

М. А. Митрохин, Я. П. Щеголихин,
Д. О. Нешко, М. В. Семенкин

АНАЛИЗ ЭФФЕКТИВНОСТИ РЕШЕНИЙ ОТКРЫТОЙ БИБЛИОТЕКИ OPENCV В ЗАДАЧЕ РАСПОЗНАНИЯ АВТОМОБИЛЬНЫХ НОМЕРОВ

Аннотация.

Актуальность и цели. Объектом исследования является библиотека *OpenCV*. Предмет исследования – средства библиотеки, использование которых возможно для распознавания автомобильных номеров. Целью работы является оценка эффективности указанных средств при использовании в широком диапазоне условий.

Материалы и методы. Исследования выполнены с использованием методов распознавания образов и обработки изображений.

Результаты. Предложены механизмы совершенствования средств распознавания автомобильных номеров, на экспериментальных данных показана их эффективность.

Выводы. Предложенные решения позволяют повысить эффективность стандартных средств библиотеки *OpenCV*, основанных на каскадах Хаара, однако это требует дополнительных вычислительных затрат.

Ключевые слова: алгоритмы, обработка изображений, каскады Хаара.

М. А. Mitrokhin, Ya. P. Shchegolikhin,
D. O. Neshko, M. V. Semenkin

AN ANALYSIS OF THE OPENCV LIBRARY'S SOLUTIONS EFFICENCY IN THE PROBLEM OF LICENCE PLATE RECOGNITION

Abstract.

Background. The object of the study is the OpenCV library. The subject of the study is the means of the library for the recognition of car licence plate. The aim of the study is to evaluate the effectiveness of OpenCV library in a wide range of image acquisition conditions.

Materials and methods. The research is performed with using of the pattern recognition and image processing methods.

Results. The technique for improving the means of recognition car licence plate are proposed, their efficiency is shown on the experimental data.

Conclusions. The proposed solutions allow to increase the efficiency of standard OpenCV library tools, based on Haar cascades, but it requires additional computing costs.

Keywords: algorithms, image processing, Haar cascades.

Введение

Современные вычислительные средства позволяют решать довольно широкий круг задач автоматизации процессов человеческой деятельности. Это позволяет экономить человеческие ресурсы, сокращать затраты на персонал, не снижая при этом эффективность достижения поставленных целей. Одним из примеров автоматизации процессов являются автоматические системы контроля и наблюдения за автомобильным трафиком. Они позволяют управлять транспортными потоками через регулирование времени работы светофоров, повышать культуру вождения за счет автоматических штрафов, прогнозировать аварийные ситуации.

Создание алгоритмов работы таких систем требует проведения трудоемких исследований, поскольку они должны включать:

- обнаружение движущихся объектов;
- классификацию автомобилей и других транспортных средств;
- идентификацию водителей или владельцев движимого имущества.

Задачей настоящего исследования является определение возможностей идентификации владельцев автомобилей, а именно распознавание государственного регистрационного знака автомобиля по изображению с камер наблюдения стандартными средствами открытых библиотек. Для решения подобных задач создан ряд специализированных алгоритмов машинного зрения, реализованных в общедоступных библиотеках.

Областью интереса машинного зрения являются цифровые устройства ввода/вывода, компьютерные сети, методы и алгоритмы обработки визуальной информации. За счет возрастания сложности решаемых научно-технических задач автоматическая обработка и анализ визуальной информации становятся все более актуальными.

1. Каскады Хаара для распознавания изображений

Для решения поставленной задачи распознавания автомобильных номеров использовался следующий инструментарий:

- язык программирования Python третьей версии. Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций [1];

- открытая библиотека OpenCV (Open Source Computer Vision) третьей версии. OpenCV – библиотека с открытым исходным кодом, в которой реализованы алгоритмы компьютерного зрения, обработки изображений и численных решений общего назначения, написана на языках программирования C и C++ [2].

