



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

ИУ «Информатика и системы управления»

КАФЕДРА

ИУ-1 «Системы автоматического управления»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Детектирование лиц

Студент

ИУ1-51Б
(Группа)

30/12/2020

(Подпись, дата)

Д.И. Юдаков

(И.О. Фамилия)

Руководитель

30/12/2020

(Подпись, дата)

К.В. Парфентьев

(И.О. Фамилия)

2020 г.

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой ИУ-1
(Индекс)

К.А. Неусыпин
(И.О. Фамилия)

« 11 » сентября 20 20 г.

**З А Д А Н И Е
на выполнение научно-исследовательской работы**

по теме Детектирование лиц

Студент группы ИУ1-51Б

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)
исследовательская

Источник тематики (кафедра, предприятие, НИР) НИР

График выполнения НИР: 25% к н., 50% к н., 75% к н., 100% к н.

Техническое задание Исследовать современные подходы и методики к решению задачи детектирования лиц на фотографии.

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на 28 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
оформление графического материала в НИР не предусмотрено

Дата выдачи задания « 11 » сентября 20 20 г.

Руководитель НИР 11/09/2020 К.В. Парфентьев
(Подпись, дата) (И.О. Фамилия)

Студент 11/09/2020 Д.И. Юдаков
(Подпись, дата) (И.О. Фамилия)

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ.....	3
1. НЕЙРОННЫЕ СЕТИ	4
2. СВЁРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ	7
3. МЕТОДЫ ДЕТЕКТИРОВАНИЯ ЛИЦ НА ИЗОБРАЖЕНИИ.....	12
3.1. Эмпирический подход	13
3.2. Метод характерных инвариантных признаков.....	15
3.3. Распознавание с помощью шаблонов	17
3.4. Метод обнаружения по внешним признакам	18
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	23
ПРИЛОЖЕНИЯ.....	24
Приложение 1	24
Код простой свёрточной сети для определения цифры по её изображению	24
Приложение 2	28
Код к анализу методов детектирования лиц	28

ВВЕДЕНИЕ

Сегодня во всех личных устройствах используется идентификация человека. С её помощью устройство «понимает», что оно принадлежит данному человеку, и не даёт другим людям доступ к личной информации. Всё чаще для этого используют биометрические методы: идентификация по отпечатку пальца или с помощью распознавания лица. В связи с простой распознавания лица и обилием камер во всех аспектах жизни человека, последний метод становится всё более актуальным. Распознавание лица можно разбить на три пункта:

- 1) найти лицо на фотографии или видео
- 2) сравнить найденное лицо с лицами, хранящимися в базе данных
- 3) сравнить найденное лицо с эталонным в базе данных.

Но если человек может определить и сравнить лицо с лицами, хранящимися в его памяти, за доли секунды, то без должного обучения машины технологии распознавания, она не сможет отличить человека от любого другого объекта и тем более его идентифицировать. При определении знакомого лица человек ориентируется на индивидуальные его черты, а именно на расстояние между глазами, величину носа, губ, цвет кожи и так далее. Для начала компьютеру необходимо научиться понимать, что перед ним человек, а не кот. При этом положение человека и ракурс камеры играет большую роль.

В работе будет исследовано детектирование лиц людей с использованием гистограммы направленных градиентов (HOG – Histogram of Oriental Gradients) и свёрточных нейронных сетей (CNN – Convolutional Neural Network). В первой части научно-исследовательской работы будет рассмотрен теоретический материал по детектированию лиц. Во второй части речь пойдёт о практической реализации детектирования лиц. Объектом исследования является детектирование лиц в различных положениях, а также точность методов детектирования.

Поставленной целью является создание программы для детектирования лиц на фотографии с помощью библиотеки *dlib* и языка высокого уровня *Python*.

Для достижения цели будет необходимо выполнять ряд задач. Поставленные задачи:

- 1) Сделать обзор теоретического материала по нейронным сетям, описать их виды, предназначение;
- 2) Подробно исследовать технологию свёрточной нейронной сети;
- 3) Изучить теорию по современным методам распознавания лиц, выявить их достоинства и недостатки.

1. НЕЙРОННЫЕ СЕТИ

Нейронная сеть (или искусственная нейронная сеть, ИНС) – это математическая модель, а также её программная или аппаратная реализация, построенная по принципу организации и функционирования *биологических нейронных сетей* – сетей нервных клеток живого организма. Если же говорить проще, то нейронная сеть – это последовательность *нейронов*, соединённых между собой синапсами. На рис. 1 изображена схема простой нейросети, состоящей из трёх слоёв: зелёным обозначены входные нейроны, голубым – скрытые нейроны (скрытый слой) и жёлтым – выходной нейрон.

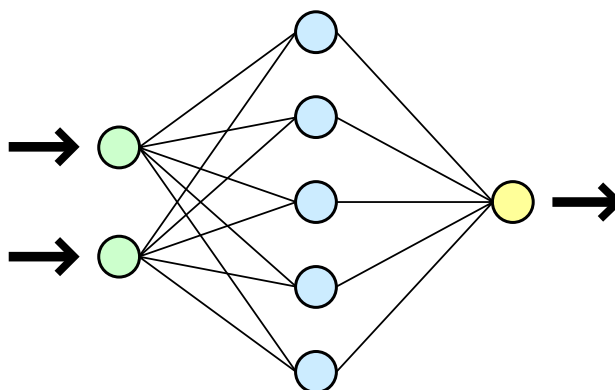


Рис. 1. Схема простой нейронной сети.

Математически, нейрон представляют как некоторую нелинейную функцию от единственного аргумента – линейной комбинации всех входных сигналов. Данную функцию называют *функцией активации* или *передаточной функцией*. На рис. 2 изображена схема нейрона, цифрой 1 обозначены нейроны, выходные сигналы которых поступают на вход данного, цифрой 2 – сумматор входных сигналов, цифрой 3 – вычислитель передаточной функции и цифрой 4 – нейроны, на входы которых подаётся выходной сигнал данного.

