

День 1: HOG vs CNN

1 HOG дескрипторы. Введение

Первым заданием будет подсчитать HOG фичи используя функцию `hog()` из библиотеки *scikit-image*. Вот пример использования этой функции.

Визуализировать гистограмму активаций для отдельного блока и градиенты для нескольких изображений на ваш выбор.

Найти примеры изображений на которых HOG дескриптор не даёт приемлемых результатов.

2 HOG дескрипторы. Похожесть.

Сейчас мы реализуем 2 простых дескриптора основанных на представлении гистограмм, а также функцию для нахождения совпадений между двумя наборами дескрипторов.

Напишите ф-ю `descriptors_hog(img, x, y, m)` для подсчёта HoG фич в точках $p = (x, y)$ для картинки *img* с размером окна m . Используйте функцию `hog()` из первого упражнения. Функция возвращает матрицу размера $length(p) \times d$, где d размерность фичей.

Модифицируйте `descriptors_hog(img, x, y, m)` заменив `hog()` на функцию `histrg()` прикрепленную к упражнению. Новая функция должна иметь имя `descriptors_color(img, x, y, m)`

Напишите ф-ю `findnn` вида

$$[Idx, Dist] = \text{descriptors_hog}(D_1, D_2),$$

которая находит для каждого дескриптора из набора D_1 ближайший дескриптор из D_2 используя либо евклидово расстояние, либо χ^2 расстояние.

3 HOG vs CNN. FashionMNIST.

В этом упражнении мы сравним эффективность HOG фичей + SVM с эффективностью CNN.

Посчитайте HOG дескрипторы для всех MNIST изображений. Натренируйте на этих фичах SVM с `rbf`-ядром. Сравните результат на `train` и `val` датасетах. Визуализируйте `confusion matrix`.

Затем используя стандартный пример CNN для MNIST от `tensorflow` просто натренируйте сеть на MNIST датасете.

Но, как известно, MNIST слишком простой датасэт, так что давайте потренируемся на чём-то большем: возьмём FashionMNIST датасэт собранный компанией Zalando. Проведите такой же эксперимент с классификатором на HOG фичах уже на FashionMNIST датасэте. Затем опять натренируйте CNN на этом датасэте. Как изменилось отношение между двумя подходами на более сложном датасэте.