

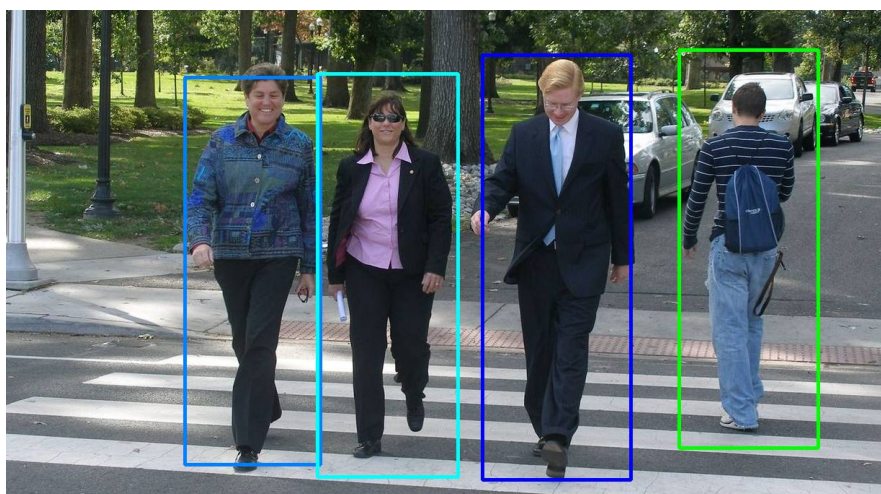
Выделение пешеходов

Влад Шахуро



Обзор задания

В данном задании предлагается реализовать выделение пешеходов на изображении с помощью скользящего окна, пирамиды изображений, признаков HOG и классификатора linear SVM.



Описание задания

Реализация детектора делится на две логические части: обучение детектора и детектирование объектов с обученной моделью. Базовой частью (**5 баллов**) считается реализация обучения детектора без бутстрэппинга и детектирование на пирамиде изображений без подавления немаксимумов.

Обучение детектора

Для обучения детектора из изображений обучающей выборки вырезаются патчи, соответствующие объектам, и приводятся к одному размеру (например, 128×64 пикселей). Затем из изображений вырезаются патчи, соответствующие фону (т.е. пересекающиеся с патчами, соответствующими объектам, не более чем на $0.1 - 0.2$. Подробнее о мере пересечения см. ниже). Так как патчей с фоном намного больше, чем патчей с объектами, количество выбираемых патчей с фоном нужно ограничить (например, не более 2000 патчей). Кроме того, патчи с фоном можно выбирать по разреженной сетке или случайным образом.

Патчи, содержащие объекты, и патчи, не содержащие объектов, составляют положительные и отрицательные примеры соответственно. Из этих примеров извлекаются признаки HOG и полученная матрица признаков вместе с вектором меток подается на обучение линейному SVM. Подбор оптимальных параметров детектора можно проводить с помощью кросс-валидации.

Качество детектора можно повысить с помощью процедуры бутстрэппинга (в статьях по компьютерному зрению — *bootstrapping* или *hard negative mining*). Процедура состоит в обучении

детектора в несколько стадий: на первой стадии происходит обычное обучение, на всех последующих полученная модель применяется к новым патчам с фоном, в обучающую выборку добавляются ложноположительные срабатывания и модель обучается заново на дополненной обучающей выборке. Кроме того, изображения пешеходов можно зеркалировать.

Детектирование объектов на изображении

Детектирование производится на пирамиде изображений методом скользящего окна. Для изображения, на котором нужно найти объекты, строится пирамида изображений. Минимальный и максимальный размер изображений в пирамиде рассчитывается из размера модели и возможных размеров объектов на изображении. Количество изображений в пирамиде определяет компромисс между скоростью и качеством детектора.

На каждом изображении в пирамиде скользящим окном того же размера, что и размер обученной модели, с фиксированным шагом выделяется патч. Для патча считаются признаки HOG, которые затем передаются на классификацию в линейный SVM. Шаг скользящего окна также определяет компромисс между скоростью и качеством детектора. Предлагается использовать шаг 8 или 4 пикселей.

В окрестности объектов на изображении детектор будет давать множественные отклики. Их можно убрать с помощью процедуры подавления немаксимумов. На вход процедура получает окна и меру уверенности классификатора в том, что это окно содержит объект. Окна сортируются по убыванию мер уверенности окон. В цикле выбирается самое вероятное окно, убираются все окна, мера пересечения которых с данным окном не меньше фиксированного параметра (например, 0.5), и самое вероятное окно помечается как просмотренное. Мера пересечения двух окон определяется как отношение площади пересечения окон к площади объединения окон.

Для ускорения детектора часто используется техника предпросчета признаков. Эта техника основана на том, что патчи, извлекаемые скользящим окном, частично пересекаются. Вместо того, чтобы для каждого патча изображения считать градиенты, гистограммы в ячейках и нормированные блоки, их можно предпросчитать для изображения целиком, и затем для каждого патча только конкатенировать нужные блоки. При этом предполагается, что размер ячейки равен 8×8 пикселей и шаг скользящего окна равен 8 пикселей. Для скользящего окна с шагом 4 пикселя нужно предпросчитать дополнительно ячейки, сдвинутые на 4 пикселя, и по ним нормированные блоки.

Интерфейс программы, данные и скрипт для тестирования

Необходимо реализовать две функции: `train_detector`, обучающую модель детектора, и `detect`, проводящую детектирование объектов на одном изображении с обученной моделью. Функция детектирования возвращает матрицу окон размера $N \times 5$, где N — количество окон. Каждая строка матрицы описывает окно — $[x_0, y_0, x_1, y_1, measure]$, где x_0, y_0 — координаты левого верхнего угла окна, x_1, y_1 — координаты правого нижнего угла окна, $measure$ — мера уверенности классификатора для данного окна. Окна с разных изображений пирамиды нужно привести к исходному масштабу изображения.

Скрипт для тестирования `detection_test` принимает на вход директорию с тренировочной и тестовой выборками, обучает детектор и оценивает качество детектирования путем подсчета площади под Precision-Recall-кривой. Результаты детекторов будут проранжированы и в зависимости от качества можно будет получить до 10 бонусных баллов. Максимальное время обучения детектора — 1 сутки, максимальное время детектирования на 1 изображении — 3 минуты.

Полезные ресурсы

[Dalal, Triggs. Histograms of Oriented Gradients for Human Detection](#) — оригинальная статья про HOG.