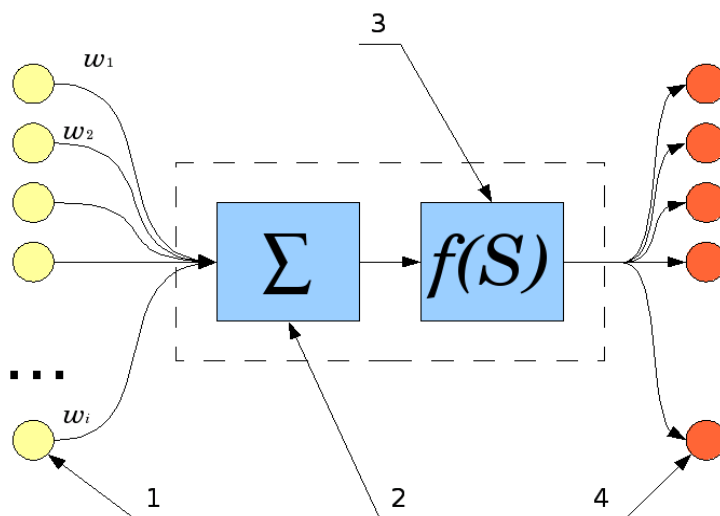


Рис. 2. Схема искусственного нейрона.

Описать нейрон уравнением можно так:

$$y = f(u), \text{ где } u = \sum_{i=1}^n w_i x_i + w_0 x_0,$$

где x_i и w_i — соответственно сигналы на входах нейрона и веса входов,

u — индуцированное локальное поле,

$f(u)$ — передаточная функция.

Возможные значения сигналов на входах нейрона обычно считают заданными в интервале $[0, 1]$. Дополнительный вход x_0 и соответствующий ему вес w_0 используются для инициализации нейрона. Под инициализацией понимается смещение активационной функции нейрона по горизонтальной оси, то есть формирование порога чувствительности нейрона. Иногда к выходу нейрона специально добавляют некую случайную величину, называемую сдвигом.

Передаточная функция нейрона определяет зависимость сигнала на выходе нейрона от взвешенной суммы сигналов на его входах. В большинстве случаев она является монотонно возрастающей, а область значений лежит в $[-1, 1]$ или $[0, 1]$. Нейрон полностью характеризуется своей передаточной функцией. Рассмотрим основные типы передаточных функций.

Линейная передаточная функция. Задаётся формулой:

$$f(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x \geq 1 \\ x & \end{cases}$$

Иногда возможен сдвиг функции по обеим осям. Одним из преимуществ является дифференцируемость на всей числовой оси.

Пороговая передаточная функция. Иногда называют *функцией Хевисайда*. Представляет собой резкий скачок на некоторый уровень T . Задаётся формулой:

$$f(x) = \begin{cases} 1, & x \geq T \\ 0 & \end{cases}$$

Здесь $T = -w_0 x_0$ — сдвиг функции активации относительно горизонтальной оси. Недостаток — не дифференцируема на всей оси абсцисс.

Сигмоидальная передаточная функция. Является самой часто используемой функцией. Существует несколько видов её математической записи. *Логистическая функция:*

$$\sigma(x) = \frac{1}{1 + e^{-tx}}, \text{ где}$$

t — параметр функции, определяющий её крутизну. Важным достоинством этой функции является простота её производной:

$$\frac{d\sigma(x)}{dx} = t\sigma(x)(1 - \sigma(x)).$$

Особенностью нейронов с такой передаточной характеристикой является их усиление сильных сигналов существенно меньше, чем слабых. Это предотвращает насыщение от больших сигналов.

Другой математической записью сигмоидальной функции является *гиперболический тангенс*:

$$th(Ax) = \frac{e^{Ax} - e^{-Ax}}{e^{Ax} + e^{-Ax}}.$$

Она отличается от логистической функции тем, что её область значений лежит в интервале $(-1, 1)$.

Радиально-базисная функция передачи. Она принимает в качестве аргумента расстояние между входным вектором и некоторым наперёд заданным центром активационной функции. Значение этой функции тем выше, чем ближе входной вектор к центру. В качестве такой функции можно взять *функцию Гаусса*:

$$y = e^{-\frac{(S-R)^2}{2\sigma^2}}, \text{ где}$$

$S = \|X - C\|$ — расстояние между центром C и вектором входных сигналов X ,

σ — определяет скорость спада функции при удалении вектора от центра и называется *шириной окна*,

R — определяет сдвиг активационной функции по оси абсцисс.

В качестве расстояния между векторами могут быть использованы различные метрики, обычно используется евклидово расстояние.

Также в качестве передаточных функций могут использоваться: *экспонента, тригонометрический синус, модуль, квадратичная*.

Нейронные сети применяются во многих областях, например, для распознавания образов и классификации, кластеризации, прогнозирования, аппроксимации, сжатия данных и ассоциативной памяти, в задачах анализа данных, оптимизации и в других задачах. Подробнее остановимся на задаче распознавания образов.

2. Свёрточные нейронные сети

Для решения задачи распознавания образов используются следующие виды искусственных нейронных сетей: перцептрон и свёрточные нейронные сети в обучении с учителем, сети адаптивного резонанса в обучении без учителя и сеть РБФ (радиально-базисных функций) в смешанном обучении. Обучение с учителем подразумевает под собой известность выходного пространства решений нейронной сети, обучение без учителя – нейронная сеть формирует выходное пространство решений только на основе выходных воздействий. Рассмотрим свёрточные нейронные сети. В качестве описания будем использовать собственную реализацию простой свёрточной нейронной сети для определения цифры, написанной чёрной ручкой на белом листе, из базы MNIST (Modified National Institute of Standards and Technology) – объемной базы данных цифр, написанных рукой. Примеры изображений изображены на рис. 3.



Рис. 3. Примеры изображений из базы данных MNIST.

