

Донейросетевые подходы к работе с изображениями

Никита Андреев

План доклада

- Цифровое изображение
- Задачи распознавания изображений
- Основные алгоритмы распознавания изображений
- Методы извлечения признаков

Цифровое изображение

Матрица, в каждой ячейке - цвет пикселя, заданный с помощью конечного набора значений.

Например:

- RGB. Диапазон значений для каждого цвета [0, 255]
- Градации серого.

$$Y' = 0.2126R + 0.7152G + 0.0722B$$

Распознавание изображений

- Нужно везде
- Хотим понять, что изображено на картинке
- Можно распознавать объекты, например:
 - Дорожные знаки
 - Люди
 - Лица

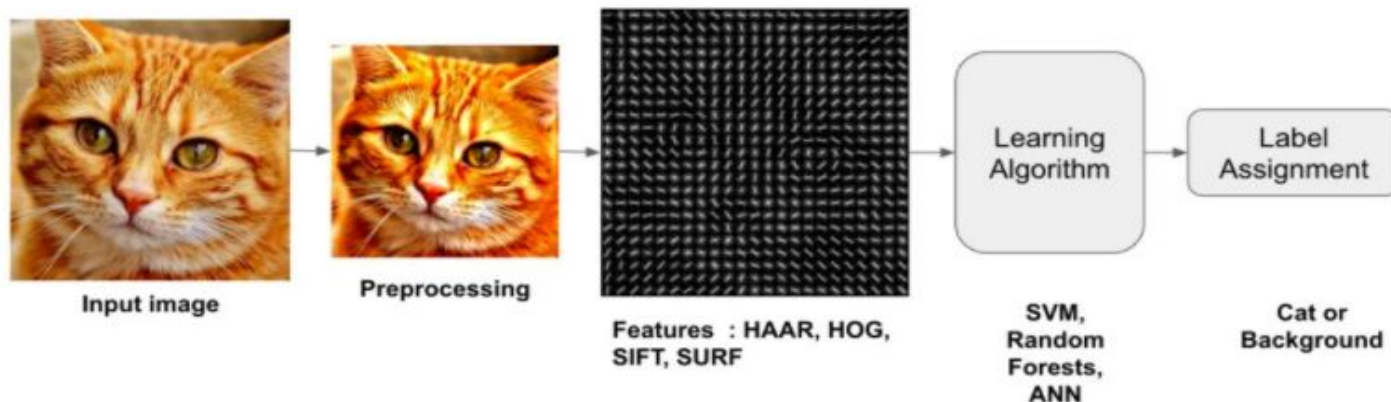
Первый алгоритм распознавания лиц изобретен в 2001 году Полом Виолой и Майклом Джонсом.

Входные и выходные данные

- На вход подается изображение
- На выходе мы имеем в качестве ответа принадлежность к одному из классов
- Задача - построить модель, которая будет относить изображение к нужному классу

Алгоритм распознавания изображений

1. Загрузка изображения
2. Предобработка
3. **Извлечение признаков**
4. Обучение модели
5. Классификация изображений



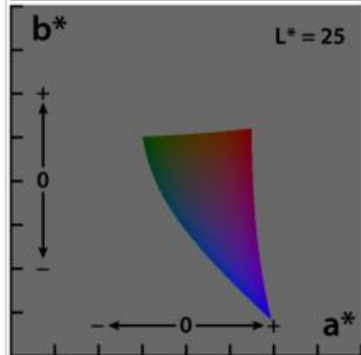
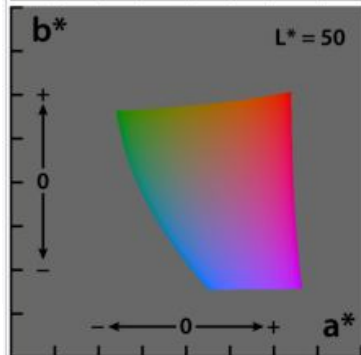
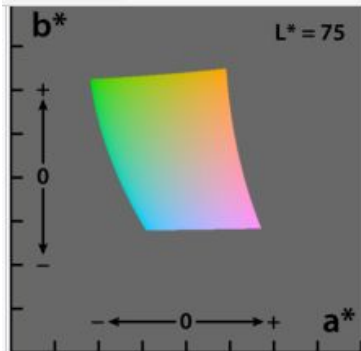
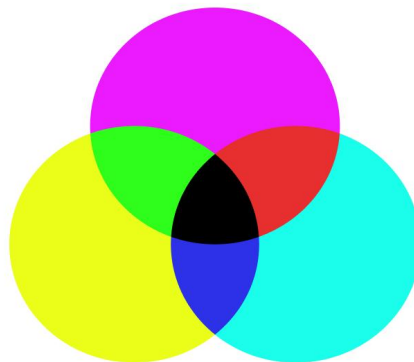
Предобработка

