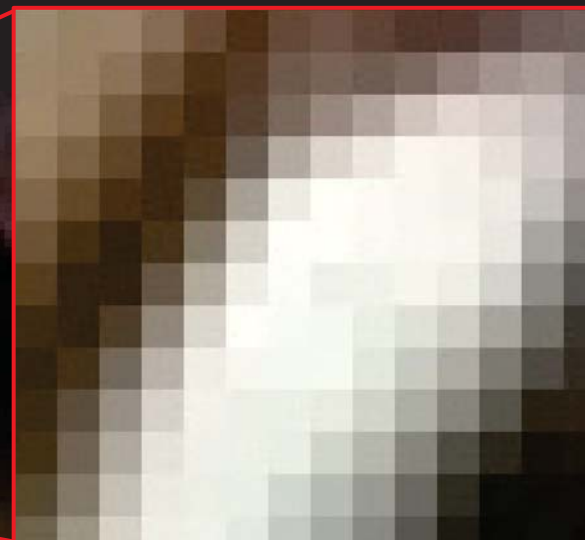
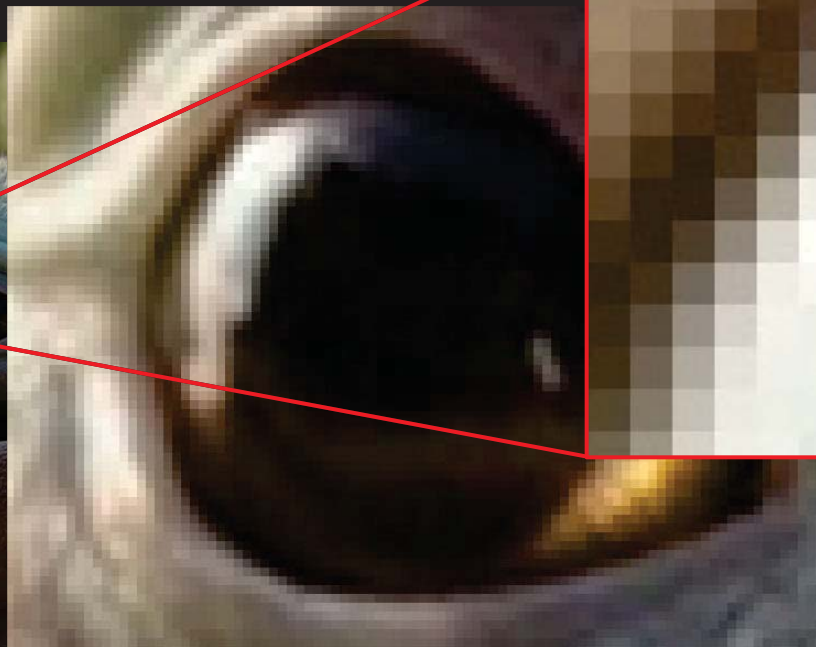


