

Санкт-Петербургский государственный университет
Математико-Механический факультет
Кафедра системного программирования

Снижение разрешения гистограммы
ориентированных градиентов для
детекции объектов малого размера

Курсовая работа студента 371 группы
Русинова Павла Александровича

Научный руководитель к.ф.-м.н, доц. А.Т. Вахитов

Санкт-Петербург
2014

Оглавление

[Оглавление](#)

[Введение](#)

[Постановка задачи](#)

[Обзор существующих решений](#)

[Каскадный HOG:](#)

[Вычисление HOG-дескриптора](#)

[SVM-классификатор](#)

[Алгоритм](#)

[Реализация](#)

[Тестирование](#)

[Заключение](#)

[Список литературы](#)

Введение

Гистограмма направленных градиентов (HOG-дескриптор) является преобразованием изображения в многомерный вектор, таким что сопоставляет изображениям схожих объектов близкие по значению вектора. Такое преобразование является необходимым для успешного использования классификатора (SVM-классификатор, нейросеть). Для классификации HOG-дескрипторов обычно используется SVM.

На практике в компьютерном зрении чаще всего возникает следующая задача: на произвольном изображении необходимо найти все интересующие нас объекты. Для этого используется метод скользящего окна по нескольким масштабам. Суть метода заключается в следующем: изначальное входное изображение масштабируется несколько раз в большую и меньшую стороны, т. е. получается несколько изображений разного размера (необходимость данного шага обусловлена тем, что изображение может содержать объекты разного размера). По каждому из этих изображений выполняется проход окном определенного размера (для которого обучен классификатор). На каждом шаге окна по изображению вычисляется HOG-дескриптор попавшей в это окно части изображения, далее данный дескриптор отправляется классификатору и если классификатор дает положительный ответ, эта часть изображения обозначается как содержащая искомый объект.

Таким образом, при использовании данного метода для обработки одного изображения необходимо вычислить до нескольких тысяч HOG-дескрипторов, что является очень ресурсоёмкой задачей. Данная проблема вызывает трудности при построении систем распознавания, в особенности при построении систем, работающих в реальном времени. Поэтому оптимизация метода скользящего окна по нескольким масштабам является актуальной.

Постановка задачи

Возможный подход к оптимизации может быть следующим: из исходного обученного классификатора получить несколько классификаторов для окон разных размеров. Затем при обработке изображения проходить по нему окнами разных размеров (соответствующих полученным классификаторам). Для ускорения работы метода скользящего окна при детекции объектов малого размера можно использовать классификаторы для уменьшенных окон вместо увеличения изображения и использования исходного классификатора.

Целью данной работы является разработка и реализация алгоритма масштабирования HOG-дескрипторов, алгоритма получения новых классификаторов для окон других размеров из исходного классификатора и последующая реализация

метода скользящего окна по нескольким масштабам, использующего масштабированные классификаторы (в данной работе в качестве классификатора используется SVM-классификатор).

Для выполнения курсовой работы были поставлены следующие задачи:

- Изучить структуру HOG-дескриптора и алгоритм его получения, изучить алгоритм работы SVM-классификатора
- Изучить существующие подходы к оптимизации метода скользящего окна
- Реализовать алгоритм масштабирования HOG-дескриптора и SVM-классификатора, обученного на наборе HOG-дескрипторов.
- Реализовать метод скользящего окна, использующий масштабированные классификаторы
- Протестировать его
- Сравнить метод с первоначальным подходом

Обзор существующих решений

Каскадный HOG:

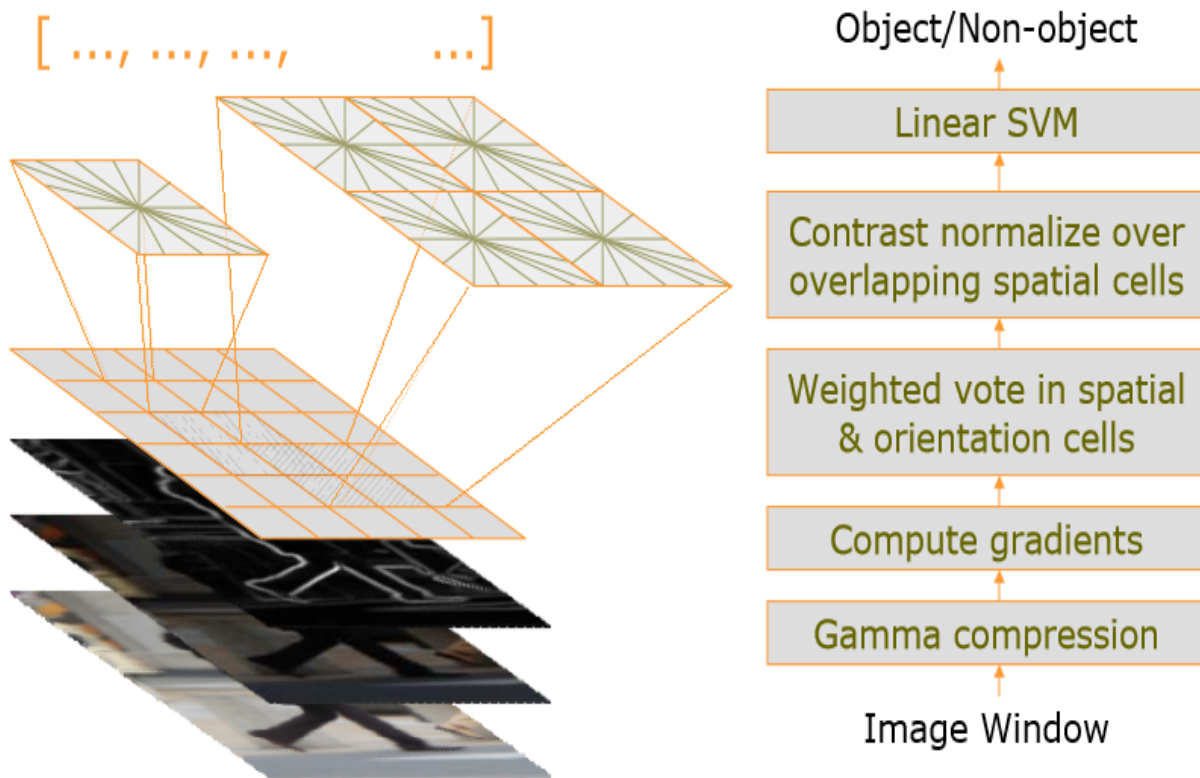
При данном подходе при вычислении HOG-дескриптора используются блоки разных размеров. Итоговый классификатор получается из набора слабых классификаторов (их точность немного выше, чем если просто делать случайные предположения).

Слабые классификаторы выбираются из сгенерированных классификаторов, у которых блоки разного размера расположены случайным образом. Таким образом получается каскад классификаторов, первые из которых работают быстрее и отвергают заведомо не содержащие искомым объектов окна.

Этот подход позволяет добиться ускорения обработки изображения до 30 раз, но при этом падает точность распознавания.

Вычисление HOG-дескриптора

HOG-дескриптор из исходного изображения вычисляется следующим образом: На первом шаге выполняется нормализация цвета и гамма-коррекция. Далее вычисляются значения градиентов по вертикали и по горизонтали. Затем изображение разбивается на равномерную сетку ячеек и для каждой ячейки вычисляется ее гистограмма направлений градиента. Каждый пиксель в ячейке участвует во взвешенном голосовании для каналов гистограммы. После того как гистограммы посчитаны, ячейки объединяются в более крупные связанные области - блоки (блоки могут перекрываться, то есть иметь общие ячейки). После этого происходит нормирование значений каждого блока. Таким образом, HOG-дескриптор является объединением нормированных блоков. Далее вычисленный дескриптор может быть классифицирован.



SVM-классификатор

Задача классификации заключается в следующем: по входному объекту, который представлен n -мерным вектором определить к какому классу он принадлежит.

Обучающей выборкой является набор векторов, при этом известно к какому классу принадлежит каждый вектор. При обучении классификатора строится приближенная функция, которая сопоставляет произвольному объекту класс, которому этот объект принадлежит.

В случае SVM-классификатора, при обучении по обучающей выборке строится гиперплоскость, разделяющая пространство объектов на классы. Она задается следующим уравнением:

$\langle w, x \rangle = b$, где w - перпендикуляр к разделяющей гиперплоскости; b - скалярный порог;

При этом вектор w является линейной комбинацией опорных векторов - элементов обучающей выборки, которые лежат ближе всего к разделяющей гиперплоскости.

В случае использования в качестве объектов классификации HOG-дескрипторов опорными векторами является некоторый набор HOG-дескрипторов из обучающей выборки. Поэтому для получения SVM-классификаторов для окон новых размеров из исходного классификатора можно масштабировать его опорные вектора, рассматривая их как HOG-дескрипторы.

