

УДК 004.93

РАСПОЗНАВАНИЕ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ БАЗОВЫХ СРЕДСТВ ЯЗЫКА PYTHON И БИБЛИОТЕКИ OPENCV

Бурмистров А.В., Ильичев В.Ю.

*Калужский филиал ФГОУ ВО «Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)», Калуга, e-mail: patrol8@yandex.ru*

Распознавание статических и динамических изображений является операцией, необходимой для развития современных технологий во многих отраслях науки и техники: в мехатронике и робототехнике, в медицине, в сфере обеспечения безопасности людей, для выявления нарушений правопорядка и др. При этом для языка программирования Python имеется библиотека компьютерного зрения OpenCV, позволяющая успешно осуществлять как процедуры распознавания образов, так и применять многие другие методы обработки изображений. При этом программы, написанные на Python, отличаются наглядностью, а также простотой понимания и корректировки, совершенствования. Поэтому целью данной работы является разработка программных продуктов, позволяющих продемонстрировать основные функциональные возможности модуля OpenCV по распознаванию образов различного заданного типа путём применения каскадного алгоритма Хаара, а также некоторые дополнительные функции. В ходе исследования разработаны две программы: для определения контуров объектов и для выделения определённых классов объектов. Приведены примеры применения программ для указания кромок деталей, а также для распознавания лиц, глаз, морд кошек и выделения их на изображении. Даны рекомендации по дальнейшему совершенствованию разработанных алгоритмов и по расширению сфер применения созданных методик распознавания образов.

Ключевые слова: робототехника, распознавание образов, определение кромок деталей, каскадный алгоритм Хаара, язык Python, модуль OpenCV

OBJECT RECOGNITION IN IMAGES USING BASIC PYTHON LANGUAGE TOOLS AND OPENCV LIBRARY

Burmistrov A.V., Ilichev V.Y.

Kaluga Branch of Bauman Moscow State Technical University, Kaluga, e-mail: patrol8@yandex.ru

Recognition of static and dynamic images is an operation necessary for the development of modern technologies in many branches of science and technology: in mechatronics and robotics, in medicine, in the field of ensuring the safety of people, to detect violations of law and order, etc. At the same time, for the Python programming language, there is a library of computer vision OpenCV, which allows you to successfully perform both pattern recognition procedures and use many other image processing methods. At the same time, programs written in Python are distinguished by clarity, as well as ease of understanding and adjustment, improvement. Therefore, the purpose of this work is to develop software products that allow you to demonstrate the main functionality of the OpenCV module for recognizing images of various given type by using the cascade Haar algorithm, as well as some additional functions. During the study, two programs were developed – to determine the outlines of objects and to highlight certain classes of objects. Examples are given of using programs to indicate the edges of parts, as well as to recognize faces, eyes, snouts of cats and highlight them in the image. Recommendations are given for further improvement of the developed algorithms and for expansion of the scope of application of the created pattern recognition techniques.

Keywords: robotics, pattern recognition, detail edge determination, Haar cascade algorithm, Python language, OpenCV module

Целью разработки технологий компьютерного зрения является имитация, воспроизведение в приложении к электронно-механическим устройствам человеческих зрительно-когнитивных способностей для реализации следующих задач [1]:

1) выделение отдельных важных элементов анализируемого изображения (распознавание);

2) в случае анализа динамичных сцен – предсказание траектории движения некоторых объектов.

В настоящее время эти задачи всё более активно решаются в таких отраслях знаний, как робототехника (например,

для классификации объектов или объезда препятствий) [2], медицина (для обнаружения опухолей при магниторезонансной томографии, ультразвуковом и рентгеновском исследованиях тканей, при реконструкции органов), информационная и техносферная безопасность (для обработки биометрических данных, распознавания опасных перемещений), мехатроника (для автоматизированной сборки промышленных изделий, дефектоскопии, сортировки деталей).

Естественно, данные задачи могут быть эффективно решены только с применением современных компьютерных технологий и алгоритмов. Пожалуй, наиболее совер-

шенные и при этом наглядно реализованные способы компьютерного зрения реализованы в среде программирования Python [3]. Для Python на данный момент существует специальный модуль компьютерного зрения OpenCV [4], включающий в себя самые совершенные математические и кибернетические приёмы.

Операции, которые можно осуществить с помощью OpenCV, являются очень разнообразными и разноплановыми. К ним относятся, например, следующие действия [5]:

1) считывание статических и динамических сцен из файлов многочисленных существующих форматов, а также отображение изображений во всплывающем динамическом окне и их запись почти во всех существующих форматах;

2) определение и изменение свойств сцен: размеров, типа цветовой гаммы, осуществление таких операций, как вращение, деформация, морфинг, а также других способов модификации как всего изображения, так и отдельных его элементов;

3) реализация описанных выше функций компьютерного зрения с помощью таких операций, как изменение цветового пространства, поиск краевых линий и точек объектов сцены, обнаружение определённых объектов с использованием образцовых моделей изображений, например таких характерных, как лица или глаза.