Мотивация выбора свёрточной сети связана с тем, что если использовать обычную нейронную сеть, то, взяв картинку 224×224 пикселей и три цветовых канала (RGB), мы получим $224 \times 224 \times 3 = 150528$ входов, что уже довольно много. И если будем использовать 1024 нейрона, то получится $150428 \times 1024 = 150$ миллионов весов только для первого слоя. Такая нейронная сеть будет огромна и практически невозможна в обучении. Вторая причина выбора состоит в том, что цифры могут сдвигаться относительно центра и обычная сеть не сможет определить цифру, поскольку каждый нейрон отвечает за каждый пиксель изображения.

Изображения в базе данных MNIST имеют размер 28×28 пикселей, соответственно будем использовать массив с 784 элементами, а сеть будет иметь 784 входа. Выходной слой будет состоять из 10 выходов, по одному на каждую цифру.

Свёрточные нейронные сети содержат в себе свёрточный слой, который состоит из множества фильтров, являющимися двумерными массивами или матрицами. Операция свёртки включает в себя следующие этапы:

- 1) «Накладывание» фильтра на верх изображения в некоторую ячейку
- 2) Поэлементное перемножение элементов фильтра и значений изображения
- 3) Суммирование результатов перемножения и результат суммирования присваиваем ячейке, в которую фильтр «накладывался»
- 4) Повторение первых трёх этапов для всех ячеек изображения

Будем использовать небольшой свёрточный слой с 8 фильтрами 3×3 , тем самым преобразовывая исходный массив 28×28 в массив $26 \times 26 \times 8$. Уменьшение размера изображения связано с использованием фильтра на краях изображения. В итоге для данного слоя получаем 3×3 (размер фильтра) \times 8 (количество фильтров) = 72 весовых коэффициента.

После слоя свёртки следует субдискретизирующий слой (слой пулинга). Операция пулинга позволяет существенно уменьшить пространственный объём изображения. Пулинг интерпретируется следующим образом: если на предыдущей операции свёртки уже были выявлены некоторые признаки, то для дальнейшей обработки настолько подробное изображение уже не нужно, и оно уплотняется до менее подробного. Для реализации будем последовательно выделять участки 2×2 на изображении, находить из этих 4 элементов максимальный на место этого участка записывать этот элемент, тем самым уменьшая размер вдвое.

В качестве активационной функции будем использовать обобщение логистической функции для многомерного случая – Softmax. Функция преобразует вектор z размерности K в вектор σ той же размерности, где каждая координата σ_i полученного вектора представлена вещественным числом в интервале $[0, 1]$ и сумма координат равна 1. Координаты σ_i вычисляются следующим образом:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}}.$$

Цифра в выходе, которая представлена наибольшей вероятностью, будет выходом нейронной сети.

Схема нейронной сети изображена на рис. 4.

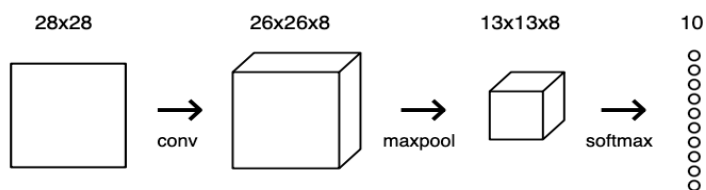


Рис. 4. Схема нейронной сети для распознавания

Кросс-энтропия, или также логарифмическая функция потерь, будет являться функцией потерь в данном случае. Можем посчитать её по формуле:

$$L = -\ln(p_c), \text{ где}$$

p_c — предсказанная вероятность для корректного класса.

Теперь необходимо после каждой прямой итерации, корректировать веса, с помощью метода обратного распространения ошибки. Он заключается в распространении сигналов ошибки от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Для возможности применения такого метода передаточная функция нейронов должна быть дифференцируема. Рассмотрим обратное распространение в каждом слое. Начнём со слоя активационной функции. Нужно посчитать $\frac{\partial L}{\partial out_s}$, где out_s — выход активационного слоя — вектор из 10 вероятностей для каждой цифры. Такая производная будет равна:

$$\frac{\partial L}{\partial out_s} = \begin{cases} 0, i \neq c \\ -\frac{1}{p_i}, i = c \end{cases}, \text{ где}$$

c — корректный класс (цифра),

p_i — вероятность для i -той цифры.

Посчитаем градиент для $out_s(c)$ через значения, поступившие в активационную функцию. Пусть t_i — выходное значение для класса i . Тогда мы можем записать $out_s(c)$ как:

$$out_s(c) = \frac{e^{t_c}}{\sum_i e^{t_i}} = \frac{e^{t_c}}{S} = e^{t_c} S^{-1}, \text{ где } S = \sum_i e^{t_i}.$$

Теперь рассмотрим некоторый класс t , который не равен c . Запишем производную:

$$\frac{\partial out_s(c)}{\partial t_k} = \frac{\partial out_s(c)}{\partial S} \left(\frac{\partial S}{\partial t_k} \right) = -e^{t_c} S^{-2} \left(\frac{\partial S}{\partial t_k} \right) = -e^{t_c} S^{-2} (e^{t_k}) = \frac{-e^{t_c} e^{t_k}}{S^2}.$$

Для $t = c$ получим:

$$\frac{\partial out_s(c)}{\partial t_c} = \frac{S e^{t_c} - e^{t_c} \frac{\partial S}{\partial t_c}}{S^2} = \frac{S e^{t_c} - e^{t_c} e^{t_c}}{S^2} = \frac{e^{t_c} (S - e^{t_c})}{S^2}.$$

Объединим полученные две формулы в одну:

$$\frac{\partial out_s(c)}{\partial t_c} = \begin{cases} \frac{-e^{t_c} e^{t_k}}{S^2}, k \neq c \\ \frac{e^{t_c} (S - e^{t_c})}{S^2}, k = c \end{cases}$$

Так же нам нужен градиент ошибки по весам $\frac{\partial L}{\partial w}$, порогам $\frac{\partial L}{\partial b}$ и входам $\frac{\partial L}{\partial input}$. Для начала мы вычислим градиенты выходных значений по весам, порогам и входам. Запишем уравнение для выхода:

$$t = w \cdot input + b.$$

Тогда градиенты будут равны соответственно:

$$\frac{\partial t}{\partial w} = input, \frac{\partial t}{\partial b} = 1, \frac{\partial t}{\partial input} = w.$$

