

ций отклика рассматривались: максимальные эквивалентные напряжения в нагелях, максимальные контактные напряжения в нагелях, максимальные эквивалентные напряжения в древесине, максимальные контактные напряжения в опорной металлической пластине.

По полученным уравнениям построены зависимости максимальных напряжений в элементах узлового соединения от расстояний между нагелями по вертикали и горизонтали при фиксированных значениях диаметра нагелей и размера опорной пластины.

Полученные зависимости дают возможность рационального подбора параметров соединения при проектировании большепролетных арок и рам.

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

Нижегородский государственный университет им. Н.И. Лобачевского

ОБ ОДНОМ ПОДХОДЕ К РЕШЕНИЮ ЗАДАЧИ ПОИСКА ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ

Одной из классических задач в компьютерном зрении является задача классификации, заключающаяся в том, чтобы предсказать наличие или отсутствие объекта некоторого класса (машина, птица, человек и т. п.) на изображении или кадрах видео. Наряду с указанной задачей выделяется задача детектирования, в которой необходимо определить конкретное положение объекта. Один из подходов к решению задачи детектирования состоит в использовании алгоритмов машинного обучения для построения модели класса и алгоритмов поиска (вывода) положения объекта на некотором заданном изображении.

В данной работе приводится схема алгоритма вывода, предложенного в [1], описана предлагаемая реализация данного алгоритма, 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/>), а также рассматриваются вопросы производительности текущей реализации. Разработка ведется в рамках популярной библиотеки компьютерного зрения с открытым исходным кодом OpenCV (<http://sourceforge.net/projects/opencvlibrary/>).

Постановка задачи. Решается задача детектирования. Требуется выполнить поиск объектов некоторого класса на изображении (сцена может содержать несколько объектов одного класса). Положение объекта определяется прямоугольником, окаймляющим границы этого объекта. Необходимо построить на изображении множество окаймляющих прямоугольников для объектов заданного класса.

Схема решения. Для решения задачи необходимо реализовать алгоритм вывода [1].

Начальным этапом алгоритма является построение пирамиды признаков, которое состоит из нескольких действий:

1. Масштабирование исходного изображения. Данная процедура позволяет получить набор изображений с увеличенным и уменьшенным разрешением – пирамиду изображений.

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 ориентации градиентов (histogram of oriented gradients (HOG) features, [1]). Под градиентом понимается разность интенсивностей соседних пикселей.

Построенная математическая модель изображения используется для определения расположения прямоугольника, окаймляющего объект. Для решения поставленной задачи необходимо наличие модели, описывающей класс детектируемых объектов. Модель объекта определяется набором компонент, каждая из которых соответствует одному из ракурсов этого объекта (например, если

осуществляется поиск человека на изображении, под ракурсом можно понимать полную фигуру в фас, в профиль и т.д.). Компонента включает совокупность фильтров:

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

Поскольку 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 (для построения модели в [1] применялся латентный метод опорных векторов, Latent SVM).

Идея алгоритма определения положения объекта состоит в том, чтобы некоторым образом оценить вероятность нахождения объекта во всех возможных положениях пирамиды изображений и выбрать наиболее вероятные положения.

Данный алгоритм включает следующие этапы:

1. Вычисление значений оценочной функции для каждого возможного положения объекта. Положение объекта определяется расположением левого верхнего угла грубого фильтра в матрице векторов признаков какого-либо уровня. Оценочная функция строится как сумма скалярных произведений векторов признаков грубого и точных фильтров модели с соответствующими векторами матрицы признаков.

2. Выбор положений, для которых значения оценочной функции превышают пороговое значение (является параметром модели). Полученные координаты определяют положение грубого фильтра, описывающего объект, в матрице признаков. Каждый вектор признаков соответствует некоторому блоку изображения, поэтому необходимо выполнить преобразование координат в координаты пикселей.

3. 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.

4. Вычисление координат правых нижних углов окаймляющих прямоугольников на основании размерности грубого фильтра.

Программная реализация

В алгоритме вывода можно выделить два основных модуля:

1. Модуль построения пирамиды признаков (feature pyramid).
2. Модуль вычисления положения объекта заданного класса (matching).