- Нужна для нормализации яркости и контраста
 - вычитание среднего значения интенсивности изображения и деление на стандартное отклонение.
 - Перевод из RGB в HSI (тон, насыщенность, интенсивность)

$$\left\{ \begin{array}{l} H = \begin{cases} \theta; B \leq G \\ 360 - \theta; B > G \end{cases} \text{ где } \theta = \arccos \left(\frac{\frac{1}{2} * ((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \\ S = 1 - \frac{3}{(R + G + B)} \min(R, G, B) \\ I = \frac{1}{3}(R + G + B) \end{array} \right.$$

Предобработка

- Преобразование цветового пространства
 - Перевод в CIE $L^*a^*b^*$ (**CIELAB**) (L^* - степень освещения от черного (0) до белого (100), a^* от зеленого (-) до красного (+), b^* от синего (-) до желтого (+))
 - CMYK (Cyan - голубой, Magenta - пурпурный, Yellow - желтый, Key color - черный)



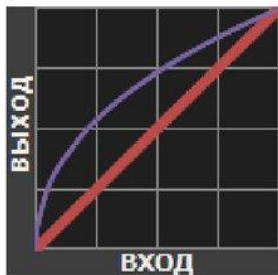
Предобработка

Гамма-коррекция

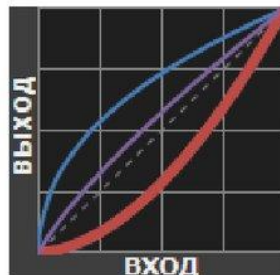
$$V_{new} = A * V_{old}^{\gamma}$$

Используется при
конвертации из
RAW формата в
стандартный(JPEG,
TIFF, PSD).

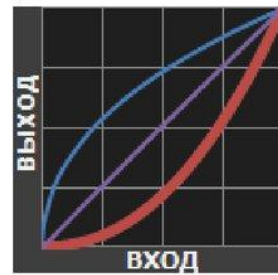
1.0



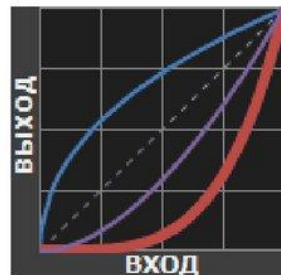
1.8



2.2



4.0



Извлечение признаков

- упрощение изображения путем извлечения важной информации, содержащейся в изображении, и исключения остальной информации.

Пример:

1. Хотим найти пуговицы рубашки и пальто на изображениях
2. Запустив детектор края на изображении, мы можем упростить изображение.
3. Все еще можно легко различить круглую форму кнопок на этих краевых изображениях(отбросили несущественную информацию)

Оказывается, мы можем сделать гораздо лучше, чем простое обнаружение краев и найти функции, которые гораздо более надежны. (например, фиксируют не только форму, но и смогут “отличить” от автомобильных шин)

Histogram of Oriented Gradients

1. Изменение размеров (64 x 128) + (изменение цветового пространства)
2. Вычисление значений градиентов

-1	0	1
----	---	---

-1
0
1

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}$$

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

3. Разбиение изображения на клетки 8 x 8

Histogram of Oriented Gradients

Вычисление гистограммы для клетки 8 x 8.

