УДК 004.932 DOI: 10.17587/mau/17.187-192

Н. А. Сергиевский, программист-аналитик "Элвис-Неотек", dereyly@gmail.com,
А. А. Харламов, д-р техн. наук, директор ООО "Микросистемы"

Структурное детектирование зрительных образов для мобильного робота

Описывается подход к детектированию объектов в реальном масштабе времени. Процесс детектирования объектов разделен на две части: (1) генерация гипотез и (2) проверка гипотез. Генерация гипотез осуществляется с помощью простой структурной модели на основе комбинации отрезков. Проверка гипотез использует подход на основе сверточных сетей, которые формируют вектор признаков на основе адаптивной подвыборки последнего сверточного слоя. Далее признаки классифицируются алгоримом "случайный лес". Точность данного подхода сопоставима с современными методами детектирования объектов, такими как SPPNet и RCNN, а время работы составляет 4 кадра в секунду на процессоре, что в 7 раз быстрее SPPNet.

***Ключевые слова**: детектирование объектов, компьютерное зрение, зрение роботов, нейронные сети, глубокое обучение, случайный лес*

Введение

В последние годы наблюдается значительное увеличение интереса к робототехнике. Перед робототехникой в настоящее время стоит задача — сделать мобильный и автономный робот, который мог бы взаимодействовать с неизвестной средой и принимать решения без помощи оператора. Зрительный анализатор как дистантный анализатор является для робота средством анализа окружающей среды. Зрение помогает роботу построить карту помещения, локализовать свое положение на ней и отметить на ней объекты, с которыми он будет взаимодействовать.

Для взаимодействия с окружающим миром мобильный робот должен распознавать объекты, фиксируемые видеокамерой, и определять их положение. Для решения этой задачи предназначены алгоритмы детектирования объектов.

На данный момент существует множество подходов к детектированию объектов [1], однако ряд практических задач не могут быть эффективно решены с помощью известных методов и алгоритмов. Основная причина заключается в отсутствии универсальных эффективных методов машинного зрения, способных решать задачи различных классов в реальном масштабе времени. Отсюда вытекает и основная проблема существующих систем детектирования: каждая такая система жестко ориентирована на специфику обрабатываемых данных и детектируемых объектов. При изменении исходных данных либо классов детектируемых объектов необходимо адаптировать систему к изменившимся условиям, если это возможно, а также проводить длительное переобучение на новых наборах тестовых данных. Само детектирование при этом может занимать довольно значительное время. Другой характерной особенностью существующих систем и методов детектирования объектов является многократное использование входного изображения в разных масштабах, что ведет к потере производительности.

Одним из путей решения вышеназванных проблем является разработка нового метода детектирования объектов, основанного на структурном анализе разнотипных характерных особенностей изображения (или изображений видеопотока).

Процесс детектирования объектов можно разбить на несколько этапов. Во-первых, из изображения извлекаются признаки (Хаар [2], SIFT, HOG, признаки, полученные на основе использования сверточных сетей [3, 4]). Затем применяются классификаторы [2, 5] для идентификации признаков в пространстве признаков. Классификаторы применяются в режиме сканирующего окна на пирамиде изображений (признаков) или на некотором наборе регионов, в которых потенциально может находиться объект [6] (регион — это минимальная прямоугольная область, которая содержит объект).

Методы на основе генерации набора регионов или гипотез для последующей классификации дают наибольшую точность на открытых базах VOC Pascal [7], ImageNet [8]. В работе J. Uijlings [6] представлен метод выбора объединенных сегментов (Selective Search), основанный на иерархическом методе объединения сегментов на изображении, которые являются гипотезами о местоположении объекта. Проверка каждой гипотезы осуществляется с помощью метода "мешок слов" [6,9]. Автор P. Dollar описывает алгоритм генерации гипотез на основе контуров EdgeBox [10], который использует обучаемый метод получения контуров.

Для генерации гипотез был разработан быстрый метод BING, работающий за счет поиска контуров на градиентном изображении низкого разрешения, но данный метод достигает невысоких показателей покрытия объектов на изображении [11].