Теперь вычислим градиенты ошибки:

$$\begin{aligned} \frac{\partial L}{\partial w} &= \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial t} \cdot \frac{\partial t}{\partial w}, \\ \frac{\partial L}{\partial b} &= \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial t} \cdot \frac{\partial t}{\partial b}, \\ \frac{\partial L}{\partial input} &= \frac{\partial L}{\partial out} \cdot \frac{\partial out}{\partial t} \cdot \frac{\partial t}{\partial input}. \end{aligned}$$

Рассмотрим обратное распространение в слое пулинга. В нём на выход слоя влияют лишь числа, которые являются максимальными в их ячейке 2×2 , а остальные не влияют никак. Но максимальные числа не изменяются никак, поэтому градиент ошибки по входам будет равен 1.

И заканчиваться обратное распространение будет на свёрточном слое. Для вычисления градиента ошибки по фильтру, напишем уравнение для выходных пикселей в этом слое:

$$out(i, j) = \sum_{x=0}^3 \sum_{y=0}^3 image(i+x, j+y) \cdot filter(x, y).$$

Тогда градиент выходного пикселя по фильтру:

$$\frac{\partial out(i, j)}{\partial filter(x, y)} = image(i+x, j+y).$$

И градиент ошибки по фильтру:

$$\frac{\partial L}{\partial filter(x, y)} = \sum_i \sum_j \frac{\partial L}{\partial out(i, j)} \cdot \frac{\partial out(i, j)}{\partial filter(x, y)}.$$

Обучение нейронной сети проведём в 3 эпохи по 1000 итераций. После обучения тестируем и получаем неплохие результаты с точностью 0.822 и потерями 0.55.

Таким образом, можно описать основные преимущества и недостатки свёрточной нейронной сети. Преимущества:

- 1) Один из лучших алгоритмов по распознаванию и классификации изображений (далее подкрепим это высказывание практическим исследованием);

- 2) По сравнению с полносвязной нейронной сетью (типа перцептрона) – гораздо меньшее количество настраиваемых весов, поскольку одно ядро весов используется целиком для всего изображения, вместо того чтобы для каждого пикселя входного изображения вводить свои персональные весовые коэффициенты (об этом было сказано в начале раздела);
- 3) Удобное распараллеливание вычислений, а следовательно, возможность реализации алгоритмов работы и обучения сети на графических процессорах (действительно, исходное изображение можно разбивать на несколько независимых частей и работать с каждой из частей отдельно);
- 4) Относительная устойчивость к повороту и сдвигу распознаваемого изображения (поскольку в расчёт берётся не отдельно взятый пиксель изображения, а некоторая их группа);
- 5) Обучение при помощи классического метода обратного распространения ошибки.

Недостатком данной искусственной нейронной сети является большое количество варьируемых параметров сети, так что непонятно, для какой задачи и вычислительной мощности какие нужны настройки. К варьируемым параметрам можно отнести количество слоёв, размерность ядра свёртки для каждого из слоёв, количество ядер для каждого из слоёв, шаг сдвига ядра при обработке слоя, необходимость слоёв субдискретизации, степень уменьшения ими размерности, функция по уменьшению размерности (выбор максимума, среднего и т. п.), передаточная функция нейронов, наличие и параметры выходной полносвязной нейросети на выходе свёрточной – все эти параметры существенно влияют на результат, но обычно выбираются исследователями эмпирически.

3. МЕТОДЫ ДЕТЕКТИРОВАНИЯ ЛИЦ НА ИЗОБРАЖЕНИИ

Задача обнаружения лица на изображении часто является первым шагом в процессе решения задачи более высокого уровня – распознавания лица, деталей лица или его мимики. Кроме того, информация о присутствии и количестве лиц на изображении может быть полезна в системах автоматического учета числа посетителей; системах пропускного контроля в учреждениях, аэропортах и метро; автоматических системах предотвращения несчастных случаев; интеллектуальных интерфейсах «человек-компьютер»; в фототехнике для автоматической фокусировки на лице человека, а также для стабилизации изображения лица с целью облегчения распознавания эмоций; для расширения зоны стереовидения при создании систем 3D отображения. Существующие алгоритмы обнаружения лиц можно разбить на четыре категории:

- 1) Эмпирический метод;
- 2) Метод характерных инвариантных признаков;
- 3) Распознавание с помощью шаблонов, заданных разработчиком;
- 4) Метод обнаружения по внешним признакам, обучающиеся системы.

Подробно рассмотрим каждую из категорий.

3.1. Эмпирический подход

Эмпирический подход «базирующийся на знаниях сверху-вниз» (knowledge based top-down methods) предполагает создание алгоритма, реализующего набор правил, которым должен отвечать фрагмент изображения, для того чтобы быть признанным человеческим лицом. Этот набор правил является попыткой формализовать эмпирические знания о том, как именно выглядит лицо на изображениях и чем руководствуется человек при принятии решения: лицо он видит или нет. Самые простые правила:

- 1) центральная часть лица имеет однородную яркость и цвет;
- 2) разница в яркости между центральной частью и верхней частью значительна;
- 3) лицо содержит в себе два симметрично расположенных глаза, нос и рот, резко отличающиеся по яркости относительно остальной части лица.

Метод сильного уменьшения изображения для сглаживания помех, а также для уменьшения вычислительных операций предварительно подвергает изображение сильному уменьшению (рис. 5).

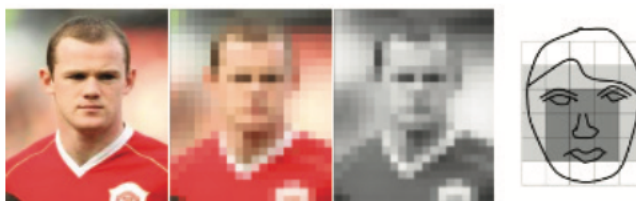


Рис. 5. Метод сильного уменьшения изображения.

