bar{\mu}_i, \Sigma_i]$ может использоваться случайная инициализация параметров модели, алгоритм k -средних, метод главных компонент и др.

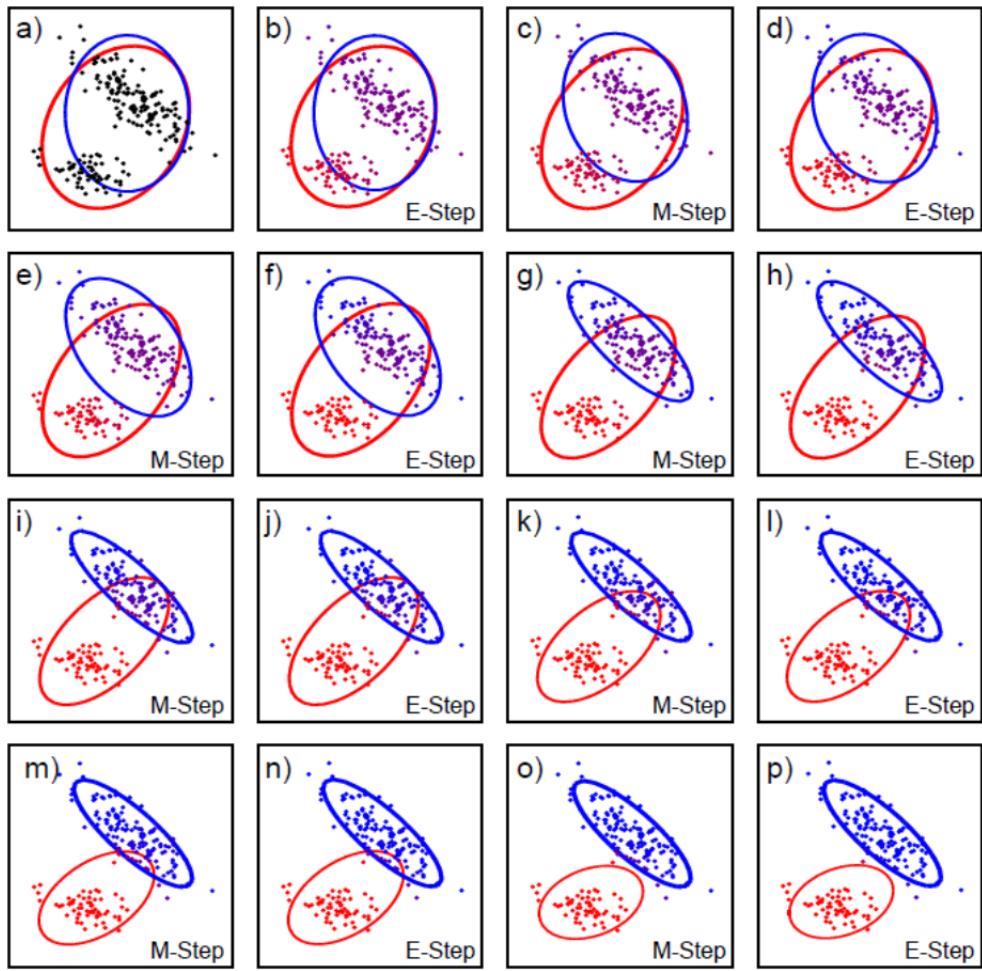


Рис. 11: a) Исходная модель. b) Е-шаг. Для каждой точки плоскости была рассчитана постериорная вероятность, порожденная из каждой гауссоиды (обозначена соответствующим цветом). c) М-шаг. Обновлены среднее, дисперсия и веса каждой гауссоиды согласно постериорным вероятностям Эллипсы отображают расстояние Махalanобиса, их толщина – вес гауссоиды. d)-t) Череда Е- и М-шагов.

