

УДК 004.93'12

Малиевский Ян Генрихович

*студент 3 курса факультета математики, информационных технологий и техники
Приамурский государственный университет им.Шолом-Алейхема Биробиджан, Россия*

Баженов Руслан Иванович

*к.п.н., доцент, зав.кафедрой информатики и вычислительной техники факультет математики,
информационных технологий и техники Приамурский государственный университет им.Шолом-
Алейхема Биробиджан, Россия*

Разработка приложения на Java по выделению лица на фотографии

Аннотация: В статье рассматривается реализация на языке программирования Java алгоритма выделения лица на фотографии. Представлен исходный код разработанного приложения. Для иллюстрации работоспособности приложения, взяты три фотографии различных людей.

Ключевые слова: Java, OpenCV, распознавание, лицо

Одним из важнейших векторов в развитии искусственного интеллекта является распознавание образов, которое уже не первое десятилетие продолжает использовать все вычислительные мощности, предоставляемые ему человечеством. Определение лиц –развитая подветвь распознавания образов. Системы распознавания лиц все чаще становятся нужны в мире, будь то электронные замки, которые пропускают только определенных людей или пароли на компьютерах. Определение людей на фотографиях в социальных сетях, да что там, любой современный фотоаппарат или веб- камера имеет в своем функционале какой-либо алгоритм для этого.

Для того чтобы работать с алгоритмом распознавания лиц вначале требуется составить базу каскадов, дабы избежать ошибок. В каскадах хранятся образцы не только лиц, но и объектов, принимаемых за них. В каскадах описаны цветовые пятна и их перекрещивания таким образом, чтобы программа поняла нужно ему это или нет.

Признаки, применяемые алгоритмом, применяют суммирование пикселей из прямоугольных зон. На рис.1 показано четыре разнообразных признака. Размер каждого признака подсчитывается как сумма пикселей в белых прямоугольниках, из которой отнимается сумма пикселей в черных зонах. При хранении изображения в интегральном формате, проверка прямоугольного признака на определенной позиции проводится за определенное количество времени, именно здесь проявляется их преимущество перед более точными вариантами. Каждая прямоугольная зона в применяемых признаках всегда смежна с иным прямоугольником, поэтому расчет признака с двумя прямоугольниками заключается в шести обращениях в интегральный массив, для признака с тремя прямоугольниками – в восьми, с четырьмя прямоугольниками – в девяти.

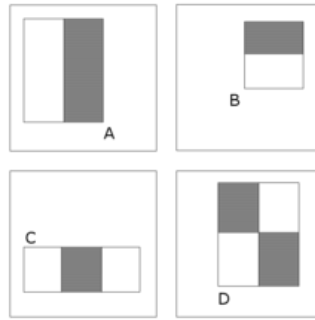


Рис.1 - Признаки

Р.Гонсалес, Р.Вудс описали этот и другие методы распознавания образов [6]. Создатели метода П.Виола и М.Дж.Джонес описали распознавание образов в режиме реального времени [14]. Л. М.Местецкий рассмотрел методы распознавания лиц с точки зрения математики [10]. Р.Sinha, В.Balas, Y.Ostrovsky и R.Russel исследовали распознавание образов мозгом [13]. Козлов П.В., Липин Ю.Н., Южаков А.А. описали алгоритм распознавания лица человека [8]. Достоинства и недостатки методов распознавания человека по характеристикам лица описывают А.Б.Щербань и К.С.Тихонова [11]. Концепцию алгоритма распознавания и анализа лица человека исследуют Д.В.Бутенко, А.В.Перепелицын [5]. Я.Ю.Дорогий описывает построение офтальмогеометрического классификатора для задачи распознавания человека по лицу [7]. А.А.Макаренко, В.Т.Калайда рассмотрели применение нейросетевой технологии для систем распознавания лица человека на групповом изображении [9].

OpenCV (англ. *Open Source Computer Vision Library*, библиотека компьютерного зрения с открытым исходным кодом) — библиотека алгоритмов компьютерного зрения, обрабатывания изображений и численных алгоритмов всеобщего назначения с открытым кодом. Создана на C/C++, также ведется разработка для Python, Java, Ruby, Matlab, Lua и прочих языков. OpenCV — это комплект алгоритмов и библиотек для работы с компьютерным зрением. Поиск объектов на изображениях, распознавание символов и все такое прочее. Г.Брадски и А.Каеблер описали

основные функции и методы OpenCV [12]. JavaCV является оболочкой для OpenCV для использования OpenCV в Java.

Алгоритм реализуемой программы:

1. Открываем изображение или берем его с камеры методами JavaCV.
2. Программа находит на фотографии лица и вставляет рамки вокруг

них.

Создается переменная класса изображения, используемого в JavaCV и окно, в которое будет выводиться изображение (рис 2).

```
IplImage originalImage;
CanvasFrame frame = new CanvasFrame("Face Detection");
frame.setDefaultCloseOperation(CanvasFrame.EXIT_ON_CLOSE);
```

Рис. 2. Создание переменной изображения и окна

Окно создано, теперь в переменную нужно «поместить» изображение.

Сначала код, где берется изображение с камеры (рис 3).

```
OpenCVFrameGrabber grabber = new OpenCVFrameGrabber(0);
grabber.start();
originalImage = grabber.grab();
if (originalImage == null) return;
```

Рис. 3. Получение изображения с камеры

Здесь переменная *grabber* – объект с которого будет происходить захват. 0 — это индекс устройства, он может быть больше нуля, если камер несколько. Значение -1 несет смысл «любая доступная камера». Метод *start()* запускает камеру, а метод *grab()* передает текущий кадр с камеры в переменную.