Таким образом, библиотека OpenCV позволяет исследователю производить огромное количество всевозможных операций, помогающих реализовать компьютерное зрение. Целью данного исследования является разработка программных продуктов для демонстрации и практического применения описанных выше базовых возможностей библиотеки OpenCV.

Материалы и методы исследования

В настоящее время существуют версии библиотеки OpenCV для языков C++ и Python. Язык Python выбран для разработки программ в данной работе по причине полностью свободного его распространения (и открытости исходных кодов абсолютного большинства библиотек), высокой скорости исполнения программного кода, совершенства применяемых алгоритмов компьютерного зрения и их универсальности при решении огромного количества разнообразных задач [6].

Рассмотрим последовательность расположения блоков созданного программного кода для обработки образца любой подаваемой на вход программы сцены-изображения:

1. Загрузка двух библиотек – функций взаимодействия с операционной системой

Os, позволяющих, например, находить файлы в любой папке на жёстком диске, а также функций компьютерного зрения (OpenCV – краткое название cv2).

2. Чтение файла в графическом формате jpeg, вывод его в окно на экране и перезапись в ту же папку с тем же именем, но уже в другом формате – png.

3. Преобразование палитры цвета изображения (для примера производится преобразование стандартной цветовой палитры RGB в оттенки серого).

4. На данном этапе выполняется алгоритм распознавания краёв изображения. Результат этого преобразования также записывается в графический файл с заданным разрешением. Обе эти операции объединены в одну команду.

Второй программный код предназначен для выполнения более «продвинутой» операции компьютерного зрения – определение лица на изображении. Он также содержит загрузку модуля cv2 и анализируемого изображения. В отличие от первой программы используется специальная функция каскадного алгоритма Хаара [7] HaarCascadeClassifier, в качестве параметров которой используется обрабатываемое изображение и файл xml. Этот файл содержит данные, характерные именно для моделирования искомого объекта и используемые в алгоритме Хаара.

Например, для обнаружения лица, расположенного анфас, из директории установки модуля OpenCV (cv2) необходимо выбрать файл haarcascade_frontalface_default.xml. В разработанной программе применяется особая разновидность алгоритма Хаара, позволяющая выделить на изображении лицо в нужном масштабе путём использования функции detectMultiScale. В качестве аргументов функции используются: ссылка на обрабатываемое изображение, фактор увеличения – указывает, во сколько раз увеличивается изображение во время каждого этапа обработки (чем меньше это значение, тем больше шанс совпадения искомого объекта с моделью из файла xml, т.е. больше вероятность его обнаружения), минимальное количество соседей – чем больше это значение, тем меньше вероятность обнаружения объекта, но выше качественное совпадение объекта с моделью из xml (по умолчанию принимается значение, равное 5). Также в функции detectMultiScale возможно использование дополнительных параметров, влияющих на способ применения каскадного алгоритма Хаара [8].

Затем вокруг области с обнаруженной деталью изображения (лицом) рисуется прямоугольник.

Каскадный алгоритм Хаара может быть использован для поиска любых обладающих характерными особенностями деталей изображения – животных, дорожных знаков, лиц и глаз людей и т.д. В каждом случае на вход функции CascadeClassifier необходимо «подгружать» файл xml, описывающий особенности искомой детали изображения.

Результаты исследования и их обсуждение

Рассмотрим результаты применения разработанных программных кодов для реализации технологий компьютерного зрения. В качестве изображения для обнаружения контуров технических объектов на комбинированной сцене (наиболее характерный случай при обработке изображения роботами) используется рис. 1, представляющий собой фото зубчатых колёс различного типа [9].



Рис. 1. Изображение деталей для обработки

Представленное изображение уже преобразовано программой в серую цветовую гамму. Результат работы первого описанного кода на Python (для определения граней объектов) представлен на рис. 2.