Для распознавания автомобильных номеров применялась функция `haarcascade_russian_plate_number`. Эта функция реализована с использованием каскадных классификаторов Хаара. Этот метод использует разновидность сканирующей обработки изображения [3] и достаточно эффективно работает в реальном времени. На вход алгоритма подается изображение, для каждой части которого вычисляются признаки Хаара. Для получения признаков вычисляется свертка части изображения с примитивами Хаара:

$$I_t(x, y) = \sum_{u=-k}^k \sum_{v=-l}^l I(x-u, y-v) \cdot g_t(u, v),$$

где $I(x, y)$ – часть исходного изображения; $k = (p-1)/2$ и $l = (q-1)/2$; $g_t(u, v)$ – t -й примитив Хаара размером $p \times q$.

Каждый примитив (рис. 1) переставляет собой несколько смежных прямоугольных зон, каждая из которых может содержать одно из двух значений – плюс 1 (светлая) и минус 1 (темная). Примитивы различаются расположением светлых и темных зон и их ориентацией [4].

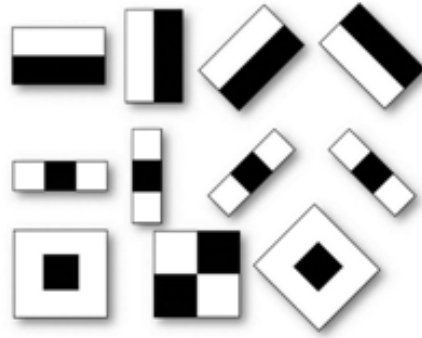


Рис. 1. Примитивы Хаара

По каждой полученной свертке вычисляется признак:

$$f^t = \sum_{(x,y) \in A} I_t(x, y) - \sum_{(x,y) \in B} I_t(x, y), \quad (1)$$

где A – множество координат, соответствующих светлой части примитива Хаара, участвующего в операции свертки; B – множество координат, соответствующих темной части примитива Хаара.

Полученные по выражению (1) признаки подаются на алгоритм классификации [5], который принимает решение в соответствии с выражением

$$h(I) = \begin{cases} 1, & \text{если } \sum_{t=1}^T a_t \cdot h_t(f^t) \geq \frac{1}{2} \sum_{t=1}^T a_t, \\ 0, & \text{иначе,} \end{cases}$$

где T – количество примитивов Хаара; a_t – веса, определяемые в результате работы алгоритма 1.

Алгоритм 1 – Алгоритм обучения классификатора изображений

Вход:

1 Вычисленные признаки для изображений $(f_i, s_i)_{i=1 \dots N}$, где $f_i = (f_i^1, f_i^2, \dots, f_i^T)$ – набор признаков, вычисленных для i -го изображения, $t = 1, \dots, T$, T – количество примитивов Хаара, N – количество примеров, $s_i \in (0,1)$ – метка отрицательного и положительного примера соответственно

Выход: Решение алгоритма классификации

1 Для $i = 1, \dots, N$

1.1 Для $t = 1, \dots, T$

1.1.1 Инициализация веса $w_{t,i} = \frac{1}{2 \cdot m}$ для $(\forall i) (s_i = 0)$ и

$w_{t,i} = \frac{1}{2 \cdot l}$ для $(\forall i) (s_i = 1)$, где m и l – количество отрицательных и положительных примеров соответственно.

2 Для $t = 1, \dots, T$

2.1 Нормализация текущего веса $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^N w_{t,j}}$.

2.2 Обучение классификатора h_t

$$w_{t,i} = \arg \min_{e_t} \left(\sum_{i=1}^N w_{t,i} \cdot \left| h_t(f_i^t) - s_i \right| \right).$$

2.3 Обновление весов $w_{t+1} = w_t \cdot a_t^{1-e_t}$, где $a_t = \log \left(\frac{1-e_t}{e_t} \right)$.

3 Выход

Достоинством описанного подхода является высокая скорость работы и то, что каскады являются достаточно универсальным классификатором для большинства геометрических примитивов и объектов.