Метод RCNN [12] основан на сверточных сетях, и каждый фрагмент изображения (гипотеза о местоположении объекта) перемасштабируется в изображение размером 224 × 224 × 3 пикселя, что является стандартным входом сети AlexNet [3]. Затем на основе вектора признаков сверточной нейрон-

ной сети регион (гипотеза) классифицируется и уточняется положение окаймляющего прямоугольника. Задача классификации и регрессии решается с помощью линейного метода опорных векторов [5].

В алгоритме SPPNet [13] осуществляется пространственная выборка максимальных признаков (max_pooling) в регионе, полученном с помощью выбора объединенных сегментов (SelectiveSearch), создается несколько наборов ячеек и формируется вектор признаков одинакового размера для регионов разного размера и с разным соотношением сторон. В этом методе использована сверточная нейронная сеть ZFNet, обученная на задаче классификации ImageNet. Для детектирования используется последний сверточный слой, и для определения принадлежности классам или фону и уточнения положения рамки с помощью регрессии обучаются только полносвязные слои. В методе используются признаки, полученные на разных масштабах изображения.

В алгоритме Fast RCNN [14] используется подход, примененный в методе SPPNet [13]. Вектор признаков формируется из предобученной нейронной сети на одном масштабе изображения (600 × 600). Создается специальный слой выбора региона интереса. В отличие от алгоритма SPPNet, не только полносвязные слои, но и вся нейронная сеть дообучается детектированию. Функция потерь одновременно включает компоненты как классификации, так и регрессии. Данный подход дает наибольшую точность на известных тестовых базах по детектированию объектов VOC Pascal, ImageNet.

Несмотря на тот факт, что за последнее время в разы улучшилось качество детектирования [12], причем оно осуществляется почти в реальном масштабе времени [13, 14], остается несколько сдерживающих моментов. Во-первых, поиск гипотез (регионов) работает по-прежнему долго. Во-вторых, показатели алгоритмов получены на дорогостоящих видеокартах, которые потребляют 275...375 Вт [15], что неприемлемо для мобильного робота.

В данной статье будет рассмотрен алгоритм быстрого формирования (генерации) гипотез на основе структурной сочетаемости отрезков и метод проверки гипотез на основе признаков сверточных нейронных сетей.

Задача детектирования объектов на мобильном роботе

Мобильный робот — это устройство с ограниченными вычислительными ресурсами, на котором предполагается использование слабого процессора: i3-i5 или arm. В некоторых случаях доступны графические карты типа Nvidia Tegra. Алгоритмы анализа видеопотока должны работать в реальном масштабе времени, для того чтобы робот смог реагировать на поступающую информацию и процесс обработки видеоданных не занимал бы все ресурсы процессора. Система анализа видеопотока тесно связана с системой навигации и передает в навигационный модуль данные о положении и классе объекта. Изображения объектов, поступающие на вход, не сильно меняются в зависимости от вертикального угла наклона. Объекты, расположенные менее чем на 50 % площади кадра, можно не рассматривать, поскольку они являются несущественными для системы навигации в данный момент времени. Для тестирования задачи детектирования объектов на мобильном роботе была создана база "Стулья" (рис. 1, см. четвертую сторону обложки).

Реальные практические задачи обычно допускают большое число ограничений и допущений. В отличие от классического теста VOC Pascal [7] в базе "Стулья" содержатся объекты большего размера, и перекрытие объектов не может превышать более 50 % площади изображения.

1. Детектирование объектов внутри помещения

Задача детектирования объектов решается в два этапа:

1) формирование гипотез;

2) проверка гипотез.

Гипотезы обычно формируются за счет простых признаков на изображении, таких как сегменты, контуры или границы. Для реализаии зрения роботов внутри помещения хорошим признаком можно считать отрезки. Отрезки принадлежат объектам-артефактам, таким как столы, стулья, двери и т. д. Этот класс объектов вполне охватывает задачу семантической навигации внутри помещения.

Формирование гипотез. К формированию гипотез, как и к прочим компонентам системы детектирования объектов на мобильном роботе, предъявляются требования по быстродействию. Для этого правила формирования гипотезы должны удовлетворять следующим условиям: гипотезы должны вычисляться быстро и быть устойчивыми к простым искажениям, таким как поворот, размытие и артефакты JPEG [11].