- Имеем 64 пары (величина градиента, направление(угол от 0 до 180))
- Строим гистограмму с 9 ячейками(bins), каждая ячейка отвечает за свое направление 0, 20, 40 ... 160 градусов
- Если значение это 0, 20 ... 160, то оно учитывается в соответствующей ячейке, иначе учитывается половина значения в двух соседних(например для 2 с углом 30: по 1 попадёт в ячейки 20 и 40)

Histogram of Oriented Gradients

Блоки и их нормализация

Нормализация нужна для лучшей устойчивости к изменениям яркости и контрастности

- Объединяем клетки в блоки 16 x 16 (то есть в блоке 4 клетки)
- Получаем для каждого блока 36 элементный вектор
- Нормируем его

$$f = \frac{v}{(\|v\|_1 + e)}$$

Histogram of Oriented Gradients

Получаем вектор признаков:

- Перемещаем блок размером 16x16 с шагом 8.
- После каждого перемещения добавляем в ответ вектор, описывающий текущий блок 16x16
- Получаем вектор признаков

Если начальное изображение 64x128, то в итоге мы имеем вектор размера $7 * 15 * 36 = 3780$.

SIFT(Scale-invariant feature transform)

Алгоритм SIFT в 1999 году. Он позволяет сравнивать изображения, подвергнутые таким трансформациям как изменение масштаба, смещение объекта на сцене, повороты камеры или объекта. Алгоритм работает с чёрно-белыми изображениями.

SIFT

1. Выбираются числа σ_0 , k , s , такие что $k = 2^{\frac{1}{s}}$
2. Из исходного изображения I_0 формируются s изображений I_h , такие что размер I_h в два раза меньше размера I_{h-1}
3. К исходному изображению применяется функция Гауссова размытия (фильтр Гаусса) с разными радиусами σ_i

$$L_{i,h}(x, y, \sigma_i) = G(x, y, \sigma_i) * I_h(x, y)$$

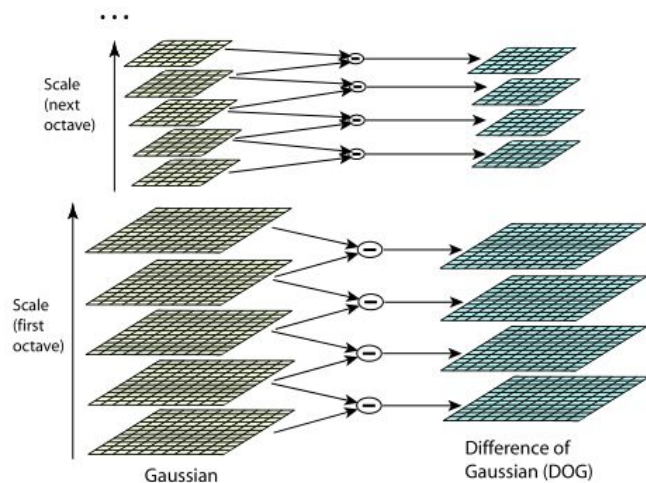
$$G(x, y, \sigma_i) = \frac{1}{2\pi\sigma_i^2} e^{-\frac{(x^2+y^2)}{2\sigma_i^2}}$$

SIFT

4. Рассчитываются пиксельные разности изображений, размытых Гауссовым фильтром $D_{i,h}(x, y, \sigma_i)$

$$D_{i,h}(x, y, \sigma_i) = L_{i,h}(x, y, \sigma_i) - L_{i,h}(x, y, \sigma_{i-1}),$$

5. Формируется h пирамид разностей гауссианов $DoG_h : \forall i DoG_h \in D_{i,h}$

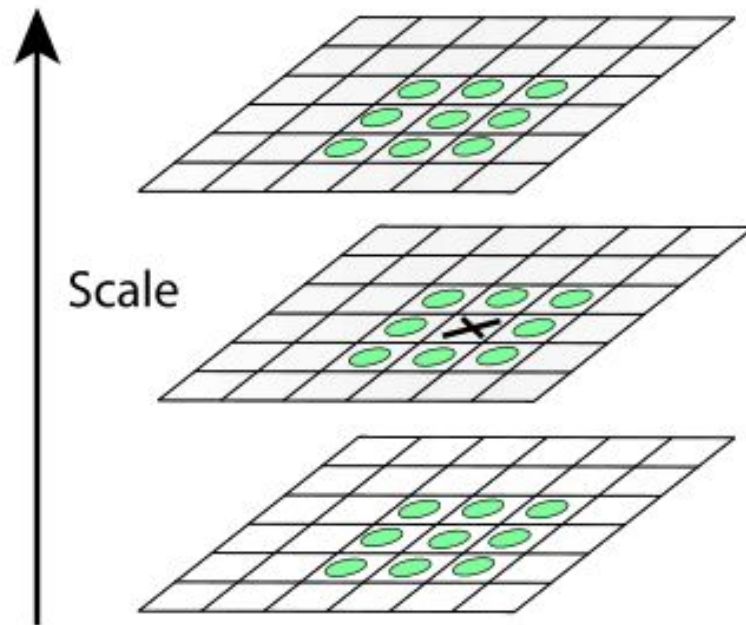


Scale-invariant feature transform

Особая точка - локальный экстремум разности гауссианов

Для каждой точки нужно посмотреть на 26 её соседей.

