

ОБ ОДНОМ МЕТОДЕ ПОВЫШЕНИЯ СКОРОСТИ ПОИСКА ОБЪЕКТОВ МЕТОДОМ СКРЫТЫХ ОПОРНЫХ ВЕКТОРОВ ЗА СЧЕТ ПРИМЕНЕНИЯ КАСКАДНЫХ СХЕМ

Н.Ю. Золотых, Е.А. Козинов, В.Д. Кустикова, И.Б. Мееров, А.Н. Половинкин

Нижегородский госуниверситет им. Н.И. Лобачевского

E-mail: nikolai.zolotykh@gmail.com, evgeniy.kozinov@gmail.com, valentina.kustikova@gmail.com, mib@uic.nnov.ru, alexey.polovinkin@gmail.com

Введение

Статья является развитием работы [4] и направлена на повышение скорости детектирования объектов. Рассматривается реализация каскадного метода Latent SVM [1]. Дается анализ точности детектирования объектов на тестовом множестве изображений PASCAL Visual Object Classes Challenge 2007 (VOC 2007, <http://pascallin.ecs.soton.ac.uk/challenges/VOC/>). Сравнивается скорость детектирования объектов с использованием базового метода Latent SVM [2] и его каскадной модификации [1].

Разработка ведется в рамках популярной библиотеки компьютерного зрения с открытым исходным кодом OpenCV (<http://sourceforge.net/projects/opencvlibrary/>).

Постановка задачи

Задача детектирования объектов состоит в том, чтобы определить наличие объекта на изображении и найти его положение. Положение определяется координатами прямоугольника, окаймляющего объект. Задача детектирования является сложной. Сложность обусловлена многими причинами, среди которых можно выделить несколько основных:

1. Разнообразие форм и цветов представителей класса объектов. Например, в случае множества транспортных средств – это разнообразие моделей и цветов автомобилей.
2. Перекрытие детектируемых объектов. Естественная ситуация, когда с точки зрения камеры объекты видны только частично, например, обозревается только крыша и ветровое стекло автомобиля.
3. Разная степень освещенности объектов. В зависимости от времени суток одни и те же объекты могут выглядеть абсолютно по-разному.

Схема решения

Множество всех методов решения задачи детектирования условно делят на три группы:

- 1) методы, основанные на извлечении признаков, наиболее характерных для объекта некоторого класса (машины, автобусы, люди и т.д.);
- 2) методы поиска объекта по заданному шаблону;
- 3) методы детектирования движущихся объектов.

Метод поиска объектов Latent SVM относится к первой группе методов. Формально алгоритм можно разделить на две части:

- 1) построение пирамиды признаков;
- 2) поиск объекта на всех уровнях пирамиды признаков.

Построение пирамиды признаков состоит из следующих действий:

1. Масштабирование исходного изображения. Данная процедура позволяет получить набор изображений с увеличенным и уменьшенным разрешением – пирамиду изображений. Этап масштабирования обеспечивает поиск объектов разного размера.
2. Построение матриц векторов признаков для каждого изображения в пирамиде. Указанная процедура включает два этапа:
 - а) Разбиение изображения на блоки фиксированного размера. Блок состоит из подмножества пикселей, каждый из которых характеризуется интенсивностью цвета.
 - б) Вычисление значений компонент вектора признаков для блока на основании гистограммы ориентированных градиентов (histogram of oriented gradients (HOG) features, [2]). Под градиентом понимается разность интенсивностей соседних пикселей.

Для ускорения поиска объектов в работе [1] также предлагается построить вторую пирамиду признаков, содержащую меньшее количество элементов в наборах векторов признаков. Вторая пирамида получается из первой путем применения PCA (Principle Component Analysis) преобразования [5] к исходным наборам векторов признаков. По существу построение сводится к проекции векторов признаков в пространство меньшей размерности. Коэффициенты матрицы преобразования вычисляются на всей тестовой выборке и являются едиными для объектов всех классов.