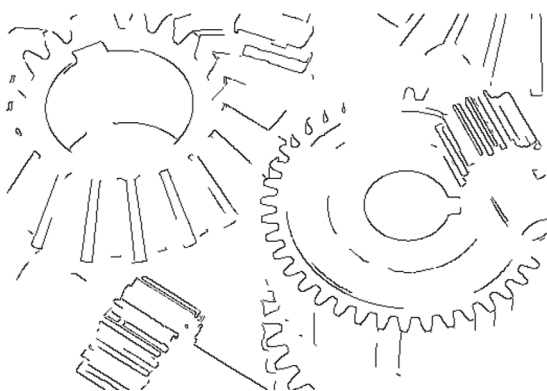


Рис. 2. Обработанное изображение деталей с обнаруженными их контурами

Из рис. 2 можно сделать вывод, что в результате компьютерной обработки сложной механистической сцены грани деталей определены достаточно отчётливо. Для получения ещё более качественных результатов рекомендуется производить дополнительную предварительную обработку изображения: повышать его резкость и контрастность, нормализовать по яркости [10]. Рекомендуется произвести отдельное исследование по изучению влияния перечисленных параметров на качество распознавания контуров.

На рис. 3 приведено изображение для демонстрации работы второго созданного программного кода с параметрами по умолчанию, предназначенного для обнаружения области, содержащей лицо (эта задача ставится на практике очень часто – например, большое количество случаев обработки графических потоков с камер наблюдения направлено именно на такое распознавание).

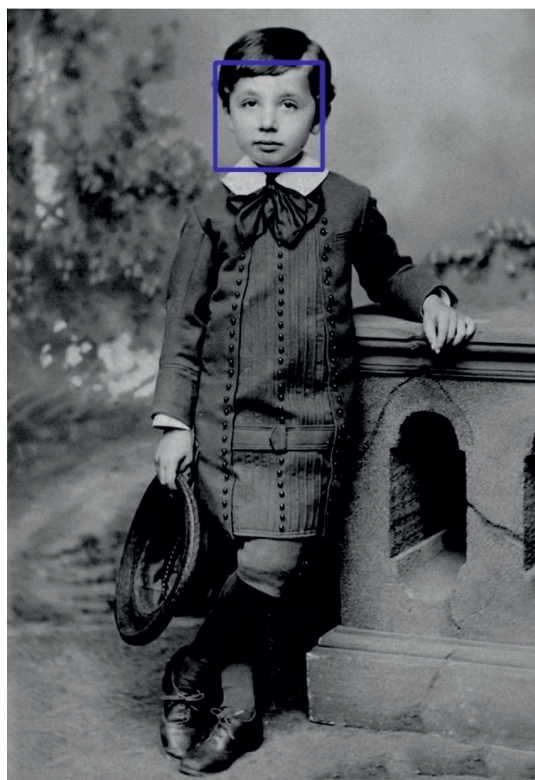


Рис. 3. Результаты распознавания лица на изображении

Как видно из рисунка, распознавание успешно произведено даже для изображения не очень высокого качества, содержащего большое количество посторонних деталей. Лицо человека выделено из всей представленной сцены очень точно.