Скорость работы алгоритма генерации гипотез часто достигается за счет эффективных процедур проверки гипотез и простоты правил формирования гипотез [16].

Отрезки можно считать структурообразующими особенностями объектов (тех объектов, с которыми может взаимодействовать мобильный робот). Комбинация из нескольких отрезков может дать информацию о возможном нахождении объекта в заданной области и задать эту область с помощью окаймляющего прямоугольника.

Конструкции из отрезков на следующем уровне обработки объединяются в так называемые дескриптивные области (обычно содержащие три отрезка в определенных отношениях друг к другу), которые могут интерпретироваться как элементы объектов. Дескриптивную область можно считать хорошо дискриминирующей структурой.

Для выдвижения гипотез к изображению применяется система шаблонов. Изображение представ-

ляется как набор отрезков прямых. Для каждого отрезка на изображении в его окрестности подбираются соразмерные отрезки для построения комбинации из трех отрезков (рис. 2, см. четвертую сторону обложки). Далее используется геометрическое сравнение с классами отрезков, которое реализуется в несколько шагов:

- находится среднее геометрическое центров (центр масс) отрезков;
- определяется порядок отрезков в комбинации;
- вычисляются расстояния между шаблонами и тройками отрезков на изображении;
- полученные расстояния сравниваются с порогом.

Расстояния между тройками вычисляются по формуле

$$d = c_1 d^L + c_2 d^\alpha + c_3 d^p, \quad (1)$$

где c_i ($i = 1, 2, 3$) — константы; d^L — разница между нормированными длинами отрезков:

$$d^L = \frac{1}{k} \sum_{j=1}^{k} |l_j^{tmpl_1} / K^{tmpl_1} - l_j^{tmpl_2} / K^{tmpl_2}|, \quad (2)$$

здесь l —длина отрезка, $k = 3$ — число отрезков в шаблоне, $tmpl_1$, $tmpl_2$ — индексы разных наборов отрезков (шаблонов), K — нормализация по расстояниям между центрами отрезов:

$$K = \frac{1}{k} \sum_{i=1}^{k} \sum_{j=i+1}^{k} (p_j^{middle} - p_i^{middle})^2, \quad (3)$$

p^{middle} — координаты центра отрезка;

d^α — расстояние между углами в сравниваемых шаблонах:

$$d^\alpha = \frac{1}{k} \sum_{j=1}^{k} (\Delta\alpha_j^{tmpl_1} - \Delta\alpha_j^{tmpl_2})^2, \quad (4)$$

здесь $\Delta\alpha$ — разница между углами наклона пары отрезков; d^p — разность попарных расстояний между концами отрезков:

$$d^p = \frac{1}{k} \sum_{j=1}^{k} \sum_{i=1}^{2} (p_{ij}^{tmpl_1} / K^{tmpl_1} - p_{ij}^{tmpl_2} / K^{tmpl_2}), \quad (5)$$

здесь k — число попарных комбинаций внутри набора отрезков 1-2; 1-3; 2-3, $p_{ij}^{tmpl_n}$ — попарное расстояние между концами отрезков внутри набора отрезков, при $i = 1$ вычисляется расстояние между ближайшими концами отрезков,

Порядок для сопоставления соответствующих отрезков определяется возрастанием угла (по часовой стрелке) от центра масс шаблона до центра отрезка. Угол наклона вычиляется относительно оси абсцисс, т. е. берутся три точки: центр отрезка (x^{middle}, y^{middle}), центр масс шаблона (x^M, y^M), проекция центра отрезка (x^{middle}, y^M).