Для проведения экспериментов установления качества распознавания было разработано тестовое приложение, схема функционирования представлена на рисунке

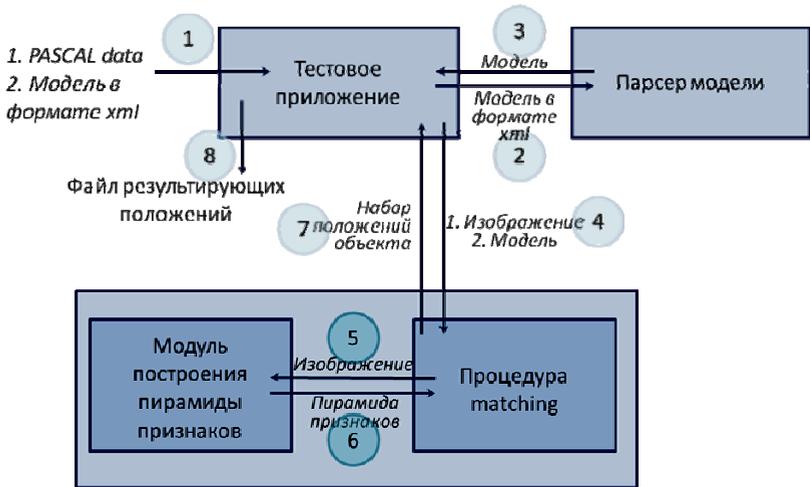


Рис. Схема функционирования тестовой подсистемы

1. Приложение принимает на вход путь до набора изображений и их описаний PASCAL VOC и модель объекта в формате xml (рисунок, п. 1).

2. Xml-файл модели разбирается парсером, который реализован в виде дополнительного модуля (рисунок, п. 2, 3).

3. Для каждого изображения выполняется поиск возможных положений объекта в соответствии с входной моделью (рисунок, п. 4, 7). С этой целью строится пирамида признаков для указанного изображения согласно процедуре, описанной ранее (рисунок, п. 5, 6). Последовательное получение изображений из набора осуществляется с помощью функциональности библиотеки OpenCV.

4. Найденные наиболее вероятные положения объекта сохраняются в результирующем файле (рисунок, п. 8), формат которого приемлем для вычисления метрики качества алгоритма вывода с помощью VOC Development Kit (<http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/index.html#devkit>). Сохранение осуществляется также средствами OpenCV.

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.

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:

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

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 таблице 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, причем как в худшую (наибольшая разница составляет 0,019 на классе bicycle), так и в лучшую сторону (наибольшая разница – 0,023 на классе pottedplant). Значения средней точности, полученные в [1], в настоящее время являются лучшими из известных результатов, о чем свидетельствует первое место в PASCAL VOC Challenge 2009 (detection problem).

Несмотря на практически идентичное качество детектирования, текущая реализация в среднем обрабатывает одно изображение в 2–2,5 раза медленнее, чем реализация авторов алгоритма, которая решает задачу детектирования объектов одного класса на изображении примерно за 2 секунды ([1]). Данный факт объясняется тем, что настоящая версия является последовательной, в отличие от той, с которой проводится сравнение. Дальнейшее распараллеливание для систем с общей памятью позволит достигнуть лучших результатов по скорости.

**Средняя точность детектирования объектов, полученная
с использованием текущей реализации алгоритма вывода
и представленная разработчиками алгоритма,
на данных VOC 2007**

Название класса объектов	Средняя точность (average precision) текущей реализации	Средняя точность (average precision) реализации авторов статьи [1]	Среднее время обработки одного изображения (с)
aeroplane	0,278	0,280	4,205
bicycle	0,525	0,544	4,412
bird	0,006	0,007	3,746
boat	0,123	0,145	3,623
bottle	0,260	0,262	3,438
bus	0,409	0,398	4,435
car	0,457	0,463	3,215
cat	0,163	0,160	5,032
chair	0,154	0,163	4,890
cow	0,167	0,167	4,696
dining-table	0,238	0,243	5,185
dog	0,070	0,050	4,850
horse	0,437	0,438	5,117
motorbike	0,384	0,382	5,120
person	0,338	0,342	3,747
potted-plant	0,101	0,079	3,417
sheep	0,166	0,172	3,130
sofa	0,214	0,221	4,998
train	0,331	0,340	4,208
tvmonitor	0,380	0,393	4,208

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

проведенных экспериментов. Данная реализация интегрирована в библиотеку с открытыми исходными кодами OpenCV, использование которой не требует дополнительного дорогостоящего программного обеспечения.

Среднее время поиска объектов одного класса на изображении превышает время, достигнутое авторами статьи [1] на тех же классах. Профилировка разработанного приложения показала, что наиболее трудоемким этапом в представленной схеме 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.

Список литературы

1. Object Detection with Discriminatively Trained Part Based Modes / P.F. Felzenszwalb [et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2010. – Vol. 32, No. 9. – P. 1627–1645.
2. Dictionary of Computer Vision and Image Processing / R. Fisher [et al.]. – John Wiley. – 2005.