Последняя строка проверяет не пуста ли переменная *originalImage* и, если это так повторяет предыдущую строку. Загрузить изображение из файла можно, с помощью метода *cvLoadImage* (рис. 4).

```
originalImage = cvLoadImage(grabborfile);
```

Рис. 4. Получение изображения из файла

RGB-изображение, загруженное через *cvLoadImage* для поиска требуется преобразовать изображение в градации серого. Сначала создается пустое изображение с размерами загруженного, а затем копируется и преобразуется (рис. 5).

```
IplImage grayImage = IplImage.create(originalImage.width(),
                                     originalImage.height(), IPL_DEPTH_8U, 1);
cvCvtColor(originalImage, grayImage, CV_BGR2GRAY);
CvMemStorage storage = CvMemStorage.create();
```

Рис. 5. Преобразование изображения в градации серого

В этом моменте написания кода, происходит главное – на основе каскада *CASCADE_FILE* выводится ряд найденных лиц.

Рассмотрим константу *CASCADE_FILE*. Для программиста это будет файл *haarcascade_frontalface_alt.xml*, а для компьютера – критерий для того, чтобы находить лица на изображении. Каскады для лиц, носов, ртов, автомобильных номеров и т.д. располагаются в папке с OpenCV. А также возможно организовать их создание с помощью программ, располагающихся в папке *bin* директории, куда был установлен OpenCV. Таким образом, можно настроить каскад под определенное лицо, но для настройки понадобится несколько суток и выборка, состоящая из двух – четырех тысяч изображений, ведь чем больше фотографий в каскаде, тем меньше вероятность ошибки (рис. 6).

```
String CASCADE_FILE = "C:\\opencv\\data\\haarcascades\\haarcascade_frontalface_alt.xml";
CvHaarClassifierCascade cascade = new CvHaarClassifierCascade(cvLoad(CASCADE_FILE));
CvSeq faces = cvHaarDetectObjects(grayImage, cascade, storage, 1.2, 3,3);
```

Рис. 6. Подключение каскада

Теперь нужно выделить лица на основном изображении (*originalImage*) (рис. 7).

```
for (int i = 0; i < faces.total(); i++) {
    CvRect r = new CvRect(cvGetSeqElem(faces, i));
    cvRectangle(originalImage, cvPoint(r.x(), r.y()), cvPoint(r.x() + r.width(),
                                                              r.y() + r.height()), CvScalar.YELLOW, 1, CV_AA, 0);
}
```

Рис. 7. Выделение лиц на изображении

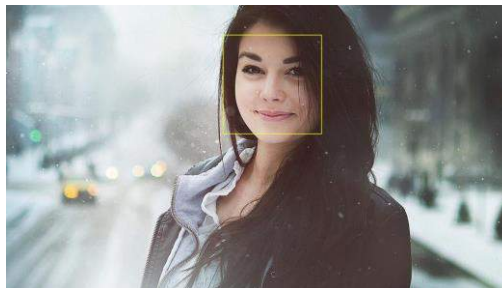


Рис. 8 Фото с точным распознаванием одного лица

В ходе решения задачи был изучен алгоритм распознавания образов Виолы-Джонса. На основе этого метода на Java было создано приложение, с помощью которого лица на фотографии выделяются рамкой. Данные исследования можно использовать в преподавании [1-4].

Список литературы:

1. Баженов Р.И. О применении балльно-рейтинговой системы для оценивания курсовых работ по дисциплине «Интеллектуальные системы и технологии» // Приволжский научный вестник. 2014. № 5 (33). С. 135-138.
2. Баженов Р.И. Об организации деловых игр в курсе «Управление проектами информационных систем» // Научный аспект. 2014. Т. 1. № 1. С. 101-102.

3. Баженов Р.И. Информационная безопасность и защита информации: практикум. Биробиджан: Изд-во ГОУВПО «ДВГСГА», 2011. 140 с.
4. Баженов Р.И. Об организации научно-исследовательской практики магистрантов направления «Информационные системы и технологии» // Современные научные исследования и инновации. 2014. № 9-2 (41). С. 62-69.
5. Бутенко Д.В., Перепелицын А.В. Концепция алгоритма распознавания и анализа лица человека // Программные продукты и системы. 2011. № 1. С. 43.
6. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.
7. Дорогий Я.Ю. Построение офтальмогеометрического классификатора для задачи распознавания человека по лицу // Информационные технологии. Радиоэлектроника. Телекоммуникации. 2012. Т. 2. № 2. С. 24-33.
8. Козлов П.В., Липин Ю.Н., Южаков А.А. Алгоритм распознавания лица человека // Вопросы защиты информации. 2011. № 1. С. 52- 57.
9. Макаренко А.А., Калайда В.Т. Применение нейросетевой технологии для систем распознавания лица человека на групповом изображении // Информационные технологии. 2007. № 9. С. 27-32.
10. Местецкий Л.М. Математические методы распознавания образов. М.: МГУ, 2002. С. 42 -44.
11. Щербань А.Б, Тихонова К.С. Достоинства и недостатки методов распознавания человека по характеристикам лица // Современные информационные технологии. 2010. № 11. С. 196-206.
12. Bradski G., Kaebler A., Learning OpenCV. Published by O'Reilly Media, 2008. 512 с.
13. Sinha P., Balas B., Ostrovsky Y., Russel R. Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About, Massachusetts Institute of Technology [Электронный ресурс]. Режим доступа: http://web.mit.edu/bcs/sinha/papers/19results_sinha_etal.pdf (дата обращения: 14.12.2014)
14. Viola P., Jones M.J. Robust real-time face detection // International Journal of Computer Vision. 2004. Т. 57. №2. С. 142 – 154.

© Малиевский Я.Г., Баженов Р.И., 2015

Дата публикации: 29.01.2015