При обучении СГР-модели важно предупредить случаи, когда эллипсы гауссоид могут превратиться в точку (проблема "схлопывания гауссоид") или покрыть всю выборку. Для предотвращения этого обычно вводят ограничения на элементы дисперсионной матрицы Σ_i .

3 Нейронные сети

Среди алгоритмов распознавания образов особое место занимают нейронные сети. Повышенный интерес к данной технологии в наши дни обусловлен успехами сетей в распознавании широкого класса паттернов, новыми численными методами для обучения многоуровневых нелинейных моделей и прогрессом в области производства недорогих и эффективных векторных процессоров для расчетов. При решении задач компьютерного зрения наибольшей популярностью пользуются сверточные нейронные сети (СНС). Модель СНС улучшает качество распознавания каскадного классификатора на ЛБШ за счет дополнительной фильтрации шумов.

В середине прошлого столетия ученые Торстен Визель и Дэвид Хьюбел исследовали зрительную кору головного мозга кошки и обнаружили, что существует два различных типа клеток. Так называемые простые клетки реагируют на прямые линии (полосы света, края стимулов) под определенным углом. Сложные клетки реагируют на движение линий (полос света, краев стимулов) в определенных направлениях. Спустя некоторое время, это открытие нашло применение в прикладной области информатики: Кунихика Фукусима воплотил идею простых и сложных клеток в неокогнитроне

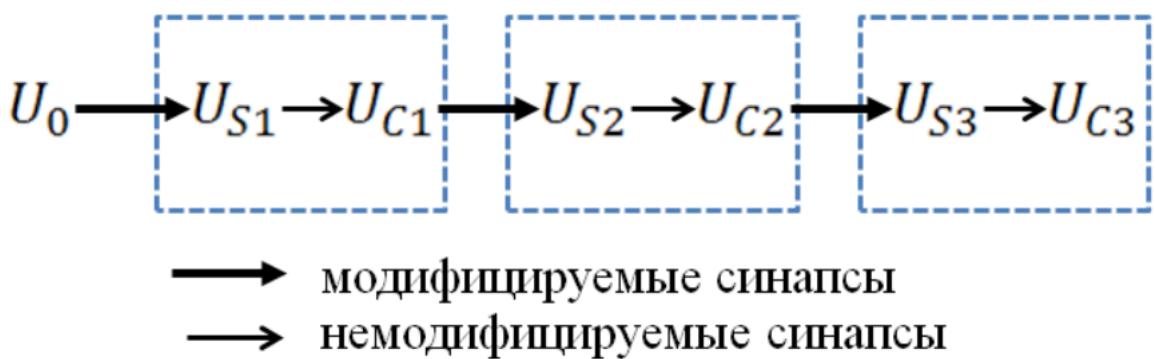


Рис. 12: Упрощенная структура неокогнитрона

Неокогнитрон состоит из ретины (сетчатки) U_0 и последовательности групп нейронов, каждая из которых, в свою очередь, содержит два типа нейронных слоев (рис. 12). Простые нейроны (из слоев U_S) принимают сигналы от

сложных нейронов предыдущего слоя и обнаруживают одинаковые признаки, но в разных местах сигнала. Сложные нейроны (из слоев U_C) объединяют отклики простых нейронов из своей группы и стремятся редуцировать зависимость реакции на признаки от их положения. Таким образом, неокогнитрон приобретает способность детектировать определенные комбинации признаков, независимо от их положения на ретине.