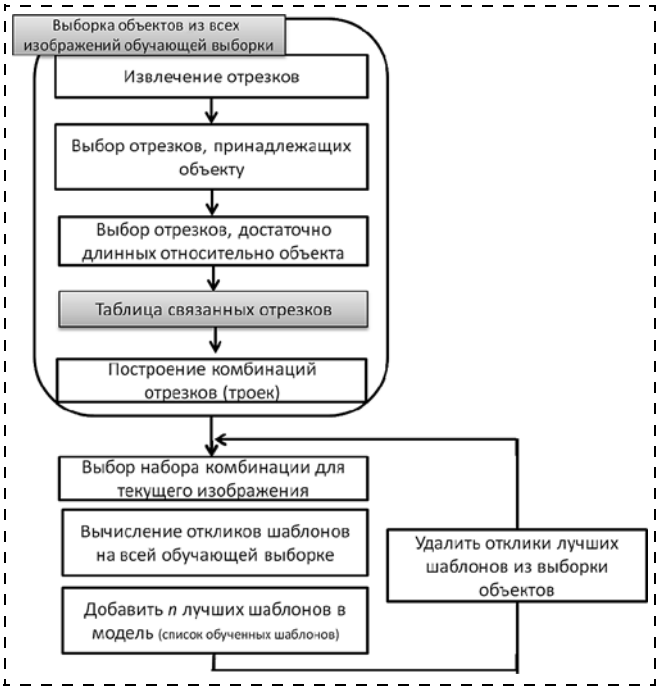

Рис. 3. Алгоритм жадного обучения шаблонов (для генерации гипотез)

Вместо вычисления расстояния между каждой парой шаблонов эффективнее представить шаблон в форме дескриптора:

$$\mathbf{D} = [l_{1...k}/K, \Delta\alpha_{1...k}, p_{1...2,\ 1...k}/K]. \quad (6)$$

Данный дескриптор можно использовать в обучаемых алгоритмах, таких как "случайный лес" или метод ближайшего соседа.

Обучение с помощью метода ближайшего соседа можно представить диаграммой (рис. 3).

Алгоритм генерации гипотез включает следующие шаги:

- поиск отрезков;
- фильтрация парных комбинаций с помощью таблицы связных отрезков;
- построение дескриптора;
- сравнение дескриптора с обученными шаблонами.

Алгоритм генерации гипотез и обучение этого алгоритма являются эффективными с точки зрения требуемого объема вычислений. Отрезки вычисляются с помощью алгоритма EDlines [17], имеющего сложность $O(N)$ (N — число пикселей на изображении). Этап фильтрации гипотез имеет сложность $O(n^2)$ (n — число отрезков) и оставляет только n^2 комбинаций из трех отрезков. Этап формирования дескрипторов имеет сложность $O(n^2)$. Этап сравнения дескрипторов имеет сложность $O(n\lg(m))$, где m — число обученных шаблонов, которые хранятся в kd-дереве [20].

Как видно из рис. 4 (см. четвертую сторону обложки), финальные гипотезы сильно ограничивают возможное положение объекта на сцене.

Проверка гипотез. Стандартной практикой применения нейронных сетей к новой задаче является обучение последнего слоя классам в рассматриваемой задаче с помощью линейного метода опорных векторов. Например, для детектирования объектов RCNN [12] взяли сеть AlexNet, обученную на задаче классификации объектов для 1000 классов, и заменили на классификацию объектов при детектировании в задаче VOC Pascal для 20 классов, т. е. RCNN переобучает только последний (полносвязный) слой нейронной сети.

Если рассмотреть модель SPPNet, которая обучает свой набор полносвязных слоев после адаптивной выборки, то эта модель тратит большую часть времени на вычисление полносвязных слоев [14].

Если рассмотреть алгоритм, использующий признаки с последнего сверточного слоя, то они уже будут линейно неразделимы. В этом случае можно применить другой способ быстрой классификации — "случайный лес".

В сверточной сети последовательно применяются операции свертки, подвыборки (max-pooling) и нелинейной функции активации (ReLU [3] или logsig). Рассмотрим нейронную сеть, представленную на рис. 5. На вход данной сети подается цветное изображение размером (в пикселях) 224 × 224 × 3, и оно сворачивается с 96 фильтрами размером 11 × 11 × 3 и шагом 4, в результате получается набор карт признаков размером 55 × 55 × 96. Этот набор карт признаков проходит через нелинейную функцию ReLU, далее осуществляется свертка с 256 фильтрами 5 × 5 × 48 (нейронная сеть разбита на две части) и применяются операции подвыборки (max-pooling) и нелинейности ReLU. И так далее, последний слой — сверточный: 13 × 13 × 256, что в 16 раз меньше исходного изображения.

