

Системы автоматического распознавания автономеров

12, декабрь 2014

Юзов М. В., Пугачёв Е. К.

УДК: 004.8

Россия, МГТУ им. Н.Э. Баумана

iu6.pugachev@yandex.ru

Введение

Сегодня технологии компьютерного зрения становятся все более востребованными. Системы распознавания автомобильных номеров широко применяются в различных сферах, где требуется идентифицировать автомобиль, например, контроль скоростного режима, въезда на территории различных предприятий, стоянок и т.д. Подобные системы являются неотъемлемой частью обеспечения безопасности.

На рынке присутствует много готовых продуктов, но большинство из них ориентировано на работу на трассе, то есть с автомобилями, движущимися в несколько потоков и на высокой скорости. Этим обусловлена их высокая стоимость. Основными заказчиками таких систем в основном являются госучреждения. Малым предприятиям, где необходимо осуществлять контроль доступа или вести учет автомобилей, нет смысла покупать дорогостоящие системы, ориентированные на другие задачи. Для малых предприятий примером является система «НомерОк». При достаточно низкой цене «НомерОк» имеет ряд недостатков: используется только на ОС Windows, только протокол RTSP и распознает только стандартные номера [1]. Большинство остальных систем работает только с ОС Windows и имеют высокие цены. С целью устранения вышеуказанных недостатков, было принято решение исследовать существующие алгоритмы распознавания номеров и предложить свой подход к реализации системы распознавания.

Обзор алгоритмов

Процесс идентификации автомобильного номера в общем случае делится на три части:

- Локализация номерного знака на изображении;
- Сегментация;
- Распознавание символов.

Рассмотрим эти этапы более подробно.

Локализация

Самый простой и очевидный способ локализации номерной пластины на изображении - поиск прямоугольного объекта с заданным соотношением площади контура к его периметру. Сначала выполняется поиск границ с применением различных фильтров, например, детектор границ Кэнни (рис. 1), а затем производится анализ контуров, в результате которого выделяется прямоугольная область, удовлетворяющая нашим условиям. В рамках данной работы была разработана программа, реализующая данный алгоритм. Программа написана на языке C++ с использованием библиотеки компьютерного зрения OpenCV (рис. 2).



Рисунок 1 – Результат работы детектора границ Кэнни.

```
cvConvertImage(_image, temp, CV_BGR2GRAY); //преобразуем в бинарное
cvSmooth(temp, temp, CV_GAUSSIAN, 3, 0, 0, 0); //гауссовское сглаживание
cvErode(temp, temp, NULL, 1); //эрозия
cvDilate(temp, temp, NULL, 1); //дилатация
cvCanny(temp, temp, 100, 50, 3); //детектор границ Кэнни
```

Рисунок 2 – Вызов детектора границ Кэнни

Второй способ – вертикальная и горизонтальная проекция изображения на оси у и х. Это один из методов статистического анализа. Вертикальную проекцию обычно используют после применения фильтра поиска вертикальных границ (выделения высокочастотных пространственных компонент изображения (рис. 3)). На графике (рис. 4) видно два максимума, которым вероятнее всего соответствует левая и правая границы номера [2]. Недостаток данного метода заключается в том, что на кадре должно быть минимум объектов, кроме машины, то есть она должна занимать весь кадр, иначе возможны ложные срабатывания.



Рисунок 3 – Результат работы фильтра поиска вертикальных границ.

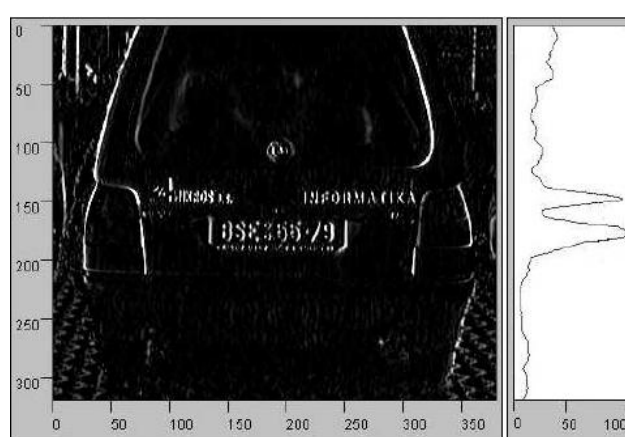


Рисунок 4 – Вертикальная проекция на ось y.

Перечисленные алгоритмы показывают хороший результат на чистых номерах и при нормальных условиях освещенности. Но как быть с грязными номерами? С этой задачей хорошо справляются алгоритмы, решающие задачу классификации, то есть использующие признаки (примитивы) и поддающиеся обучению. Пример такого алгоритма – метод Виолы – Джонса. В стандартном методе Виолы – Джонса используются примитивы Хаара (рис. 5), а в расширенном методе, например, в библиотеке OpenCV используются дополнительные признаки (рис. 6) [3].

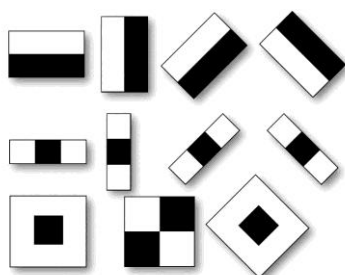


Рисунок 5 – Примитивы Хаара

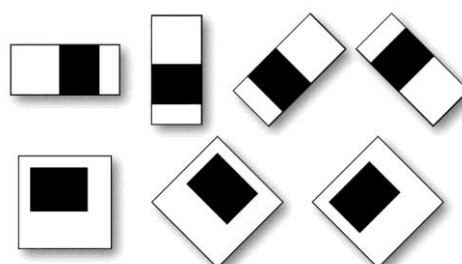


Рисунок 6 – Доп. признаки

Ключевым преимуществом данного метода является его адаптивность, то есть в процессе машинного обучения его можно натренировать на работу с широким спектром входных изображений (различная освещенность, угол наклона номерной пластины и т. д.). Таким образом, алгоритм Виолы – Джонса значительно повышает гибкость системы распознавания и снижает процент не распознанных номеров.