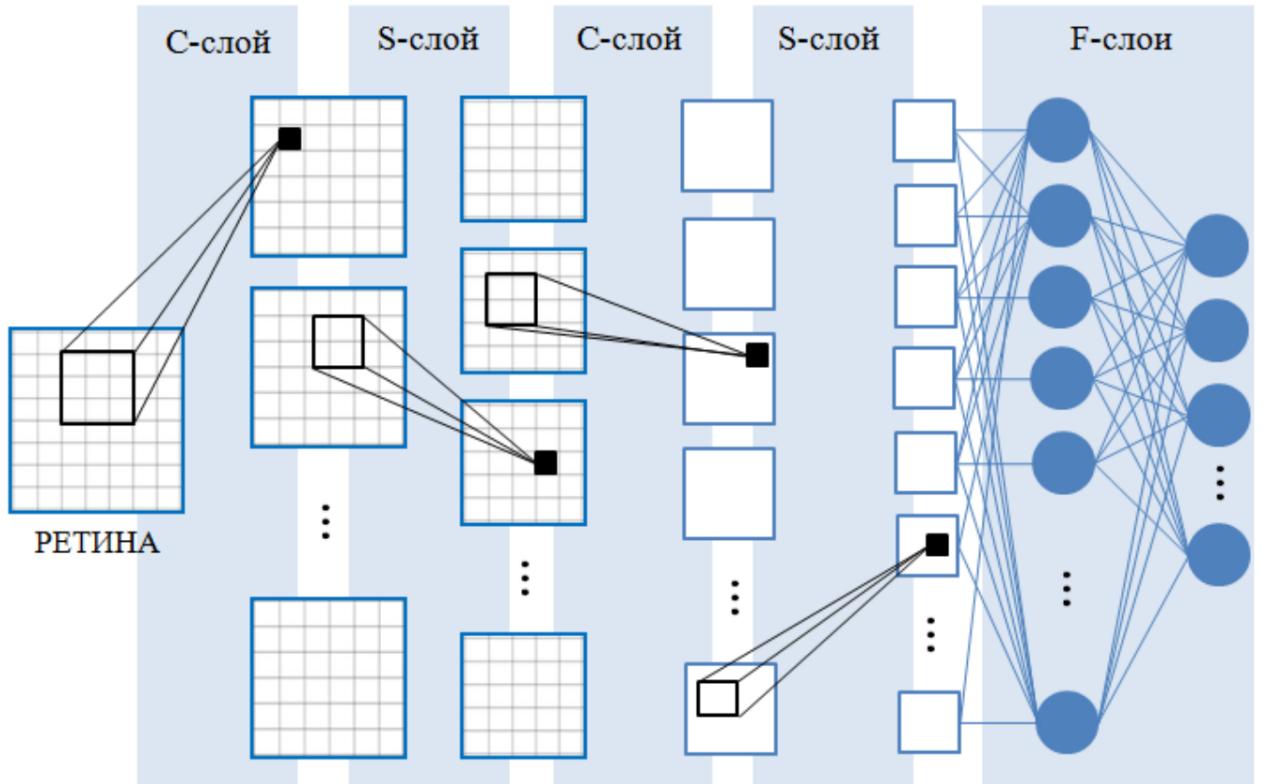


Рис. 13: Архитектура сверточной нейронной сети

Отличие неокогнитрона от многослойного персептрона заключается в трех инновационных для нейронных сетей на тот момент архитектурных принципах:

- локальные рецептивные поля,
- разделяемые веса,
- пространственная субдискретизация.

Развивая идею неокогнитрона, Ян ЛеКун вместе с соавторами предложили структуру классической сверточной нейронной сети (СНС) и алгоритм для ее обучения с учителем, а также применение СНС для решения широкого круга

задач распознавания образов. СНС, предложенные ЛеКуном (рис. 13), это иерархичные архитектурные многослойные идеи нейронные неокогнитроны, сети, в которых обеспечивающие сходятся некоторую три степень инвариантности представления данных.

Устройство СНС заключается в чередовании сверточных слоев (convolutional, C-слои), реализующих концепцию простых клеток зрительной коры с гибкими (обучаемыми) весами, и субдискретизирующих слоев (sub-sampling, S-слои), построенных по примеру сложных клеток кортекса с жесткими весами. Завершается подобная каскадная модель, как правило, полно связанными нейронными слоями (fully-connected, F-слои). Различные архитектуры СНС успешно используются во многих сложных приложениях, связанных с обработкой визуальных данных.