Возьмем обученную нейронную сеть и применим ее к изображению $M \times N \times 3$, тогда карта признаков **M** последнего сверточного слоя будет иметь размер $\frac{M}{16} \times \frac{N}{16} \times 256$. Далее нужно вычислить дескриптор в заданной зоне — прямоугольной зоне гипотезы (рис. 5). Применим операцию адаптивной подвыборки (max-pooling) [13]. Для этого вычислим координаты прямоугольника в пространстве карт признаков **M** по формулам [13]

$$x_1' = \text{floor}\left(\frac{x_1 - b_0 + b}{s} + 0{,}5\right);$$

$$x_2' = \text{ceil}\left(\frac{x_2 - b_0 - b}{s} - 0{,}5\right), \qquad (7)$$

где x_1 — координата левого угла прямоугольника, x_2 — правого; x_1', x_2' — координаты в пространстве карт признаков **M**; $s = 16$ — коэффициент уменьшения размера карты признаков относительно исходного изображения; b_0 и b — смещения. Координаты y_1', y_2' вычисляются аналогично формуле (7).

В качестве дескриптора (вектора признаков) объекта, находящегося в заданной прямоугольной области, будем рассматривать не только подвыборку (max-pooling) в карте признаков **M,** но и пространственную решетку с несколькими масштабами $\{m_1 \times m_1; m_2 \times m_2; m_3 \times m_3 ...\}$. Координаты ячеек пространственной решетки вычисляются по формуле (7). Размер вектора признака **V** будет $m_1 \times m_1 \times 256 + m_2 \times m_2 \times 256 + ...$

Для классификации каждой гипотезы используем метод "случайный лес" [19] на пространстве признаков **V.** Данный подход является эффективным с точки зрения затрат на вычисления. Сложность классификации составит $O(Ln_F \lg K)$, где L — число гипотез, n_F — число деревьев, K — размерность **V**.

Практические результаты

Процесс детектирования объектов разделен на две части: (1) генерация гипотез; и (2) проверка гипотез. Каждый из этапов обучался на базе "Стулья". Размер изображения из базы был приведен к 480 пикселям по максимальной из сторон (ширине либо высоте). Использовалась модель нейронной сети CNN-F[18], которая имеет меньше связей и работает быстрее, чем ALexNet. Для обучения второго этапа на изображениях были выделены гипотезы, и в качестве положительных примеров были взяты гипотезы, которые более чем на 50 % (по мере IOU [7]) пересекались с прямоугольником разметки (см. рис. 3, внизу, см. четвертую сторону обложки). Пространственная решетка {1 × 1, 2 × 2, 4 × 4} определяет размерность вектора признаков **V** 5376. Также использовался предыдущий сверточный слой с пространственной решеткой {4 × 4}, размерность вектора признаков составила 9472.

В методе "случайный лес" формируются семь деревьев с глубиной 16 для каждого дерева и случайным подмножеством из 3000 признаков. Точное положение рамки тоже обучалось с помощью метода "случайный лес", была построена регрессионная модель из четырех параметров смещения для прямоугольника, нормированная на ширину гипотезы.

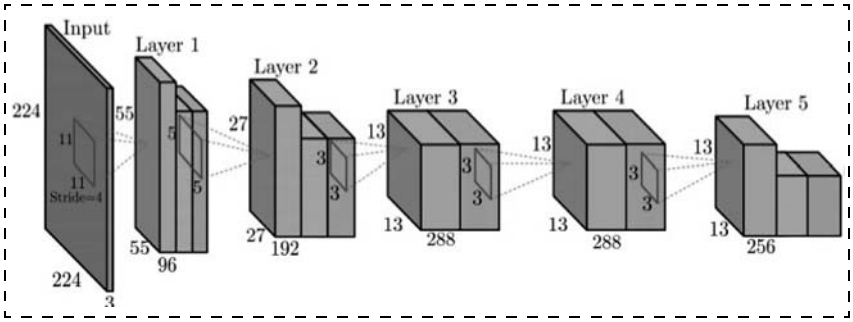

Рис. 5. Сверточная нейронная сеть для классификации изображений [3]. После каждого слоя идет нелинейная функция ReLU. Полносвязные слои отсутствуют в рассматриваемой модели

