

Семинар 1. Работа с изображениями

Описание

Первый семинар курса посвящён освоению базовых приёмов **загрузки, анализа и преобразования изображений** с помощью библиотек **OpenCV** и **NumPy**. Цель — закрепить фундаментальные понятия первой лекции: что такое изображение как дискретный сигнал, как интерпретировать пиксели как численные значения интенсивности, и как анализировать распределения яркости и цвета.

Задания выстроены последовательно: от чтения и визуализации до анализа гистограмм и частотных представлений. Выполняя их, студент должен осознать, что операции над изображением — это не “магия обработки”, а математика над массивами чисел.

Задания

1. Загрузка и первичный анализ изображений

Цель: освоить базовые операции чтения и визуализации изображений, познакомиться с их внутренним представлением в памяти.

Инструкции:

1. Загрузите три изображения в разных форматах: `.bmp`, `.png`, `.jpg`.
2. Выведите их размеры (`shape`), тип данных (`dtype`) и глубину каналов.
3. Выведите первые 5×5 элементов матрицы яркости для одного изображения.
4. Сравните размеры файлов и объясните различия между форматами.

Подсказка:

```
import cv2
img = cv2.imread("example.png", cv2.IMREAD_UNCHANGED)
print(img.shape, img.dtype)
```

2. Гистограмма яркости и линейное растяжение

Цель: научиться визуализировать распределение интенсивностей и управлять контрастом изображения.

Инструкции:

1. Преобразуйте цветное изображение в оттенки серого (`cv2.cvtColor`).
2. Постройте гистограмму яркости (256 бинов).

3. Примените линейное растяжение диапазона яркости (нормализацию) с помощью `cv2.normalize` или вручную по формуле:
$$I'(x, y) = \frac{I(x, y) - I_{\min}}{I_{\max} - I_{\min}} \times 255.$$
4. Повторно постройте гистограмму и визуально сравните результат.

Ожидаемый результат: увеличенный контраст и расширение гистограммы на весь диапазон 0–255.

3. Эквализация гистограммы и локальный контраст

Цель: изучить методы улучшения видимости деталей.

Инструкции:

1. Примените глобальную эквализацию гистограммы (`cv2.equalizeHist`) к тёмному изображению.
2. Сравните с адаптивной эквализацией CLAHE:

```
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
result = clahe.apply(gray)
```

3. Постройте гистограммы для каждого варианта (оригинал, equalized, CLAHE).
4. Сравните визуально и количественно (по стандартному отклонению интенсивности).

Ожидаемый результат: глобальная эквализация усиливает контраст во всём изображении, CLAHE — локально, не усиливая шум.

4. Цветовые каналы и цветовые пространства

Цель: познакомиться с преобразованием между цветовыми моделями и разделением каналов.

Инструкции:

1. Загрузите цветное изображение и преобразуйте его в пространства **HSV**, **Lab**, **YCbCr**.
2. Разделите каналы и визуализируйте каждый отдельно.
3. Постройте гистограмму компоненты яркости (**V**, **L***, **Y**) и сравните с гистограммой яркости в RGB.
4. Сделайте вывод: в каком пространстве лучше отделяются освещённость и цвет.

Подсказка:

```
hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv)
```

5. Форматы и потери при сжатии

Цель: изучить эффект компрессии JPEG и визуализировать потери данных.

Инструкции:

1. Сохраните изображение в JPEG с качеством 95, 60 и 30:

```
cv2.imwrite("img_60.jpg", img, [cv2.IMWRITE_JPEG_QUALITY, 60])
```

2. Загрузите обратно и вычислите абсолютную разницу с исходным:

```
diff = cv2.absdiff(img, restored)
```

3. Вычислите среднюю ошибку:
$$\text{MSE} = \frac{1}{MN} \sum_{x,y} (I(x,y) - I'(x,y))^2$$
4. Отобразите `diff` как изображение ошибок и сравните визуально влияние степени сжатия.

Ожидаемый результат: при сильной компрессии появляются блочные артефакты и размытые границы.

6. Частотное представление (дополнительное задание)

Цель: закрепить понимание частотных характеристик изображения.

Инструкции:

1. Выполните 2D-Фурье-преобразование (`np.fft.fft2`) и визуализируйте амплитудный спектр.
2. Сконструируйте маску низкочастотного фильтра (например, круг радиуса 40 пикселей).
3. Примените фильтр к спектру и выполните обратное преобразование (`ifft2`).
4. Сравните исходное, отфильтрованное и спектральные представления.

Ожидаемый результат: низкочастотная фильтрация сглаживает изображение, уменьшая резкость; визуально спектр становится более компактным.