4 Экспериментальные исследования

Библиотека Dlib реализует алгоритмы машинного обучения и позволяет использовать описанные выше методы. Библиотека распространяется под свободной лицензией Boost Software License и включает в себя много полезных классов и функций. Основные возможности:

- Потоки (переносимый API для работы с потоками, межпотоковое взаимодействие по средствам pipe, локальные данные потока, пул потоков, механизм запуска глобальных функций в отдельном потоке и др.)
- Сетевое программирование (переносимый API для работы с TCP сокетами, TCP сервер, HTTP сервер и др.)
- Графический интерфейс пользователя (переносимый и потокобезопасный GUI API)
- Численные алгоритмы (матрицы и операции над ними, алгоритмы нелинейной оптимизации, целые числа с диапазоном ограниченным только ресурсами системы, генератор псевдослучайных чисел и др.)
- Алгоритмы машинного обучения
- Алгоритмы для вычислений и обучения байесовских сетей.
- Обработка изображений (чтение и запись формата Windows BMP, выделение границ, компьютерное зрение и др.).
- Алгоритмы сжатия и проверки целостности (CRC32, MD5, LZP и др.).
- Тестирование (потокобезопасный класс ведения лога, модельное тестирование, макросы проверки предусловий).
- Полезные классы общего назначения (разбор XML, разбор параметров командной строки, различные контейнеры, base64 и др.).

Для простоты, воспользуемся языком программирования Python (листинг 1). Этот скрипт умеет определять множество лиц на одном изображении.

Листинг 1: Пример простого скрипта для определения лица

```
1 import dlib
2 from skimage import io
3
4 img = io.imread('imagewithface.jpg')
5
6 detector = dlib.get_frontal_face_detector()
7 # Run the face detector, upsampling the image 1 time to find smaller faces.
8 dets = detector(img,1)
9
10 print "number of faces detected: ", len(dets)
11 win = dlib.image_window()
12 win.set_image(img)
13 win.add_overlay(dets)
14
15 raw_input("Hit enter to continue")
```



Рис. 14: Определение множества лиц

На рисунках 14 и 15 представлены результаты работы скрипта. Предложенный метод детектирования позволяет уверенно обнаруживать большую часть лиц на фотографиях, включая лица с большим углом отклонения.



Рис. 15: Определение одного лица

Для встраиваемых решений подобный подход может оказаться избыточным, но для классической обработки изображений на сервере с архитектурой x86_64 реализована оптимальная поддержка со стороны аппаратуры, результатом чего является высокое быстродействие и хороший показатель определения лиц.

Заключение

В данной работе рассмотрены наиболее распространенные методы выделения информативных признаков цифровых изображений, такие как признаки Хаара, ЛБШ, дискретное косинусное преобразование. Представлены базовые алгоритмы классификации признаков в постановке задачи детектирования. Приведён краткий обзор работы нейронной сети.

В практической части представлен скрипт на языке Python с использованием библиотеки dlib и результаты определения лиц на нескольких изображениях.

Список литературы

1. Вежневец В. Обнаружение и локализация лица на изображении.
2. Тимошенко Д.М. Методы автоматической идентификации личности по изображениям лиц, полученным в неконтролируемых условиях.
3. Vladimir Vezhnevets. Method For Localization Of Human Faces In Color-Based Face Detectors And Trackers.
4. Sung H. Yoon, Gi T. Hur, and Jung H. Kim. Recurrent Neural Network Verifier for Face Detection and Tracking
5. Karin Sobottka, Ioannis Pitas. Segmentation and Tracking of Faces in Color Images.
6. Fu Jie Huang, Tsuhan Chen. Tracking of Multiple Faces for Human-Computer Interfaces and Virtual Environments.
7. Бустинг // MachineLearning.ru: [сайт]. URL: <http://goo.gl/rvF45r>