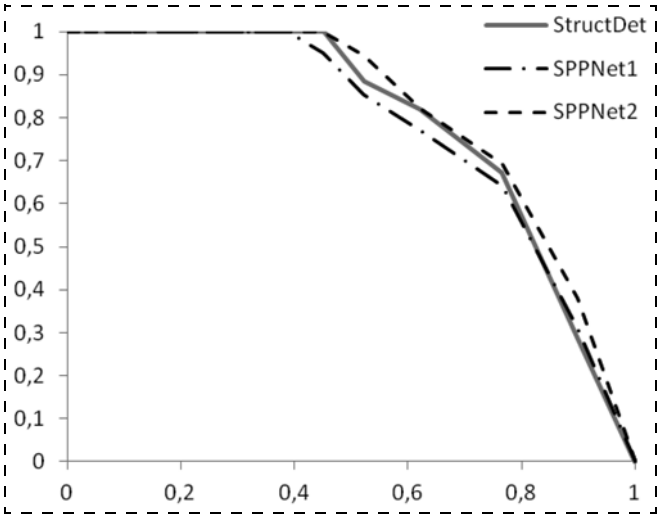

Рис. 6. График точность/полнота на базе "Стулья" для моделей SppNet и структурного детектирования

В качестве отрицательных примеров рассматривали гипотезы, которые менее чем на 30 % пересекаются с прямоугольником разметки. На первом этапе выбирали случайные отрицательные примеры. Далее рассматривали сложные (hard negative) примеры, которые при классификации давали ложные результаты. Сходимость по сложным примерам была достигнута за три итерации по всей обучающей выборке.

Алгоритм показал точность, сравнимую с алгоритмом SPPNet (рис. 6): AP(SPPNet1) = 0,69 против AP(StructDet) = 0,78, но превосходил его по быстродействию: 0,2 с против 1,5 с (на процессоре). Дообученная версия SPPNet на выборке "Стулья" имеет точность AP(SPPNet2) = 0,8.

Полное время работы алгоритма детектирования объектов (включая этап генерации гипотез) составило 0,25 с в одном потоке на CPU i7, что удовлетворяет требованиям, предъявляемым к системе детектирования на мобильном роботе.

Выводы

Необходимость улучшения алгоритмов машинного зрения — это постоянно открытая тема, несмотря на большие достижения в этой области. Ученые в области машинного зрения последнее время работают явно или неявно не с изображениями целиком, а с его фрагментами, используя при этом алгоритмы сегментации, поиска особенностей на изображении, или выстраивая прогрессивные многослойные нейронные сети, реагирующие локально в точках наибольшей информативности. По мнению авторов, основное усилие нужно прилагать к построению моделей с упорядоченной структурой фрагментарного представления объектов.

В работе представлен подход к детектированию сложных объектов на основе использования двух уровней поиска разнородных элементов на изображении с их взаимодействием и построения эффективного множества границ объекта для его последующей классификации.

Продемонстрирован алгоритм детектирования объектов без сканирующего окна, который основан на построении гипотез, и описан каждый шаг и фрагмент этого алгоритма, начиная с поиска особенностей и заканчивая проверкой гипотез.

Показано, что предложенный алгоритм работает в семь раз быстрее известного алгоритма SPPNet. Высокая производительность позволяет использовать алгоритм структурного детектирования для решения практических задач на роботе в реальном масштабе времени.

Список литературы

1. **Girshick, Ross Brook**. From rigid templates to grammars: Object detection with structured models. University of Chicago, 2012.

2. **Paul V., Jones M. J.** Robust real-time face detection // International journal of computer vision. 2004. V. 57, N. 2. P. 137—154.

3. **LeCun Y.**Gradient-based learning applied to document recognition // Proc. of the IEEE. 1998. V. 86, N. 11. P. 2278—2324.

4. **Krizhevsky A., Sutskever I., Hinton G. E.** Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012.

5. **Fan R. E., Chang K. W., Hsieh C. J., Wang, X. R., Lin C. J.** LIBLINEAR: A library for large linear classification // The Journal of Machine Learning Research. 2008. V. 9. P. 1871—1874.

6. **Uijlings J. R., van de Sande K. E., Gevers T., Smeulders A. W.** Selective search for object recognition // International journal of computer vision. 2013. V. 104, N. 2. P. 154—171.