2. Экспериментальная оценка и совершенствование решений

При использовании данной функции была выявлена проблема, которая сильно ограничивала возможности ее применения. Функция распознает номера, расположенные лицевой частью к экрану или же с отклонениями в $\pm 2^\circ$ от горизонтального положения. При реальной съемке трудно ожидать выполнения этих условий. Даже при высоком качестве изображения, но при повернутом номере распознавания не происходит (рис. 2).



Рис. 2. Пример изображения, на котором номер не распознается

Это является следствием того, что в наиболее распространенном наборе примитивов Хаара содержатся горизонтально ориентированные и развернутые на 45° примитивы. Решение данной проблемы возможно в двух направлениях. Первое направление заключается в добавлении в набор примитивов, повернутых на углы, отличные от 0 и 45° град. При этом необходимо переобучать алгоритм на исходном или новом наборе данных. Второе направление заключается в предварительном повороте изображения перед классификацией. При этом переобучение алгоритма классификации не потребуется. Так как требуемый угол поворота не известен для каждого изображения, то было решено классифицировать изображение, постепенно поворачивая его от -30° до 30° с шагом в 2° . Такой подход дает положительные результаты (рис. 3) [6].



Рис. 3. Считывание с осуществлением поворота

После применения поворота изображения значительно увеличился диапазон обнаружения номеров, однако обнаружилась проблема, которая заключается в особенностях реализации самого поворота изображения. Когда изображение поступает на вход в программу, создается окно, которое соответствует его размерам, в котором далее и происходит последующая обработка. Однако в нашем способе осуществляется поворот изображения, поэтому края обрабатываемого изображения выходили за пределы окна, где функция не может работать. Достаточно часто за пределы окна выходят и номера, которые нужно обнаружить.

Данная проблема потребовала весьма простого решения. К изображению был добавлен контур так, чтобы за границы окна выходило не само изображение, а лишь его контур. Тогда номера будут оставаться в нужных границах и будут обнаружены. Соответственно, чтобы не добавлять лишний контур по краям фото, необходимо учитывать размеры самого изображения.

Для этого изображение приводится к квадратному виду для автоматического изменения параметров окна.

Далее считываются размеры готового квадрата и осуществляется добавление однотонных полос сверху, снизу, справа и слева. Увеличение сторон отображаемой области изображения в 1,5 раза исключает выход информативных частей за границы экрана [7]. Это значение получено расчетным путем, часть изображения именно такого размера выходит за рамки при повороте изображения на 45° .

Сделанные усовершенствования позволяют распознать номера, который ранее оставались нераспознанными (рис. 4).

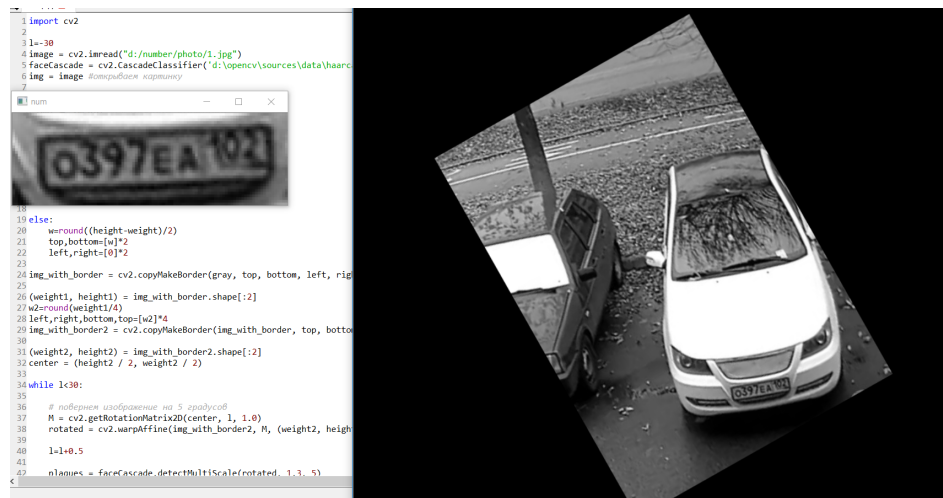


Рис. 4. Пример изображения с распознанным номером

Таким образом, усовершенствованный алгоритм запишется в виде алгоритма 2.