Модель объекта некоторого класса определяется набором компонент, каждая из которых соответствует одному из ракурсов этого объекта. Компонента включает совокупность фильтров [2]:

- грубый фильтр, определяющий набор векторов признаков, наиболее характерных для всего объекта;
- совокупность точных фильтров, описывающих отдельные части объекта.

Для повышения скорости поиска объектов в каскадном методе Latent SVM в модель включается дополнительная информация [1]:

- PCA-проекции грубого и точных фильтров;
- по два порога для каждого фильтра и его проекции:
 - порог идентификации присутствия объекта или его части, который определяет возможность присутствия объекта или его части в определенном месте изображения;
 - порог гипотетического отклонения определяет, насколько часть объекта может быть смещена от своего идеального расположения относительно грубого фильтра.

В исходной реализации [4] алгоритма, описанного в работе [2], существенную часть времени занимает второй этап – этап поиска объектов. Построение пирамиды признаков не претерпело существенных изменений, была добавлена реализация построения дополнительной пирамиды признаков. Вторая часть алгоритма для оптимизации по скорости была полностью переработана.

Идея алгоритма поиска положения объекта состоит в том, чтобы некоторым образом оценить вероятность нахождения объекта во всех возможных положениях пирамиды признаков и выбрать наиболее вероятное. Для определения положения объекта в алгоритмах [1, 2] вычисляется значение оценочной функции для каждого возможного положения. Расположение объекта определяется положением левого верхнего угла грубого фильтра в наборе векторов признаков какого-либо уровня. Оценочная функция стро-

ится как сумма скалярных произведений наборов векторов признаков грубого и точных фильтров модели с соответствующими наборами векторов признаков для изображения.

Основное время при вычислении положения объекта в алгоритме [2] занимало вычисление скалярных произведений векторов признаков грубого и точных фильтров модели с соответствующими наборами векторов признаков для изображения – сверток. Для оптимизации в [1] предлагается существенно сократить количество вычисляемых сверток. Для этого используются серия дополнительных порогов, введенных в рамках новой модели объекта. Модифицированный алгоритм состоит из следующих этапов:

1. Построение множества возможных положений объекта. Для этого применяется преобразованная дополнительная пирамида признаков (проекция исходной пирамиды в пространство меньшей размерности) и соответствующие ей фильтры. На данном этапе выполняются следующие шаги:
 - а) Вычисляется свертка грубого фильтра (РСА) в первом возможном положении (например, в левом верхнем углу изображения на одном из уровней). Если свертка меньше порога идентификации для грубого фильтра, то необходимо перейти к следующему возможному положению фильтра.
 - б) Далее в алгоритме рассматриваются свертки точных фильтров.
 - i) Необходимо взять первый фильтр и посчитать его свертки в некоторой окрестности грубого фильтра.
 - ii) Если сумма свертки точного фильтра и текущей оценки положения объекта меньше порога гипотетического отклонения, то положение грубого фильтра не рассматривается.
 - iii) Среди всех подсчитанных положений выбирается положение с максимальной суммой текущей оценки и свертки. Сумму принимают за текущую оценку положения объекта.
 - iv) Если полученная оценка положения объекта меньше порога деформации для текущего точного фильтра, то рассматривается следующее положение грубого фильтра, иначе анализируется положение следующего точного фильтра.
 - с) Если для всех точных фильтров было найдено расположение, то положение грубого фильтра запоминается как возможное.
2. Определение наиболее вероятных положений объекта среди множества, построенного на предыдущем шаге. На данном этапе используется полная пирамида признаков. Как правило, количество возможных положений не очень большое. Процесс проверки существования объекта на полной пирамиде производится тем же способом, что и для свернутой пирамиды признаков.

При реализации поиска объектов можно заметить, что при вычислении оценки положения соседних грубых фильтров окрестности, в которых вычисляются значения сверток точных фильтров, практически совпадают. Для ускорения вычислений необходимо запоминать однажды вычисленные значения для точных фильтров и использовать их при повторной попытке вычислить свертку.