7. **Everingham M., Van Gool L., Williams C. K., Winn J., Zisserman A.** The pascal visual object classes (voc) challenge // International journal of computer vision. 2010. V. 88, N. 2. P. 303—338.

8. **Deng J., Dong W., Socher R., Li L. J., Li K., Fei-Fei L.** Imagenet: A large-scale hierarchical image database // Computer Vision and Pattern Recognition. CVPR 2009. IEEE Conference on IEEE, 2009.

9. **Sivic J., Russell B. C., Efros A., Zisserman A., Freeman W. T.** Discovering objects and their location in images // Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. 2005. V. 1.

10. **Zitnick, Lawrence C., Dollár P.** Edge boxes: Locating object proposals from edges // Computer Vision—ECCV 2014. Springer International Publishing, 2014. P. 391—405.

11. **Hosang J., Benenson R., Schiele B.** How good are detection proposals, really? *arXiv preprint arXiv:1406.6962* (2014).

12. **Girshick R., Donahue J., Darrell T., Malik J.** Rich feature hierarchies for accurate object detection and semantic segmentation // Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.

13. **He K., Zhang X., Ren S., Sun J.** Spatial pyramid pooling in deep convolutional networks for visual recognition // Computer Vision—ECCV 2014. Springer International Publishing, 2014. P. 346—361.

14. **Girshick Ross**. Fast R-CNN. *arXiv preprint arXiv:1504.08083* (2015).

15. **Лобов С. А., Сергиевский Н. А., Харламов А. А.** Адаптация алгоритма сверточных нейронных сетей на ПЛИС // Программные системы: теория и приложения. 2013. № 3 (17).

16. **Cheng M. M., Zhang Z., Lin,W. Y., Torr P.** BING: Binarized normed gradients for objectness estimation at 300fps // Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.

17. **Akinlar, Cuneyt, Cihan Topal.** EDLines: A real-time line segment detector with a false detection control // Pattern Recognition Letters. 2011. V. 32, N. 13. P. 1633—1642.

18. **Chatfield K., Simonyan K., Vedaldi A., Zisserman A.** Return of the devil in the details: Delving deep into convolutional nets // *arXiv preprint arXiv:1405.3531* (2014).

19. **Liaw A., Wiener M**. Classification and regression by randomForest // R news. 2002. V. 2, N. 3. P. 18—22.

Structural Detection of Visual Objects for Mobile Robots

N. A. Sergievskiy, dereyly@gmail.com, ELVEES-NeoTek, Zelenograd, Moscow, 124498, Russian Federation,
A. A. Kharlamov, kharlamov@analyst.ru✉, Institute of Higher Nervous Activity and Neurophysiology of the Russian Academy of Sciences (IHNA&N RAS), Moscow, 117485, Russian Federation,
Moscow State Linguistics University, Moscow, 119034, Russian Federation

Corresponding author: **Kharlamov Aleksandr A.,** Associate Professor, D.Sc., Institute of Higher Nervous Activity and Neurophysiology of the Russian Academy of Sciences (IHNA&N RAS), Moscow, 117485, Russian Federation, Moscow State Linguistics University, Moscow, 119034, Russian Federation, e-mail: kharlamov@analyst.ru

Received on October 06, 2015
Accepted on October 22, 2015

This paper presents StructDetect, a fast method for object detection. The target detection process consists of two stages: generation of a hypothesis (object proposals) (1) and verification of the hypothesis (2). Generation of the object proposals is carried out by means of a simple structural model on the basis of line segment combining. Line segment is detected by EdLines algorithm. Then a computer attributes the line segments and their pairs and creates "a connection table", which filters some combinations. Further, it creates a triple combination of the line segments filtered by "the connection table". Each combination has a handcraft descriptor based on the line segment attribute. This descriptor is used to learn kNN classifier and generate object proposals in the area of 3 line segments. These proposals define a set of candidate bounding boxes available to the detector. The second module is based on a convolutional neural network, which takes a fixed-length feature vector from each region. The convolution neural network computes once per image and features vector extracts with adaptively-sized pooling from the last convolution layer. Then the feature vectors are classified by the random forest algorithm. Accuracy of this approach is comparable with the accuracy of such modern detector methods as SPPNet and RCNN. StructDetect is 7 times faster than SPPNet and has a frame rate of 4fps on a CPU.