На таком изображении проще выявить зону равномерного распределения яркости (предполагаемая зона нахождения лица), а затем проверить наличие резко отличающихся по яркости областей внутри: именно такие области можно с разной долей вероятности отнести к «лицу». Метод построения гистограмм для определения областей изображения с «лицом»

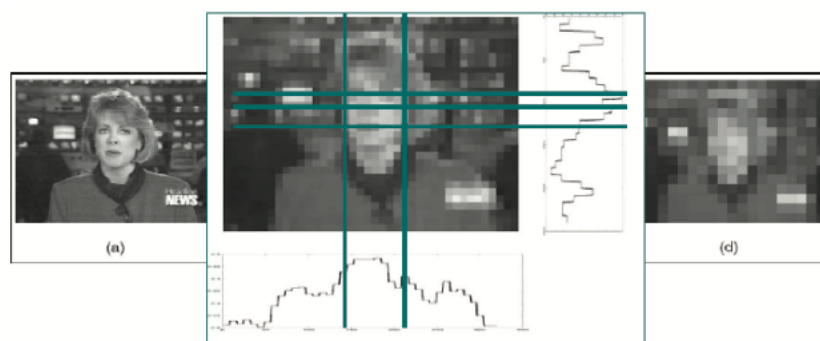


Рис. 6. Метод построения гистограмм.

строит вертикальную и горизонтальную гистограммы (рис. 6). В областях-кандидатах происходит поиск черт лица.

Данный подход использовался на заре развития компьютерного зрения ввиду малых требований к вычислительной мощности процессора для обработки изображения. Рассмотренные выше методы имеют неплохие показатели по выявлению лица на изображении при однородном фоне, они легко реализуемы с помощью машинного кода. Впоследствии было разработано множество подобных алгоритмов. Но эти методы абсолютно непригодны для обработки изображений, содержащих большое количество лиц или сложный задний фон. Также они очень чувствительны к наклону и повороту головы.

3.2. Метод характерных инвариантных признаков

Методы характерных инвариантных признаков, базирующиеся на знаниях снизу-вверх (Feature invariant approaches) образуют второе семейство способов детектирования лиц. Здесь виден подход к проблеме с другой стороны: нет попытки в явном виде формализовать процессы, происходящие в человеческом мозге. Сторонники подхода стараются выявить закономерности и свойства изображения неявно, найти инвариантные особенности лица, независимо от угла наклона и положения. Основные этапы алгоритмов этой группы методов:

- 1) детектирование на изображении явных признаков лица: глаз, носа, рта;
- 2) обнаружение: границы лица, форма, яркость, текстура, цвет;
- 3) объединение всех найденных инвариантных признаков и их верификация;

Метод обнаружения лиц в сложных сценах предполагает поиск правильных геометрических расположений черт лица. Для этого применяется гауссовский производный фильтр с множеством различных масштабов и ориентаций. После этого производится поиск соответствия выявленных черт и их взаимного расположения случайным перебором.

Суть метода группировки признаков в применении второй производной гауссовского фильтра для поиска интересующих областей изображения. Далее группируются края вокруг каждой такой области при помощи порогового фильтра. А затем используется оценка при помощи байесовской сети для комбинирования найденных признаков, таким образом происходит выборка черт лица.

Методы этой группы в качестве достоинств имеют возможность распознавать лицо в различных положениях. Но даже при небольшом загромождении лица другими объектами, возникновении шумов или засветке процент достоверного распознавания сильно падает. Большое влияние также оказывает сложный задний фон изображения. Основа рассмотренных подходов — эмпирика, является одновременно их сильной и слабой стороной. Большая изменчивость объекта распознавания, зависимость вида лица на изображении от условий съемки и освещения позволяют без колебаний отнести обнаружение лица на изображении к задачам высокой сложности. Применение эмпирических правил позволяет построить некоторую модель изображения лица и свести задачу к выполнению некоторого количества относительно простых проверок. Однако, несмотря на безусловно разумную посылку — попытаться использовать и повторить уже успешно функционирующий инструмент распознавания — человеческое зрение, методы первой категории пока далеки по эффективности от своего прообраза, поскольку исследователи, решившие избрать этот путь, сталкиваются с рядом серьезных трудностей. Во-первых, процессы, происходящие в мозгу во время решения задачи распознавания изображений изучены далеко не полностью, и тот набор эмпирических знаний о человеческом

лице, которые доступны исследователям на сознательном уровне, далеко не исчерпывает инструментарий, используемый мозгом подсознательно. Во-вторых, трудно эффективно перевести неформальный человеческий опыт и знания в набор формальных правил, поскольку чересчур жёсткие рамки правил приведут к тому, что в ряде случаев лица не будут обнаружены, и, напротив, слишком общие правила приведут к большому количеству случаев ложного обнаружения.

3.3. Распознавание с помощью шаблонов

Шаблоны задают некий стандартный образ изображения лица, например, путём описания свойств отдельных областей лица и их возможности взаимного расположения. Обнаружение лица с помощью шаблона заключается в проверке каждой из областей изображения на соответствие заданному шаблону. Особенности подхода:

- 1) два вида шаблонов: недеформируемые и деформируемые;
- 2) шаблоны заранее запрограммированы, необучаемы;
- 3) используется корреляция для нахождения лица на изображении;

Метод детектирования лица при помощи трёхмерных форм предполагает использование шаблона в виде пар отношений яркостей в двух областях. Для определения местоположения лица необходимо пройти всё изображение на сравнение с заданным шаблоном. Причём делать это необходимо с различным масштабом (рис. 7).

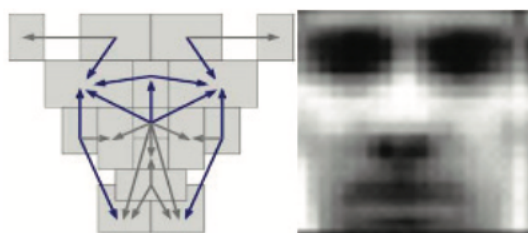


Рис. 7. Метод трёхмерных форм.