Особые точки ищутся в каждом изображении пирамиды DoG.



SIFT

Далее необходимо уточнить положение экстремума. Это делается с помощью многочлена Тейлора второго порядка, взятого в точке вычисленного экстремума

$$D(Z) = D + \frac{\delta D^T}{\delta Z} Z + \frac{1}{2} Z^T \frac{\delta^2 D}{\delta Z^2} Z$$

D — функция DoG_h , $Z = (x, y, \sigma)$ — вектор смещения относительно точки разложения

Продифференцировав и приравняв к нулю, в итоге получим смещение точки вычисленного экстремума, относительно точного

$$\hat{Z} = - \frac{\delta D^{-1}}{\delta Z^2} \frac{\delta D}{\delta Z}$$

Если смещение больше 0.5, то это значит, что экстремум лежит ближе к другому кандидату.

SIFT

- Затем требуется отбросить точки с низким контрастом. После нахождения точки экстремума проверяется само значение DoG в этой точке. Если оно меньше 0.03, то такая точка отбрасывается
- Далее необходимо отфильтровать точки, лежащие на границе объектов

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\frac{\text{Tr}(H)^2}{\Delta} = \frac{(a + \beta)^2}{a\beta} = \frac{(r + 1)^2}{r}$$

Пусть $\text{Tr}(H)$ – след матрицы, Δ – определитель

Точка продолжает рассматриваться дальше, если

$$\text{Tr}(H) = D_{xx} + D_{yy} = a + \beta$$

$$\Delta = D_{xx}D_{yy} - (D_{xy})^2 = a\beta$$

$$\frac{\text{Tr}(H)^2}{\Delta} < \frac{(r + 1)^2}{r}$$

SIFT

Для инвариантности относительно поворота вычисляется градиент и направление

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

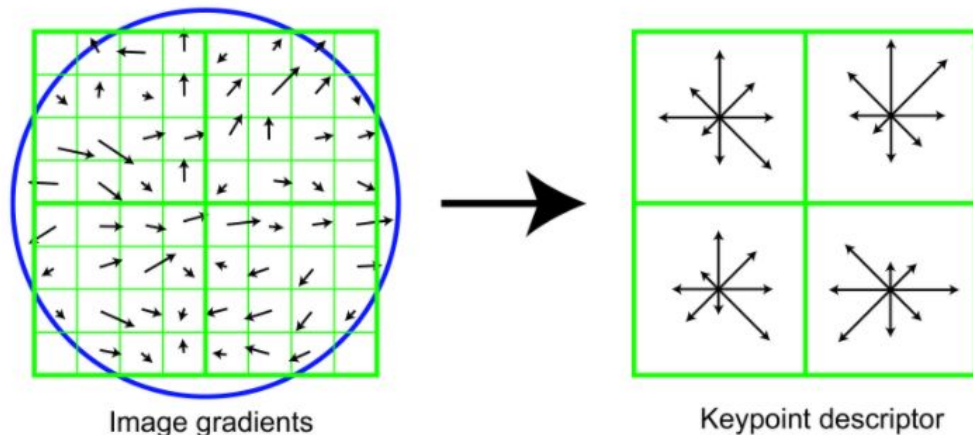
$$\theta(x, y) = \arctan\left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}\right)$$

Строится гистограмма с 36 ячейками. Точка вносит вклад равный произведению градиента и ядра Гаусса.

