



Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского

Лабораторная работа

Детектирование объектов с использованием библиотеки OpenCV

Молодежная научная школа
«Высокопроизводительные вычисления, оптимизация и приложения»
Трек «Робототехника и компьютерное зрение»

Кустикова В.Д.,
к.т.н., ст.преп. каф. МОСТ ИИТММ

Цели

- ❑ **Цель** – решить одну из классических задач компьютерного зрения – детектирование объектов – с использованием средств библиотеки OpenCV. В качестве целевых объектов для детектирования рассматриваются логотипы ННГУ, OpenCV и Intel.



Общая схема решения задачи детектирования (1)

Обучение детектора

- ❑ Подготовка тренировочных данных - множества положительных (содержащих объект) и отрицательных (не содержащих объект) примеров изображений.
- ❑ Извлечение признаков из каждого изображения тренировочного набора.
- ❑ Обучение детектора на полученном наборе признаков.
- ❑ Сохранение параметров обученной модели.



Общая схема решения задачи детектирования (2)

Тестирование детектора

- ❑ Получение изображения, на котором необходимо определить положения объектов.
- ❑ Извлечение признаков из полученного изображения. На данном этапе используется тот же алгоритм, что и при обучении детектора.
- ❑ Передача построенных признаков детектору, который принимает решение о наличии/отсутствии объекта и его местоположении в системе координат изображения.



Задачи

- ❑ Разработать приложение, осуществляющее детектирование объектов определенного класса (один из предложенных классов логотипов) на одиночном изображении и видео с помощью каскадного классификатора, т.е. обеспечить тестирование детектора.
- ❑ Обучить детектор логотипов выбранного класса на основе HAAR- и LBP-признаков.
- ❑ Применить полученные детекторы к тестовому видео. Визуально оценить качество детекторов.



Шаблон

- ❑ Ссылка на открытый репозиторий:

<https://github.com/valentina-kustikova/hpcschoool-2016-practice>



Изображение в библиотеке OpenCV

`cv::Mat` – тип данных, используемый для хранения изображений. Объявлен в заголовочном файле `opencv2/core/core.hpp`.

Функции:

- ❑ **`imread`** – загрузка изображения из файла.
- ❑ **`namedWindow`, `imshow`, `waitKey`** – создание окна для отображения, отрисовка изображения, ожидание нажатия какой-либо клавиши. Объявлены в заголовочном файле `opencv2/highgui/highgui.hpp`.



Видео в библиотеке OpenCV

`cv::VideoCapture` – класс для организации работы с видео. Объявлен в заголовочном файле `opencv2/core/core.hpp`.

Методы и операции:

- ❑ **`open`** – открытие видеопотока.
- ❑ **`isOpened`** – проверка, был ли открыт видеопотока.
- ❑ **`>>`** – потоковый ввод (получение следующего кадра видеопотока).



Геометрические примитивы в библиотеке OpenCV

`cv::Rect` – тип данных, используемый для представления прямоугольника. Объявлен в заголовочном файле `opencv2/core/core.hpp`.

Функции:

- ❑ `rectangle` – функция отображения прямоугольника на изображении. Объявлена в заголовочном файле `opencv2/imgproc/imgproc.hpp`.



Каскадный классификатор в библиотеке OpenCV

`cv::CascadeClassifier` – класс библиотеки OpenCV, используемый для представления сущности каскадного классификатора. Объявлен в заголовочном файле `opencv2/objdetect/objdetect.hpp`.

Методы:

- ❑ **`load`** – загрузка обученной модели.
- ❑ **`detectMultiScale`** – детектирование объектов разного масштаба на изображении с использованием обученной модели.



Аргументы командной строки в библиотеке OpenCV

cv::CommandLineParser – класс, используемый для разбора аргументов командной строки.

Методы:

- ❑ **has** – проверяет наличие аргумента по его названию.
- ❑ **get** – получает значение аргумента по его названию.
- ❑ **printMessage** – печатает сообщение, содержащее информацию о всех возможных аргументах.