Модели распределения опорных точек являются статистическими моделями, которые представляют объекты, форма которых может измениться. Их полезная особенность — способность выделить форму переменных объектов в пределах учебного набора с небольшим количеством параметров формы. Эта компактная и точная параметризация может использоваться для разработки эффективных систем классификации.

К достоинствам распознавания с помощью шаблонов можно отнести относительную простоту реализации и неплохие результаты на изображениях с не очень сложным задним фоном. А главным недостатком является необходимость калибровки шаблона вблизи с изображением лица. Большая трудоёмкость вычисления шаблонов для различных ракурсов и поворотов лица ставят под вопрос целесообразности их использования.

3.4. Метод обнаружения по внешним признакам

Методы обнаружения лица по внешним признакам – это методы, при которых необходимо провести этап обучения системы, путём обработки тестовых изображений. Изображению (или его фрагменту) ставится в соответствие некоторым образом вычисленный вектор признаков, который используется для классификации изображений на два класса — лицо/не лицо. Обычно поиск лиц на изображениях с помощью методов, основанных на построении математической модели изображения лица, заключается в полном переборе всех прямоугольных фрагментов изображения всевозможных размеров и проведения проверки каждого из фрагментов на наличие лица. Поскольку схема полного перебора обладает такими безусловными недостатками, как избыточность и большая вычислительная сложность, авторами применяются различные методы сокращения количества рассматриваемых фрагментов. Основные принципы методов:

- 1) Схоластика: каждый объект сканируется окном и представляет векторами ценности;
- 2) Блочная структура: изображения разбивается на пересекающиеся или непересекающиеся участки различных масштабов и производится оценка с помощью алгоритмов оценки весов векторов.

Для обучения алгоритмов требуется библиотека вручную подготовленных изображений лиц и не лиц, любых других изображений.

Стоит отметить, что важнейшей задачей является выделение сильных классификаторов. Именно они будут иметь наивысший приоритет для проверки найденных признаков в изображении. Количество же более слабых классификаторов стоит уменьшать за счёт похожести друг на друга, а также удалении классификаторов, возникших за счёт шумовых выбросов. Перечислим основные методики выполнения этих задач:

- 1) Искусственные нейронные сети;
- 2) Метод главных компонент;
- 3) Факторный анализ;
- 4) Линейный дискриминантный анализ;
- 5) Метод опорных векторов;
- 6) Наивный байесовский классификатор;
- 7) Скрытые Марковские модели;
- 8) Метод распределения;
- 9) Совмещение факторного анализа и метода главных компонент;
- 10) Разреженная сеть окон;

11) Активные модели;

12) Адаптированное улучшение и основанный на нём метод Виолы-Джонса.

Самым перспективным на сегодняшний день в плане высокой производительности и низкой частоты ложных срабатываний и большим процентом верных обнаружений является метод Виолы-Джонса.

Рассмотрим современное обнаружение лица на изображении. На первом этапе происходит обнаружение черт лица с помощью метода Виолы-Джонса, основные принципы которого:

- 1) интегральное представление изображения для быстрого вычисления необходимых объектов;
- 2) признаки Хаара для поиска нужного объекта;
- 3) бустинг для выбора наиболее подходящих признаков искомого объекта;
- 4) классификаторы для отсеивания выбранных признаков;
- 5) каскады признаков для быстрого исключения окон, в которых искомым объектом не найден.

У данного классификатора высокие точность и скорость обнаружения лиц на изображениях, но имеются и недостатки: долгое время обучения и требование большого объема данных для этого.

Детектировать лицо на изображении можно и с помощью нейронных сетей. Они основываются на идее, что за счет обобщающих способностей обученная нейронная сеть применяет полученный в процессе обучения опыт на неизвестные объекты. В процессе обучения путем решения задачи оптимизации методом градиентного спуска производится настройка весов связей между нейронами, таким образом, нейронная сеть автоматически извлекает ключевые признаки, определяет их важность и строит взаимосвязь между ними. Однако стоит учитывать, что нейронная сеть имеет недостатки. Во-первых, при внесении нового экземпляра в базу данных необходимо полностью её переобучать, что является долгим процессом. Во-вторых, можно столкнуться с проблемой переобучения, т.е. явлением, при котором алгоритм обучения вырабатывает предсказания, слишком близко соответствующие конкретному набору данных, что делает алгоритм менее точным на дополнительных данных. И наконец, довольно затруднительно формализуется сам этап выбора архитектуры сети, на котором определяется количество нейронов и характер связей между ними.

Еще одним методом детектирования лиц является алгоритм, основанный на классификаторе, обученном на гистограмме направленных градиентов (HOG) и методе опорных векторов. Идея применения HOG-дескриптора заключается в том, что внешний вид и форма объекта

в рассматриваемой области изображения можно описать распределением градиентов интенсивностей пикселей, соответствующих не каждой точке конкретно, а всей этой области. Поскольку работа HOG происходит локально, данный метод инвариантен к смещению, повороту, изменению масштаба и яркости изображения, но все равно чувствителен к перемене ориентации объекта. Таким образом, на первом этапе на изображении локализуются области, относящиеся к лицам, и можно перейти к этапу поиска ключевых точек.

Найти и описать ключевые точки лица на изображении можно различными способами, например, при помощи гистограммы направленных градиентов, т.к. данный дескриптор широко распространен и имеет много реализаций. Упрощенное представление изображения или его части, в котором отброшена посторонняя информация, называют дескриптор-функцией, она, как правило, преобразует изображение в вектор. В качестве функции HOG-дескриптор использует распределение направлений градиентов, суть которого состоит в том, что наибольшая величина градиентов соответствует краям и углам объекта, т.е. областям с резким изменением интенсивностей.

В качестве анализа методов детектирования лиц на изображении предлагается сравнить точность метода, основанного на гистограмме направленных градиентов, и свёрточной нейронной сети. Код анализа методов находится в приложении 2. Возьмём изображение с 16 относительно чётко видимыми лицами (рис. 8)



Рис. 8. Фотография с 16 лицами.