Направление ключевой точки лежит в промежутке, который покрывается максимальной компонентой гистограммы. Значения максимальной компоненты (max) и двух соседних с ней интерполируются параболой, и точка максимума этой параболы берётся в качестве направления ключевой точки

SIFT

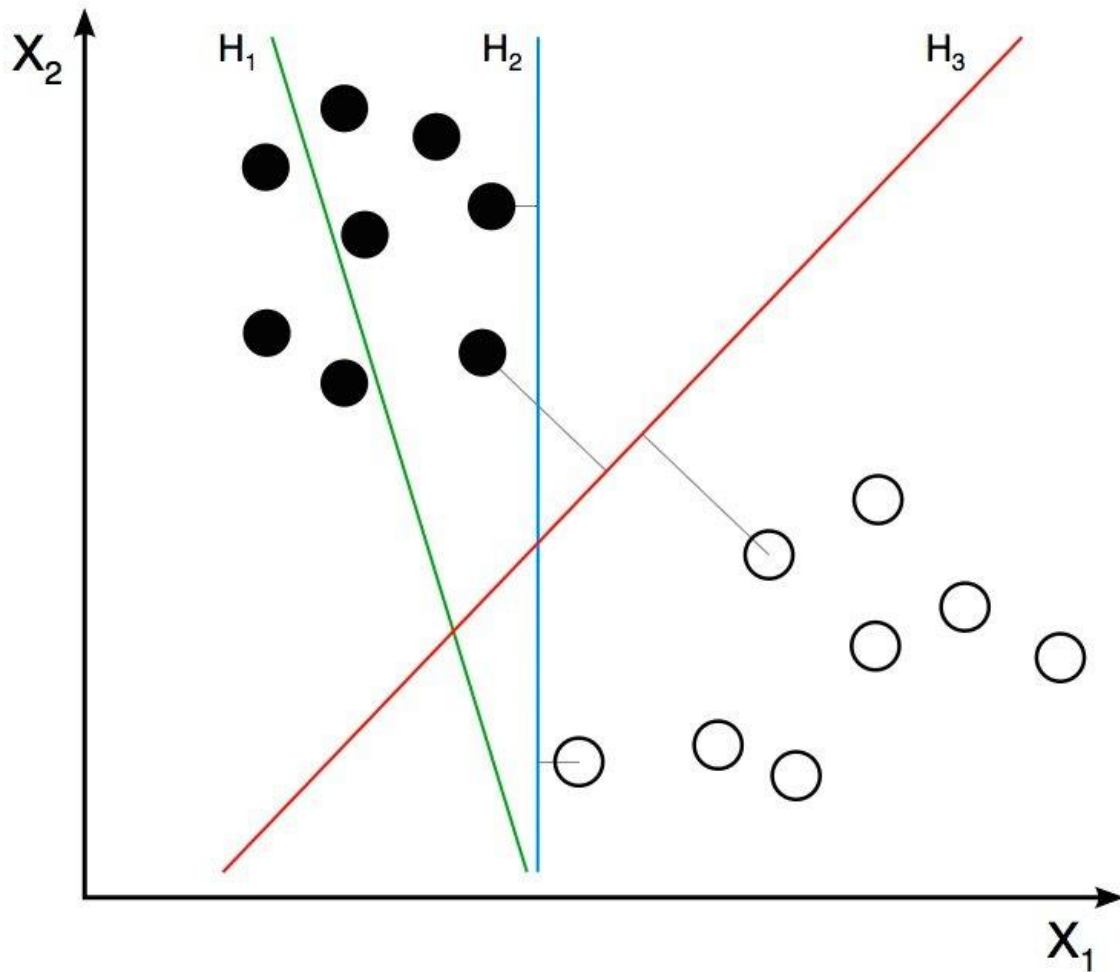
Перед вычислением дескриптора выбирается, сколько точек вокруг ключевой будет учитываться (обычно 16 точек). Далее для каждой точки строится гистограмма, аналогичная гистограмме ориентации ключевой точки, но с меньшим числом компонент (8 компонент вместо 36). Далее формируется вектор из 128 (16x8) чисел. Этот вектор и есть дескриптор SIFT



Обучение модели

Например, используется SVM.

Мы задаем координаты точек и их цвета. После чего ищется оптимальная прямая (гиперплоскость). После этого ответом на вопрос будет являться то, к какой половине (части плоскости) принадлежит новая точка (вектор)



ИСТОЧНИКИ

- <https://www.learnopencv.com/image-recognition-and-object-detection-part1/>
- https://en.wikipedia.org/wiki/Scale-invariant_feature_transform
- https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
- <https://habr.com/ru/post/106302/>
- <https://www.hse.ru/data/2014/05/30/1325164066/%D0%92%D0%9A%D0%A0.pdf>