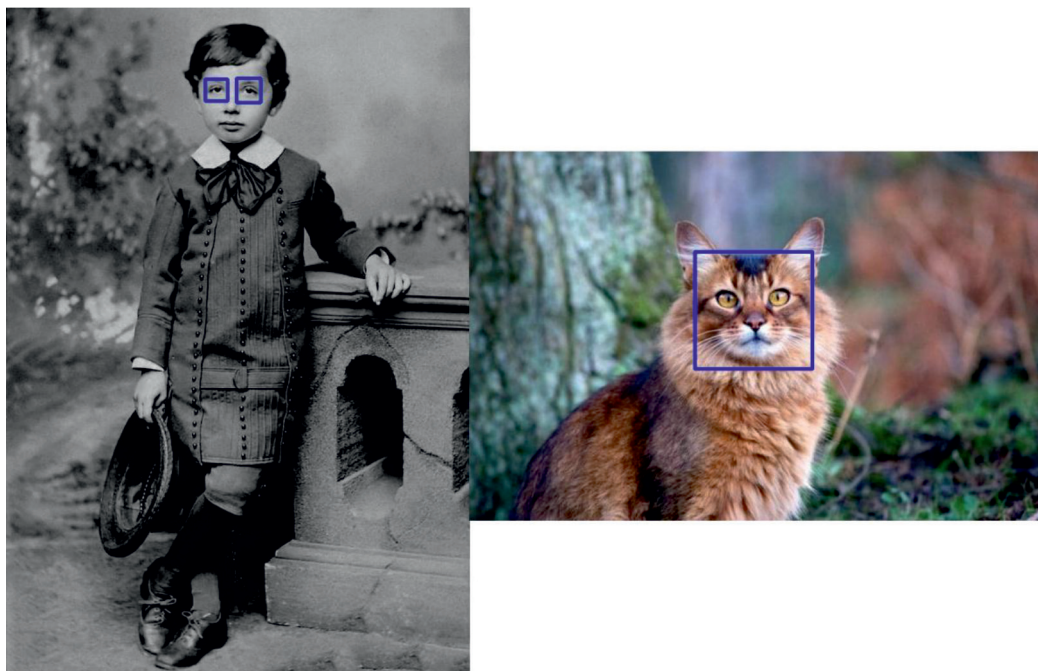


Рис. 4. Результаты распознавания человеческих глаз и морды кошки

Приведём далее результаты распознавания прочих объектов при помощи каскадного алгоритма Хаара. В директории установки модуля OpenCV для этого имеется множество соответствующих файлов xml. Также в сети Интернет можно найти множество таких файлов для выделения на изображении совершенно разных объектов. На рис. 4 представлены результаты распознавания глаз на приведённом выше изображении (использован файл из папки установки `haarcascade_eye.xml`), а также распознавание морды кошки на другом изображении (с применением соответствующего файла модели `haarcascade_frontalcatface_extended.xml`).

Заключение

По представленным результатам работы алгоритмов компьютерного зрения можно сделать следующие выводы:

1. Функции модуля OpenCV позволяют распознавать контуры деталей (например, для их обработки в промышленных условиях) или других объектов (к примеру, для поражения целей при ведении военных действий) [11].

2. Применение программных кодов с использованием функций OpenCV отлично подходит для выделения на анализируемых сценах заданных пользователем деталей (в приведённых случаях – лица,

глаза людей, морды кошек). Для изменения типа распознаваемых элементов изображения в разработанном программном коде достаточно лишь изменить ссылку на файл формата xml, содержащий существенные свойства типа искомых объектов для их математического описания.

3. Работа разработанных программ отличается быстротой исполнения и высокой точностью результатов даже в случае обработки сложных изображений с высоким разрешением.

4. Язык программирования Python позволяет использовать команды библиотеки OpenCV вместе с любыми функциями других многочисленных библиотек.

В описанном исследовании рассмотрено распознавание объектов только на статических, неподвижных изображениях. Однако библиотека OpenCV позволяет работать также с динамическими изображениями (видеопотоками).

Среда программирования Python, так же как и абсолютное большинство подключаемых к ней библиотек, является свободно распространяемой, и более того, существует огромное множество сообществ и форумов в сети Интернет, а также учебных пособий, позволяющих любому желающему в совершенстве освоить данный язык и функции его библиотек [12], успешно решать возможные трудности, возника-

ющие при их применении. В связи с этим технологиями распознавания образов могут овладеть не только опытные специалисты, но и начинающие пользователи, студенты вузов. Поэтому развитие тематики описанного исследования должно происходить всё более быстрыми темпами.

Список литературы

1. Иванов Ю.А. Технологии компьютерного зрения // Железнодорожный транспорт. 2012. № 12. С. 49.
2. Силонин В.А., Смычко М.В. Использование компьютерного зрения в автоматизированной робототехнике для охраны здоровья // Молодой ученый. 2021. № 22 (364). С. 121–123.
3. Ильичев В.Ю. Разработка программы для исследования аттрактора Лоренца и ее использование // Сложные системы. 2021. № 1 (38). С. 56–63.
4. Магамедова Д.М. OpenCV – инструмент компьютерного зрения // Тенденции развития науки и образования. 2020. № 63 3. С. 42–48.
5. Ильичев В.Ю. Использование рекурсивных функций для создания фрактальной графики средствами языка Python // Системный администратор. 2021. № 3 (220). С. 92–95.
6. Ilchev V.Y. Creation of software for research of Rössler attractor // International Journal of Humanities and Natural Sciences. 2021. № 5–1 (56). С. 31–35.
7. Белых Е.А. Обучение каскадов Хаара // Вестник Сыктывкарского университета. Серия 1: Математика. Механика. Информатика. 2017. № 1 (22). С. 42–54.
8. Object detection: face detection using Haar cascade classifiers. [Электронный ресурс] URL: https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Object_Detection_Face_Detection_Haar_Cascade_Classifiers.php (дата обращения: 08.08.2021).
9. Nasonov D., Raevsky V., Ilchev V., Volkhonskaya A. Methodology of error measurements reduction while determining integrated errors of satellite nodes of planetary gears. Vibroengineering PROCEDIA. 2021. Vol. 38. P. 90–94. DOI: 10.21595/vp.2021.22054.
10. Егоров Д.А. Методы автоматической цветовой и яркостной коррекции сигналов в системах обработки цифрового изображения // X Международная отраслевая научно-техническая конференция: сборник трудов. 2016. С. 93.
11. Краснов А.М. Управление поражением цели в комплексе авиационного вооружения со случайным изменением структуры // Труды МАИ. 2011. № 49. С. 10.
12. Яковлева Т.А., Конради Т.А. Робототехника в процессе обучения // Инновационные технологии в науке и образовании. 2015. № 3 (3). С. 143–145.