Алгоритм

Входные данные: набор опорных векторов с соответствующими коэффициентами (использующимися при получении перпендикуляра w , как линейной комбинации опорных векторов) обученного SVM-классификатора.

Каждый опорный вектор является HOG-дескриптором из обучающей выборки. Он состоит из набора блоков, каждый из которых является объединением гистограмм ячеек, лежащих внутри этого блока. Из этого вектора генерируются изображения в количестве равному числу значений в одном блоке. Каждое изображение получается следующим образом: ширина изображения - количество блоков по горизонтали, высота - количество блоков по вертикали; значение пикселя (i, j) изображения под номером k извлекается из k значения блока (i, j) . Затем все изображения масштабируются до необходимого размера, соответствующего новому дескриптору (были опробованы билинейная и бикубическая интерполяция), после чего из них генерируется новый дескриптор, соответствующий окну другого размера.

Таким образом масштабируются все опорные вектора, а соответствующие им коэффициенты умножаются на отношение $(img_area)/(new_img_area)$, где img_area - площадь окна, которому соответствует исходный HOG-дескриптор, new_img_area - площадь окна, которому соответствует масштабированный HOG-дескриптор.

После того как все опорные вектора отмасштабированы, получается SVM-классификатор, обученный для окна другого размера.

Далее полученные SVM-классификаторы, можно использовать для применения метода скользящего окна.

Реализация

Алгоритм масштабирования SVM-классификатора и метод скользящего окна с использованием результатов работы этого алгоритма были реализованы на языке C++ с библиотекой OpenCV.

Тестирование

Алгоритм масштабирования SVM-классификатора был протестирован на базе изображений рукописных цифр MNIST. Классификатор для окон исходного размера (28x28) показал точность распознавания 99.5%. Ниже приведены результаты для масштабированных классификаторов.

Размер изображения:	20x20	24x24	44x44	60x60	76x76
Билинейная интерполяция:	97.67%	98.96%	98.91%	96.31%	91.06%
Бикубическая интерполяция:	99.20%	99.29%	99.24%	97.78%	95.50%

Был реализован и протестирован метод скользящего окна, использующий масштабированные классификаторы(для окон большего и меньшего размера) Также был реализован и протестирован метод скользящего окна, использующий только уменьшенные классификаторы (вместо увеличения входного изображения и прохода по нему окном того же размера - проход по входному изображению классификатором для меньшего окна)

В качестве базы изображений была использована база изображений пешеходов INRIA Person. Метод, использующий масштабированные классификаторы(в большую и меньшую стороны) продемонстрировал падение точности распознавания на 16.3%. Метод, использующий уменьшенные классификаторы показал увеличение скорости работы на 40% при падении точности распознавания на 4.6%.

Заключение

При выполнении курсовой работы удалось достичь следующих результатов:

- Изучена структура HOG-дескриптора и алгоритм его получения, изучен алгоритм работы SVM-классификатора
- Изучены существующие подходы к оптимизации метода скользящего окна
- Реализован алгоритм масштабирования HOG-дескриптора и SVM-классификатора, обученного на наборе HOG-дескрипторов
- Реализован метод скользящего окна, использующий масштабированные классификаторы
- Алгоритм масштабирования классификаторов протестирован на базе изображений рукописных цифр и показал приемлемое падение точности распознавания
- Был протестирован метод скользящего окна, использующий масштабированные классификаторы; падение точности в случае использования классификаторов, масштабированных в большую и меньшую стороны составило 16.3%, в случае использования только уменьшенных классификаторов падение точности составило 4.6%, увеличение скорости работы - 40%.

Дальнейшие перспективы: усовершенствование алгоритма масштабирования HOG-дескрипторов (тем самым усовершенствование алгоритма масштабирования SVM-классификаторов, использующего его) для уменьшения падения точности распознавания.

Список литературы

- [1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
- [2] Qiang Zhu, et al., "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients"
- [3] P. Dollár, S. Belongie and P. Perona The Fastest Pedestrian Detector in the West BMVC 2010, Aberystwyth, UK.
- [4] S. Bauer, S. Kohler, K. Doll, and U. Brunsmann, "Fpga-gpu architecture for kernel svm pedestrian detection," in Computer Vision and Pattern Recognition Workshops (CVPRW), 2010, pp. 61 –68.