Таким образом, первый этап за счет уменьшения сложности вычисления сверток позволяет быстро отбросить неперспективные положения грубого фильтра. Второй этап позволяет точно ответить на вопрос о вероятности существования объекта в выбранном положении.

Вычисление положения грубых фильтров, как и в алгоритме [2], проводится в блочных координатах. Следовательно, для найденных положений необходимо выполнить преобразование блочных координат, соответствующих найденным положениям объекта на различных уровнях пирамиды признаков, в пиксельные координаты исходного изображения. Полученное множество точек – координаты левых верхних углов

окаймляющих прямоугольников. Далее необходимо определить границы окаймляющих прямоугольников на основании размеров грубого фильтра.

Результаты экспериментов

В качестве тестового набора использовалась база данных конкурса PASCAL VOC 2007, содержащая различные изображения объектов двадцати классов (aeroplane, bicycle, bird, bottle и др.). Представленные фотографии различаются размером изображенных на них объектов, их положением на сцене, ракурсом и степенью освещенности. Указанные факторы оказывают значительное влияние на точность построенной модели.

Так как исходная задача состояла в реализации алгоритма поиска и не включала в себя реализацию алгоритма обучения, для проведения экспериментов были использованы модели авторов статьи [1], преобразованные в формат xml в соответствии с разработанной структурой.

Оценка качества детектирования объектов с помощью реализованного алгоритма вывода выполнялась средствами VOC Development Kit, в состав которого входит модуль, позволяющий вычислить среднюю точность предсказания (average precision). Указанная метрика определяется как математическое ожидание точностей следующим образом:

$$AP = \frac{1}{11} \sum_{r \in [0; 0.1; 0.2; \dots; 1]} p(r), \quad p(r) = \max_{\bar{r}: \bar{r} \geq r} p(\bar{r})$$

где $p(\bar{r}) = \frac{a}{a + c}$ – точность, $\bar{r}$ – процент перекрытия детектированного окаймляющего прямоугольника и прямоугольника, который был размечен на исходном изображении как окаймляющий $\bar{r} \in [0; 0.1; 0.2; \dots; 1]$, a – количество объектов, для которых процент перекрытия не меньше, чем $\bar{r}$ (т. е. считается, что объект детектирован правильно), c – количество объектов с процентом перекрытия, меньшим, чем $\bar{r}$ (объект найден ошибочно).

Вычислительные эксперименты проводились с использованием следующей инфраструктуры:

- язык разработки: C;
- среда разработки: Microsoft Visual Studio 9.0;
- компилятор: Microsoft C/C++ Compiler Version 15.00.30729 (x64);
- процессор: 2 двухъядерных процессора Intel Xeon 5150 (2.66 GHz);
- память: 4 Gb;
- операционная система: Microsoft Windows Server 2008 Standard SP1 x64.

В табл. 1 приведены численные значения средней точности, полученные на текущей реализации алгоритма вывода. Сравнение с результатами, приведенными в [1], показывает, что практически на всех классах объектов наблюдаются незначительные отклонения. Также, в таблице представлены результаты точности детектирования для выполненной последовательной реализации алгоритма [2] на новых моделях из [1]. Из результатов сравнения можно видеть, что точность детектирования во всех трех реализациях практически совпадает. В табл. 1 полужирным начертанием выделены наилучшие точности детектирования для каждого класса объектов.

В данной работе наибольший интерес представляет ускорение от применения новой техники поиска объектов. В табл. 1 показано ускорение времени детектирования, полученное при использовании каскадного Latent SVM. Как видно из таблицы, среднее время детектирования реализации алгоритма [1] составляет 1.5 секунды, в то время как реализация алгоритма [2] выполняет поиск объектов на новых моделях в среднем за

10.5 секунд. Минимально достигнутое ускорение равно 3.6 раза, максимально достигнутое ускорение – 8.5.