Алгоритм 2 – Алгоритм классификации изображений
Вход:
1 Исходная часть изображения $I(x, y)$
Выход: решение алгоритма
<p>1 Расширение границ изображения</p> $I(x', y') = \begin{cases} I(x, y) & \text{для всех } x' = x \text{ и } y' = y, \\ 0 & \text{для всех } x' = x \pm 0.25 \cdot x \text{ и } y' = y \pm 0.25 \cdot y. \end{cases}$ <p>2 Для $\alpha = -30, \dots, 30$</p> $I_{\alpha}(x', y') = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \cdot I(x', y').$ <p>2.1 Для $t = 1, \dots, T$</p> <p>2.1.1 Вычисление свертки</p> $I_t(x', y') = \sum_{u=-k}^k \sum_{v=-l}^l I(x' - u, y' - v) \cdot g_t(u, v).$ <p>2.1.2 Вычисление признака $f^t = \sum_{(x', y') \in A} I_t(x', y') - \sum_{(x', y') \in B} I_t(x', y').$</p> <p>2.2 Классификация изображения</p> $h(I) = \begin{cases} 1, & \text{если } \sum_{t=1}^T a_t \cdot h_t(f^t) \geq \frac{1}{2} \sum_{t=1}^T a_t, \\ 0, & \text{иначе.} \end{cases}$ <p>3 Выход</p>

Чтобы провести тестирование, была создана тестовая база из 95 фотографий автомобилей, на которых просматриваются государственные регистрационные знаки. База была сформирована по видеозаписям, полученным с камер наблюдения. Изображения автомобилей в тестовой базе получены с разным ракурсом, качеством и разными условиями съемки (время суток, время года, загрязненность автомобилей и номеров) [8].

Тестирование на собранной базе показало, что начальный вариант программы давал неприемлемые результаты, так как распознал лишь 25 номеров из 95, что составило примерно 26 %. Первая доработка позволила повысить количество распознанных номеров до 60 из 95, т.е. 63 %. Конечное усовершенствование программы позволило довести количество распознанных номеров до 88, т.е. 93 % от общего количества.

Заключение

Результатом проведенного исследования стало совершенствование алгоритма распознавания изображений, реализованного в библиотеке *OpenCV*, что позволило повысить эффективность распознавания автомобильных номеров ее стандартными средствами. Недостатком разработанного алгоритма является увеличение вычислительных затрат, так как одно изображение необходимо обрабатывать несколько раз с разными углами поворота. В качестве направления совершенствования для устранения указанного недостатка можно выделить автоматическое определение угла, на который необходимо повернуть изображение.

Библиографический список

1. **Маккинни, У.** Python и анализ данных / Уэс Маккинни. – Москва : ДМК Пресс, 2015. – 482 с.
2. OpenCV шаг за шагом // RoboCraft. – URL: <http://robocraft.ru/blog/computervision/264.html> (дата обращения: 23.02.2019).
3. **Федотов, Н. Г.** Анализ полутоновых изображений и цветных текстур с позиции стохастической геометрии и функционального анализа / Н. Г. Федотов, Д. А. Голдуева, М. А. Мокшанина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2017. – № 2 (42). – С. 29–41. – DOI 10.21685/2072-3059-2017-2-3.
4. Комбинированный алгоритм отслеживания перемещений // Vuzlit.ru. – URL: https://vuzlit.ru/954633/kombinirovannyu_algoritm_otслеzhivaniya_peremescheniy (дата обращения: 26.02.2019).
5. **Viola, P.** Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola, M. Jones // Proceedings Conference of Computer Vision and Pattern Recognition. – 2001. – Vol. 1. – P. 511–518. – DOI 10.1109/CVPR.2001.990517.
6. Играемся с изображениями в Python // Habr. – URL: <https://habr.com/ru/post/163663/> (дата обращения: 25.01.2019).
7. OpenCV Python поворачивает изображение на X градусов вокруг определенной точки // qa.ru. – URL: <http://qa.ru.site/questions/272130/opencv-python-rotate-image-by-x-degrees-around-specific-point> (дата обращения: 27.01.2019).
8. **Лукьяница, А. А.** Цифровая обработка видеоизображений / А. А. Лукьяница, А. Г. Шишкин. – Москва : Ай-Эс-Эс Пресс, 2009. – 512 с.

