



Санкт-Петербургский
государственный университет

Санкт-Петербургский государственный университет



Санкт-Петербургский
государственный университет

Машинное зрение и OpenCvSharp

Введение в машинное зрение

- **Машинное (компьютерное) зрение** – способность компьютеров «видеть» и понимать изображения.
- Область искусственного интеллекта, переработка фото/видео для извлечения информации.
- Цель: автоматическое **распознавание** и **анализ** визуальной сцены (объектов, действий).

Примеры применения машинного зрения

- **Автономные автомобили:** распознавание дорог, знаков, пешеходов.
- **Распознавание лиц:** в телефонах, системах безопасности.
- **Медицинская диагностика:** анализ рентгеновских и МРТ снимков.

Задачи компьютерного зрения

- **Классификация изображений:** определение, что изображено (кошка, машина и т.д.).
- **Детекция объектов:** нахождение объектов на изображении и выделение их границ (например, лица, пешеходы).
- **Отслеживание (трекинг):** слежение за перемещением объекта на видео.
- **Сегментация:** разделение изображения на области (например, фон/объект).
- **Реконструкция сцен:** получение 3D-информации из 2D-изображений (опционально).

Основные понятия (цифровое изображение)

- **Пиксель:** минимальный элемент изображения; имеет цвет/яркость.
- **Разрешение:** размер изображения в пикселях (ширина x высота).
- **Цветные vs серые изображения:** RGB/BGR три канала vs один канал.
- **Матрица пикселей:** изображение = числовая матрица (или три матрицы для цвета).

Библиотека OpenCV – что это и зачем нужна?

- **OpenCV (Open Source Computer Vision Library)** – открытая библиотека компьютерного зрения.
- Написана на C/C++ (начало проекта – 2000 г.), кроссплатформенная.
- Содержит сотни алгоритмов обработки изображений: фильтры, детекторы, трекинг, и др.
- Стала стандартом для разработки CV-приложений (исследования и индустрия).

Использование OpenCV в C# – библиотека OpenCvSharp

- **OpenCvSharp** – это обёртка (wrapper) OpenCV для .NET (языки C#, VB и др.).
- Позволяет вызывать функции OpenCV из C# напрямую, не переключаясь на C++ или Python.
- Кроссплатформенная: работает на Windows, Linux, macOS (через .NET 6+).
- Простое подключение через **NuGet** (менеджер пакетов .NET).

Подключение OpenCvSharp в проект C#

- Установка через NuGet: пакет **OpenCvSharp4** + пакет **OpenCvSharp4.runtime.{OS}**.
- Для Windows можно сразу использовать **OpenCvSharp4.Windows** (all-in-one).
- После установки – подключить в коде: `using OpenCvSharp`
- Требование: установлен Visual C++ Redistributable (для Windows)
- Версии: на данный момент используется OpenCV 4.x.

Основные классы OpenCvSharp

- **Mat** – основной класс изображения (матрица пикселей).
- **Cv2** – статический класс, предоставляющий большинство функций (аналог функций cv:: в C++).
- **Point, Rect, Scalar** – вспомогательные структуры (точка, прямоугольник, цвет и т. д.).
- **Window** – класс для отображения изображений в окне (или используем Cv2.ImShow).

Пример: Чтение и вывод изображения

```
using OpenCvSharp;

Mat img = Cv2.ImRead("image.jpg", ImreadModes.Color);
if (img.Empty())
{
    Console.WriteLine("Не удалось открыть файл image.jpg");
    return;
}
Cv2.ImShow("Окно", img);
```

Пример: Цветное - градиенты серого

```
Mat gray = new Mat();  
Cv2.CvtColor(img, gray, ColorConversionCodes.BGR2GRAY);  
Cv2.ImShow("Серое изображение", gray);  
Cv2.ImWrite("gray_output.jpg", gray);
```

- После загрузки изображения как `img`, мы создаём пустой `Mat gray` и используем `Cv2.CvtColor` для конвертации цветного изображения `img` в градации серого (`gray`). Затем отображаем результат и сохраняем его в файл.

Пример: Применение фильтра размытия

```
Mat blurred = new Mat();  
Cv2.GaussianBlur(gray, blurred, new Size(7, 7), 0);  
Cv2.ImShow("Размытие 7x7", blurred);
```

- Исходное изображение (в оттенках серого) помещается в переменную gray. Применяем гауссово размытие с ядром 7x7. Отображаем результат в новом окне.

Пример: Поиск границ

```
Mat edges = new Mat();  
Cv2.Canny(blurred, edges, 50, 150);  
Cv2.ImShow("Границы (Canny)", edges);
```

- Берём размытую картинку и ищем на ней границы с помощью Cv2.Canny. Пороговые значения – 50 и 150. Результат – бинарное изображение с контурными линиями.

Пример: Детекция лица

```
// Загрузка классификатора (файл XML с признаками для лиц)
var faceCascade = new CascadeClassifier("haarcascade_frontalface_default.xml");
// Конвертируем изображение в градации серого для улучшения детекции
Mat frameGray = new Mat();
Cv2.CvtColor(img, frameGray, ColorConversionCodes.BGR2GRAY);
// Выполняем обнаружение лиц на изображении
Rect[] faces = faceCascade.DetectMultiScale(frameGray, 1.1, 4);
// Рисуем рамки вокруг обнаруженных лиц
foreach (var face in faces)
{
    Cv2.Rectangle(img, face, Scalar.Red, 2);
}
Cv2.ImShow("Обнаружение лица", img);
```

- Используем готовый классификатор (каскад Хаара) для поиска лиц. После загрузки классификатора из XML, применяем DetectMultiScale на изображении (в градациях серого). Результат – массив прямоугольников Rect с координатами лиц. Рисуем красные прямоугольники на оригинальном изображении и выводим результат.

Особенности работы с OpenCvSharp

- Синтаксис близок к C++ API OpenCV (легко портировать примеры из документации).
- Можно работать как с изображениями, так и с видео (класс VideoCapture для камеры/файла).
- GUI: Отображение изображений через Cv2.ImShow / класс Window. Возможна интеграция с WinForms/WPF (Mat -> Bitmap).
- Производительность: вычисления выполняются на C/C++ уровне (быстро), но учитывайте расходы на маршаллинг данных между .NET и нативной частью.

Полезные ссылки и материалы

1. **Официальный сайт OpenCV:** документация и примеры – opencv.org (англ.).
2. **Документация OpenCV (версии 4.x):** docs.opencv.org/4.x – описание всех функций, туториалы.
3. **GitHub OpenCvSharp:** github.com/shimat/opencvsharp – исходный код обёртки, инструкции по установке.
4. **Статья (Habr) «OpenCV на C# (OpenCvSharp)»:** практическое введение с кодом (на русском).
5. **Учебники и примеры:** поиск по “OpenCvSharp example” – много открытых проектов с кодом (GitHub, StackOverflow).
6. **Книга «Learning OpenCV» (Bradski, Kaehler)** – классическая книга по OpenCV (на англ., есть переводы) – помогает понять основы CV и как их применять с OpenCV.