Распознавание автодорожных знаков

Влад Шахуро

Graphics & Media Lab
Vision Group

Обзор задания

В данном задании предлагается реализовать классификацию автодорожных знаков с помощью SVM и признаков HOG. Такой классификатор может использоваться для автоматического составления автомобильных карт, систем помощи водителю и в роботизированных транспортных средствах.

Описание задания

В данном задании необходимо написать собственную реализацию подсчёта гистограмм ориентированных градиентов, и затем найти оптимальные параметры классификатора SVM. Опишем схему вычисления HOG:

1. Вычисляются производные изображения I_x и I_y путем свертки с обычными разностными ядрами или ядрами Собеля:

$$D_x = (-1\ 0\ 1), \quad D_y = (-1\ 0\ 1)^T,$$

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

2. Вычисляется модуль градиента по формуле

$$|G| = \sqrt{I_x^2 + I_y^2},$$

   и направление градиента по формуле

$$\Theta = \text{atan2}(I_y, I_x),$$

   где atan2 — знаковый арктангенс, принимающий значения от $-\pi$ до π.

3. (опционально) Направления градиента зеркалируются и приводятся к значениям от 0 до π.

4. Изображение разбивается на ячейки размером $cellRows \times cellCols$ пикселей и для каждой ячейки строится гистограмма направлений с $binCount$ корзин. Пиксель ячейки входит в одну из корзин гистограммы с весом, равным модулю градиента в данном пикселе. В простейшем случае ячейки не пересекаются.

5. Ячейки объединяются в блоки размером $blockRowCells \times blockColCells$, блоки могут пересекаться. Гистограммы различных ячеек в блоке конкатенируются в вектор v и нормируются:

$$v = \frac{v}{\sqrt{|v|^2 + eps}},$$

где $eps > 0$ — небольшое число, исключающее деление на ноль.

6. Конкатенация векторов v из всех блоков является дескриптором изображения.

Интерфейс программы, данные и скрипт для тестирования

Необходимо реализовать две функции. Функция извлечения признаков HOG `extract_hog` принимает на вход изображение. Извлечение признаков нужно реализовать самостоятельно. Вторая функция — функция классификации `fit_and_classify` обучает и тестирует SVM.Функция не должна осуществлять поиск оптимальных параметров для классификатора, найденные параметры должны быть уже подставлены. Использовать можно линейный или нелинейный SVM из библиотеки scikit-learn, для нелинейного нужно дополнительно подбирать параметры ядра.

Для обучения алгоритма выдается публичная выборка знаков. Программа тестируется на двух тестах. В первом тесте алгоритм обучается и тестируется на публичной выборке, во втором тесте — обучается на публичной выборке, а тестируется на скрытой выборке. Точность acc классификации на скрытой выборке конвертируется в итоговый балл:

$acc \geqslant 0.93$ — 10 баллов

$acc \geqslant 0.90$ — 8 баллов

$acc \geqslant 0.85$ — 6 баллов

$acc \geqslant 0.80$ — 4 балла

$acc \geqslant 0.75$ — 2 балла

$acc \geqslant 0$ — 1 балл

Результаты второго теста и итоговый балл скрыты до окончания срока сдачи задания. Итоговый балл считается по последней посылке с ненулевой точностью. Чтобы избежать переобучения на публичной выборке (и низкого результата на скрытой выборке), используйте кросс-валидацию на публичной выборке. Время работы на одном тесте ограничивается 6 часами. Максимальное количество потребляемой оперативной памяти — 4 Гб.

Полезные ресурсы

Dalal, Triggs. Histograms of Oriented Gradients for Human Detection — оригинальная статья про HOG. HOGgles: Visualizing Object Detection Features — статья и визуализации про построение изображения по HOG-признакам.