Таблица 1. Сравнение точности и скорости работы реализаций алгоритмов вывода [1, 2] на данных VOC 2007

Название класса объектов	Средняя точность (average precision)			Среднее время детектирования		Ускорение
	реализация алг. [1]	реализация алг. [2]	авторская реализация алг. [1]	реализация алг. [1]	реализация алг. [2]	
aeroplane	0,278	0,295	0,227	1,37	10,37	7,57
bicycle	0,57	0,57	0,493	1,38	11,42	8,28
bird	0,094	0,096	0,106	1,72	10,76	6,26
boat	0,153	0,13	0,13	2,14	11,2	5,23
bottle	0,244	0,224	0,266	2	10,19	5,1
bus	0,47	0,47	0,474	1,46	11,05	7,57
car	0,538	0,538	0,502	1,78	9,45	5,31
cat	0,163	0,167	0,188	1,35	10,8	8
chair	0,2	0,19	0,157	2,28	10,94	4,8
cow	0,24	0,23	0,231	1,57	10,88	6,93
diningtable	0,22	0,22	0,113	1,61	11,14	6,92
dog	0,109	0,1	0,123	1,35	11,41	8,45
horse	0,556	0,556	0,36	1,54	10,88	7,06
motorbike	0,449	0,449	0,371	1,29	10,87	8,43
person	0,41	0,4	0,376	2,82	10,12	3,59
pottedplant	0,124	0,113	0,136	2,38	9,71	4,08
sheep	0,18	0,18	0,227	1,73	9,18	5,31
sofa	0,29	0,335	0,231	1,52	10,57	6,95
train	0,44	0,43	0,342	1,42	11,53	8,12
tvmonitor	0,397	0,392	0,4	1,66	11,35	6,84

Заключение

Текущая реализация алгоритма вывода позволяет детектировать объекты с точностью, которая не уступает реализации разработчиков алгоритма, о чем свидетельствуют результаты проведенных экспериментов.

Новая реализация алгоритма позволила, не потеряв в точности, уменьшить время поиска объекта от четырех до восьми раз. Профилировка разработанного приложения показала, что наиболее трудоемким этапом в представленной схеме, как и в первой реализации, является вычисление значений оценочной функции. Поэтому в дальнейшем планируется выполнить алгоритмическую оптимизацию и распараллеливание задачи определения положения окаймляющих прямоугольников (matching) для систем с общей памятью (очевидно, что вычисление значений оценочной функции на каждом уровне пирамиды признаков можно производить независимо), что позволит ускорить обработку изображения в несколько раз.

Авторы благодарят В.Л. Ерухимова (компания Itseez) за ценные замечания и постоянное внимание к работе.

Работа выполнена при поддержке федеральной целевой программы «Научные и научно-педагогические кадры инновационной России», госконтракт 02.740.11.5131.

Литература

1. Felzenszwalb P.F., Girshick R.B., McAllester D., Ramanan D. Cascade object detection with deformable path model // Proceedings of the IEEE CVPR. 2010.
2. Felzenszwalb P.F., Girshick R.B., McAllester D., Ramanan D. Object Detection with Discriminatively Trained Part Based Modes // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. Vol.32, No.9. P. 1627-1645.
3. Fisher R., Dawson-Howe K., Fitzgibbon A., Robertson C., Trucco E. Dictionary of Computer Vision and Image Processing. – John Wiley. – 2005.
4. Золотых Н.Ю., Козинев Е.А., Кустикова В.Д., Мееров И.Б., Половинкин А.Н. Об одном подходе к решению задачи поиска объектов на изображениях // 10-я Международная конференция «Высокопроизводительные вычисления на кластерных системах» (НПС-2010): Материалы конференции (Пермь, ноябрь 1-3, 2010) / Пермь: Изд. ПГТУ, 2010. С.270-277.
5. Pearson K. On lines and planes of closest fit to systems of points in space // Philosophical Magazine. – 1901. – Vol. 2, No. 6. – P. 559-572.