References

1. Makkinni Ues *Python i analiz dannyykh* [Python and data analysis]. Moscow: DMK Press, 2015, 482 p. [In Russian]

2. *RoboCraft*. Available at: <http://robocraft.ru/blog/computervision/264.html> (accessed Febr. 23, 2019).
3. Fedotov N. G., Goldueva D. A., Mokshanina M. A. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki* [University proceedings. Volga region. Engineering sciences]. 2017, no. 2 (42), pp. 29–41. DOI 10.21685/2072-3059-2017-2-3. [In Russian]
4. *Vuzlit.ru*. Available at: https://vuzlit.ru/954633/kombinirovannyi_algoritmi_otслеживаниya_peremescheniy (accessed Febr. 26, 2019). [In Russian]
5. Viola P., Jones M. *Proceedings Conference of Computer Vision and Pattern Recognition*. 2001, vol. 1, pp. 511–518. DOI 10.1109/CVPR.2001.990517.
6. *Habr*. Available at: <https://habr.com/ru/post/163663/> (accessed Jan. 25, 2019).
7. *OpenCV Python povorachivaet izobrazhenie na X gradusov vokrug opredelennoy tochki* [PenCV Python rotates an image X degrees around a specific point]. qa.ru. Available at: <http://qaru.site/questions/272130/opencv-python-rotate-image-by-x-degrees-around-specific-point> (accessed Jan. 27, 2019). [In Russian]
8. Lukyanitsa A. A., Shishkin A. G. *Tsifrovaya obrabotka videoizobrazheniy* [Video image digital processing]. Moscow: Ay-Es-Es Press, 2009, 512 p. [In Russian]

Митрохин Максим Александрович

доктор технических наук, заведующий кафедрой вычислительной техники, Пензенский государственный университет (Россия, г. Пенза, ул. Красная, 40)

E-mail: vt@pnzgu.ru

Mitrokhin Maksim Aleksandrovich

Doctor of engineering sciences, head of sub-department of computer engineering, Penza State University (40 Krasnaya street, Penza, Russia)

Щеголихин Ярослав Павлович

студент, Пензенский государственный университет (Россия, г. Пенза, ул. Красная, 40)

E-mail: yaroslav.schegolikhin@yandex.ru

Shchegolikhin Yaroslav Pavlovich

Student, Penza State University (40 Krasnaya street, Penza, Russia)

Нешко Дарья Олеговна

студент, Пензенский государственный университет (Россия, г. Пенза, ул. Красная, 40)

E-mail: daria-uno@yandex.ru

Neshko Dar'ya Olegovna

Student, Penza State University (40 Krasnaya street, Penza, Russia)

Семенкин Максим Викторович

генеральный директор, ООО “КодИнсайд” (Россия, г. Пенза, ул. Суворова, 64Б)

E-mail: maxim.semekin@codeinside.ru

Semenkin Maksim Viktorovich

Director general, “CodeInside” LLC (64b Suvorova street, Penza, Russia)

Образец цитирования:

Митрохин, М. А. Анализ эффективности решений открытой библиотеки OpenCV в задаче распознавания автомобильных номеров / М. А. Митрохин, Я. П. Щеголихин, Д. О. Нешко, М. В. Семенкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2019. – № 2 (50). – С. 39–46. – DOI 10.21685/2072-3059-2019-2-4.