Выполнив поиск лиц на изображении с помощью гистограммы направленных градиентов и с помощью свёрточной нейронной сети, получили следующий результат (рис. 9, 10).

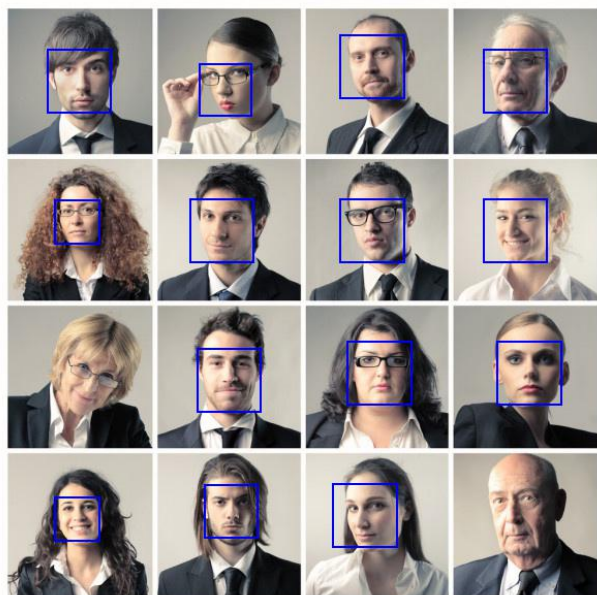


Рис. 9. Изображение с выделенными лицами HOG-методом.

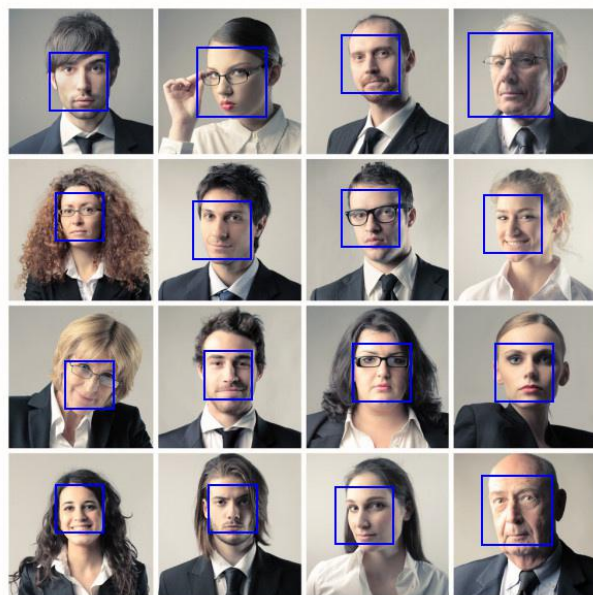


Рис. 10. Изображение с выделенными лицами с помощью свёрточной нейронной сети.

Метод, основанный на гистограмме направленных градиентов, сработал быстрее, но менее точно, чем свёрточная нейронная сеть.

Итак, мы рассмотрели основные методы детектирования лиц, описали их достоинства и недостатки, а также сравнили HOG-метод и свёрточную нейронную сеть непосредственно обнаружив лица на изображении.

ЗАКЛЮЧЕНИЕ

В работе произведён обзор нейронной сети, основных её видов, применении видов в разных отраслях науки и техники. Подробно изучен и описан один из видов нейронных сетей, а именно свёрточной нейронной сети. Были детально показаны этапы работы свёрточной нейронной сети и её слоёв по отдельности, а в последствии сделаны выводы об основных достоинствах и недостатках данного вида нейронной сети.

Первым этапом в распознавании и идентификации личности по его лицу, является обнаружение его лица. Эта задача не менее важная, чем само распознавание, ведь от правильности и чистоты детектирования зависит точность сопоставления слепка лица его отснятой копии. Для обнаружения лица на изображении в настоящее время используется множество методов, но можно выделить несколько значимых на данный момент: метод, работающий с помощью гистограммы направленных градиентов, метод, основанный на свёрточной нейронной сети, и метод Виолы-Джонса. Были описаны основные преимущества и недостатки каждого из метода, а также произведён практический анализ первых двух методов.

В дальнейшем есть возможность продолжения работы над следующим этапом – непосредственно распознаванием лица, его идентификацией, нахождением различий между двумя и более лицами.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Татаренков Д.А. Анализ методов обнаружения лиц на изображении // Молодой учёный. 2015. №4. с. 270-276.
2. G. Yang and Thomas S. Huang. «Human face detection in a complex background. Pattern Recognition», 27(1):53-63, 1994.
3. P. Viola and M.J. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001, vol. 1, pp. 511-518.
4. P. Viola and M.J. Jones, «Robust real-time face detection», International Journal of Computer Vision, vol. 57, no. 2, 2004., pp. 137-154.
5. Anil K. Jain, Jianchang Mao, K.M. Mohiuddin Artificial Neural Networks: A Tutorial, Computer, vol. 29, no. 3, March/1996, pp. 31-44.
6. C.M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.
7. Ben Graham. Fractional Max-Pooling.
8. S. Belongie, J. Malik, and J. Puzicha. Matching shapes. The 8th ICCV, Vancouver, Canada, pp. 454-461, 2001.