Сегментация

Для выделения букв и цифр (сегментации) на полученном в результате локализации изображении (рис. 7) можно с помощью обычного поиска границ, о котором мы упомянули в самом начале.



Рисунок 7 – Результат выделения символов.

Также можно выполнить горизонтальную проекцию на ось X (рис. 8) и «нарезать» номер, используя максимумы гистограммы [2]. Среди недостатков здесь ложные максимумы на загрязненных номерах.

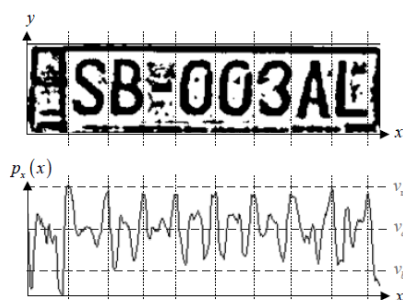


Рисунок 8 – Горизонтальная проекция номера на ось x.

Распознавание

После получения элементов номера, нужно перевести его из графического формата в текстовый – распознать.

Самый простой способ – сравнение с эталоном буквы или цифры и принятие решения на основе расстояния Хэмминга. Этот метод корректно работает только на чистых номерах. Можно повысить его эффективность, прибегнув к помощи математической статистики, но тогда возрастет и вычислительная сложность и сложность реализации.

Наиболее эффективный способ – использование искусственных нейронных сетей (ИНС), например, нейросеть Кохонена, сверточные нейронные сети. В общем случае на вход нейросети подается какой – либо входной вектор, который на основе весов нейросети преобразуется в выходной вектор (рис. 9). Обучение ИНС заключается в нахождении коэффициентов связей между нейронами (весов). В результате обучения нейросети на боль-

шом объеме обучающей выборки, ИНС может вернуть верный результат, на основе искаженных или отсутствовавших в обучающей выборке входных данных, следовательно, можно добиться высокого качества распознавания.

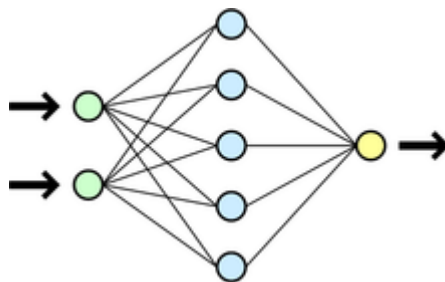


Рисунок 9 – Схема нейросети.

Наконец, можно использовать готовые библиотеки оптического распознавания символов (optical character recognition, OCR). Рассмотрим более подробно свободно распространяемую библиотеку Tesseract. Она принимает на вход как отдельный символ, так и последовательность символов, то есть текст. Интегрировать ее в проект не сложно (рис. 10), а также ее легко обучить, что немаловажно при работе в системах распознавания.

```
...
tesseract::TessBaseAPI *api = new tesseract::TessBaseAPI();
// инициализация
if (api->Init(NULL, "eng")) {
    fprintf(stderr, "Could not initialize tesseract.\n");
    exit(1);
}
// режим распознавания - по одному символу
api->SetPageSegMode(tesseract::PSM_SINGLE_CHAR);
//установка разрешенных символов
api->SetVariable("tessedit_char_whitelist", "АВСЕНКМОРХУ0123456789");
...
//загрузка изображения
api->SetImage( (uchar*)chars[i]->imageData, chars[i]->width, chars[i]->height,
chars[i]->nChannels, chars[i]->widthStep );
//строка результата
res += api->GetUTF8Text()[0];
...
//уничтожение объекта и освобождение памяти
api->End();
```

Рисунок 10 – Инициализация библиотеки и ее вызов

Выводы

В результате было выполнено сравнение наиболее популярных алгоритмов локализации и распознавания номерных знаков. Среди них выявлены наиболее точные и быстродействующие. Так, для локализации номерной пластины на данный момент одним

из самых перспективных алгоритмов является метод Виолы – Джонса или основанные на нем алгоритмы. Для распознавания с низким процентом ложных срабатываний стоит использовать нейронную сеть или какие – либо готовые библиотеки OCR. Например, хорошо себя зарекомендовала система Tesseract OCR.

В рамках работы была разработана система распознавания на основе детектора границ Кэнни и библиотеки Tesseract. На машине с процессором Intel Core i5 с тактовой частотой 2.5 ГГц и объемом оперативной памяти в 4 ГБ время распознавания составляет 0,65 с, а точность 80%. Но это точность для относительно чистых номеров и качественных снимков. Грязные номера программа обрабатывает с ошибками.

Для повышения точности локализации следует перейти к методу Виолы – Джонса. А для распознавания символов с меньшим количеством ошибок, нужно тщательно обучить Tesseract. Также в перспективе можно рассматривать интеграцию в системы учета, например, в 1С.

Список литературы

1. Официальный сайт системы «Номерок». Режим доступа: <http://www.avtonomerok.com/> (дата обращения 24.11.14).
2. Algorithmic and mathematical principles of automatic number plate recognition systems. Режим доступа: <http://javaanpr.sourceforge.net/anpr.pdf> (дата обращения 24.11.14).
3. Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц. Режим доступа: <http://habrahabr.ru/post/133826/> (дата обращения 24.11.14).