Keywords*: object detection, object proposals, computer vision, robot vision, deep learning, random forest*

For citation:

Sergievskiy N. A., Kharlamov A. A. Structural Detection of Visual Objects for Mobile Robots, *Mekhatronika, Avtomatizatsiya, Upravlenie,* 2016, vol. 17, no. 3, pp. 187—192.

DOI: 10.17587/mau/17.187-192

References

1. **Girshick, Ross Brook**. From rigid templates to grammars: Object detection with structured models, University of Chicago, 2012.

2. **Paul V., Jones M. J.** Robust real-time face detection, *International journal of computer vision*, 2004, vol. 57, no. 2, pp. 137—154.

3. **LeCun Y.** Gradient-based learning applied to document recognition, *Proc. of the IEEE*, 1998, vol. 86, no. 11, pp. 2278—2324.

4. **Krizhevsky A., Sutskever I., Hinton G. E.** Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 2012.

5. **Fan R. E., Chang K. W., Hsieh C. J., Wang, X. R., Lin C. J.** LIBLINEAR: A library for large linear classification, *The Journal of Machine Learning Research*, 2008, vol. 9, pp. 1871—1874.

6. **Uijlings J. R., van de Sande K. E., Gevers T., Smeulders A. W.** Selective search for object recognition, *International journal of computer vision*, 2013, vol. 104, no. 2, pp. 154—171.

7. **Everingham M., Van Gool L., Williams C. K., Winn J., Zisserman A.** The pascal visual object classes (voc) challenge,*International journal of computer vision*, 2010, vol. 88, no. 2, pp. 303—338.

8. **Deng J., Dong W., Socher R., Li L. J., Li K., Fei-Fei L.** Imagenet: A large-scale hierarchical image database, *Computer Vision and Pattern Recognition*, *CVPR 2009*, *IEEE Conference on IEEE,* 2009.

9. **Sivic J., Russell B. C., Efros A., Zisserman A., Freeman W. T.** Discovering objects and their location in images, *Computer Vision, ICCV 2005. Tenth IEEE International Conference*. 2005, vol. 1.

10. **Zitnick, Lawrence C., Dollár P.** Edge boxes: Locating object proposals from edges, *Computer Vision—ECCV 2014. Springer International Publishing*, 2014, pp. 391—405.

11. **Hosang J., Benenson R., Schiele B.** How good are detection proposals, really?. *arXiv preprint arXiv:1406.6962* (2014).

12. **Girshick R., Donahue J., Darrell T., Malik J.** Rich feature hierarchies for accurate object detection and semantic segmentation, *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference*, 2014.

13. **He K., Zhang X., Ren S., Sun J.** Spatial pyramid pooling in deep convolutional networks for visual recognition, *Computer Vision—ECCV 2014. Springer International Publishing*, 2014, pp. 346—361.

14. **Girshick Ross**. Fast R-CNN. *arXiv preprint arXiv:1504.08083* (2015).

15. **Lobov S. A., Sergievskiy N. A., Kharlamov A. A**. *Adaptatsiya algoritma svertochnykh neironnykh setei na PLIS* (Adaptaziya algoritma sverochnih neyronnih setey na PLIS), *Programmnye Sistemy: Teoria i Prilogeniyua*, 2013, no. 3(17) (in Russian).

16. **Cheng M. M., Zhang Z., Lin,W. Y., Torr P.** BING: Binarized normed gradients for objectness estimation at 300fps, *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference*, 2014.

17. **Akinlar, Cuneyt, Cihan Topal.** EDLines: A real-time line segment detector with a false detection control, *Pattern Recognition Letters*, 2011, vol. 32, no. 13, pp. 1633—1642.

18. **Chatfield K., Simonyan K., Vedaldi A., Zisserman A.** Return of the devil in the details: Delving deep into convolutional nets, *arXiv preprint arXiv:1405.3531* (2014).

19. **Liaw A., Wiener M**. Classification and regression by randomForest, *R news*, 2002, vol. 2, no. 3, pp. 18—22.