ПРИЛОЖЕНИЯ

Приложение 1

Код простой свёрточной сети для определения цифры по её изображению

Код главной функции, где происходит обучение и тестирование нейронной сети:

```
import mnist
import numpy as np
from conv import Conv3x3
from maxpool import MaxPool2
from softmax import Softmax

test_images = mnist.test_images()[1:1000]
test_labels = mnist.test_labels()[1:1000]
train_images = mnist.train_images()[1:1000]
train_labels = mnist.train_labels()[1:1000]

conv = Conv3x3(8) # 28x28x1 -> 26x26x8
pool = MaxPool2() # 26x26x8 -> 13x13x8
softmax = Softmax(13 * 13 * 8, 10) # 13x13x8 -> 10

def forward(image, label):
    # transform image from [0, 255] to [-0.5, 0.5]
    out = conv.forward((image / 255) - 0.5)
    out = pool.forward(out)
    out = softmax.forward(out)
    loss = -np.log(out[label])
    acc = 1 if np.argmax(out) == label else 0
    return out, loss, acc

def train(im, label, lr=.005):
    out, loss, acc = forward(im, label)
    gradient = np.zeros(10)
    gradient[label] = -1 / out[label]
    gradient = softmax.backprop(gradient, lr)
    gradient = pool.backprop(gradient)
    conv.backprop(gradient, lr)
    return loss, acc

print('MNIST CNN initialized!')
for epoch in range(3):
    print('--- Epoch %d ---' % (epoch + 1))

    permutation = np.random.permutation(len(train_images))
    train_images = train_images[permutation]
    train_labels = train_labels[permutation]

    loss = 0
    num_correct = 0
    for i, (im, label) in enumerate(zip(train_images, train_labels)):
        if i > 0 and i % 100 == 99:
            print('[Step %d] Past 100 steps: Average Loss %.3f | Accuracy: %d%%'
                  %
```

```

        (i + 1, loss / 100, num_correct)
    )
    loss = 0
    num_correct = 0
    l, acc = train(im, label)
    loss += l
    num_correct += acc

print('\n---Testing the CNN ---')
loss = 0
num_correct = 0
for im, label in zip(test_images, test_labels):
    _, l, acc = forward(im, label)
    loss += l
    num_correct += acc

num_tests = len(test_images)
print('Test Loss:', loss / num_tests)
print('Test Accuracy:', num_correct / num_tests)

```

Код свёрточного слоя:

```

import numpy as np

class Conv3x3:
    def __init__(self, num_filters):
        self.num_filters = num_filters
        self.filters = np.random.randn(num_filters, 3, 3) / 9

    def iterate_regions(self, image):
        h, w = image.shape
        for i in range(h - 2):
            for j in range(w - 2):
                im_region = image[i:(i + 3), j:(j + 3)]
                yield im_region, i, j

    def forward(self, input):
        self.last_input = input
        h, w = input.shape
        output = np.zeros((h - 2, w - 2, self.num_filters))
        for im_region, i, j in self.iterate_regions(input):
            output[i, j] = np.sum(im_region * self.filters, axis=(1, 2))
        return output

    def backprop(self, dL_dout, learn_rate):
        dL_dfilters = np.zeros(self.filters.shape)
        for im_region, i, j in self.iterate_regions(self.last_input):
            for f in range(self.num_filters):
                dL_dfilters[f] += dL_dout[i, j, f] * im_region
        self.filters -= learn_rate * dL_dfilters
        return None

```

Код слоя пулинга:

```

import numpy as np

class MaxPool2:
    def iterate_regions(self, image):
        h, w, _ = image.shape
        new_h = h // 2
        new_w = w // 2

```

```

for i in range(new_h):
    for j in range(new_w):
        im_region = image[(i * 2):(i * 2 + 2), (j * 2):(j * 2 + 2)]
        yield im_region, i, j

def forward(self, input):
    self.last_input = input
    h, w, num_filters = input.shape
    output = np.zeros((h // 2, w // 2, num_filters))
    for im_region, i, j in self.iterate_regions(input):
        output[i, j] = np.amax(im_region, axis=(0, 1))
    return output

def backprop(self, dL_dout):
    dL_dinput = np.zeros(self.last_input.shape)
    for im_region, i, j in self.iterate_regions(self.last_input):
        h, w, f = im_region.shape
        amax = np.amax(im_region, axis=(0, 1))
        for i2 in range(h):
            for j2 in range(w):
                for f2 in range(f):
                    if im_region[i2, j2, f2] == amax[f2]:
                        dL_dinput[i * 2 + i2, j * 2 + j2, f2] = dL_dout[i,
j, f2]
    return dL_dinput

```

Код активационной функции:

```

import numpy as np

class Softmax:
    def __init__(self, input_len, nodes):
        self.weights = np.random.randn(input_len, nodes) / input_len
        self.biases = np.zeros(nodes)

    def forward(self, input):
        self.last_input_shape = input.shape
        input = input.flatten()
        self.last_input = input
        input_len, nodes = self.weights.shape
        totals = np.dot(input, self.weights) + self.biases
        self.last_totals = totals
        exp = np.exp(totals)
        return exp / np.sum(exp, axis=0)

    def backprop(self, dL_dout, learn_rate):
        for i, gradient in enumerate(dL_dout):
            if gradient == 0:
                continue
            t_exp = np.exp(self.last_totals)
            S = np.sum(t_exp)
            dout_dt = -t_exp[i] * t_exp / (S ** 2)
            dout_dt[i] = t_exp[i] * (S - t_exp[i]) / (S ** 2)

            dt_dw = self.last_input
            dt_db = 1
            dt_dinputs = self.weights
            dL_dt = gradient * dout_dt
            dL_dw = dt_dw[np.newaxis].T @ dL_dt[np.newaxis]
            dL_db = dL_dt * dt_db

```

```
dL_dinputs = dt_dinputs @ dL_dt  
self.weights -= learn_rate * dL_dw  
self.biases -= learn_rate * dL_db  
return dL_dinputs.reshape(self.last_input_shape)
```

Приложение 2

Код к анализу методов детектирования лиц

```
import face_recognition as det
from PIL import Image, ImageDraw

image_name = "manyfaces.jpg"

image_hog = det.load_image_file(image_name)
face_locations_hog = det.face_locations(image_hog)
img = Image.open(image_name)
for face_location in face_locations_hog:
    top, right, bottom, left = face_location
    ImageDraw.Draw(img).rectangle((left, top, right, bottom), outline='blue',
width=2)
img.show()

image_cnn = det.load_image_file(image_name)
face_locations_cnn = det.face_locations(image_cnn, model='cnn')
img = Image.open(image_name)
for face_location in face_locations_cnn:
    top, right, bottom, left = face_location
    ImageDraw.Draw(img).rectangle((left, top, right, bottom), outline='blue',
width=2)
img.show()
```