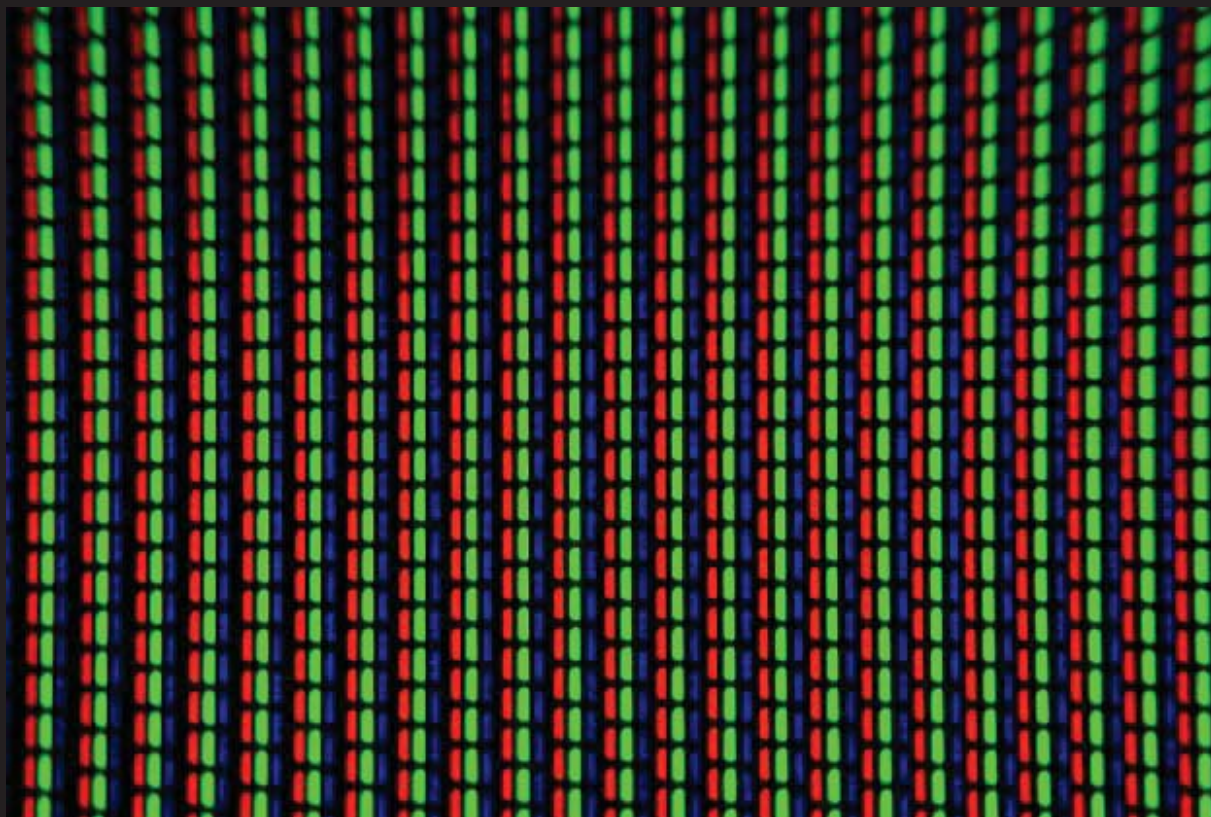
Обработка изображений

ЛКШ.2019.Зима.С'+

Представление изображений



Представление изображений



```
$ pip3 install pillow
```

Базовые действия

```
from PIL import Image
img = Image.open("image.png")
pixels = img.load()
for x in range(img.width):
    for y in range(img.height):
        r, g, b = pixels[x, y]
        pixels[x, y] = g, b, r
img.save('result.jpg')
```

Для цветных изображений пиксель — это кортеж (R, G, B)

Для изображений с прозрачностью — (R, G, B, alpha)

Как сдавать задачи

Со стандартного ввода вводится: в первой строке — **имя входного файла**, например `test.jpg`, во второй строке — **имя выходного файла**, например `result.jpg`, **дополнительные параметры** в следующих строках (если есть).

Программа должна преобразовать изображение из первого файла и записать результат во второй.

Код сдавать в ejudge, а полученные картиночки архивом кидать препам (называйте картинки номерами задач). Все решения проверяются вручную. Ждем!

Задание 1: Негатив



(R, G, B)



(255 - R, 255 - G, 255 - B)

Яркость пикселя



Просто: $(R + G + B) / 3$

Сложно: $a * R + b * G + c * B$

Задание 2: Изменение яркости

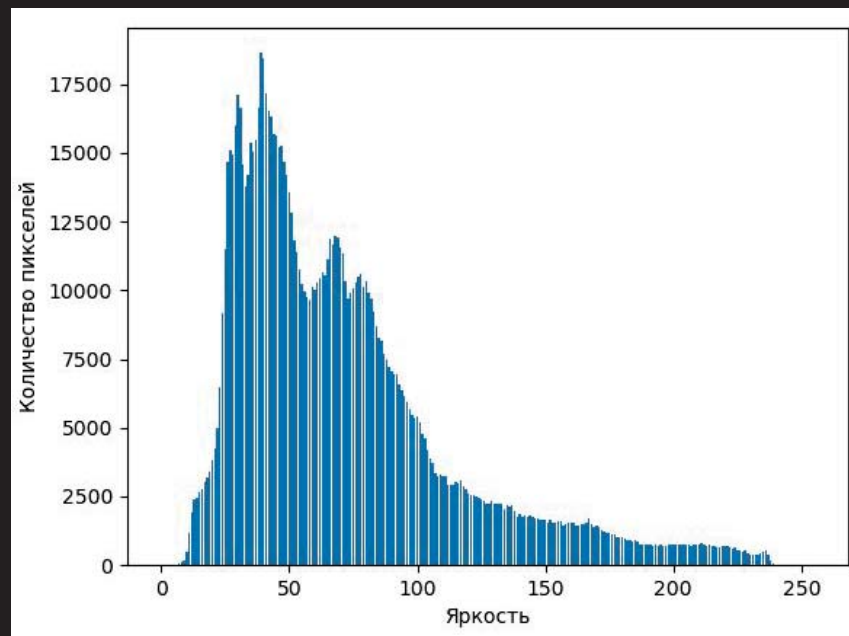


(R, G, B)

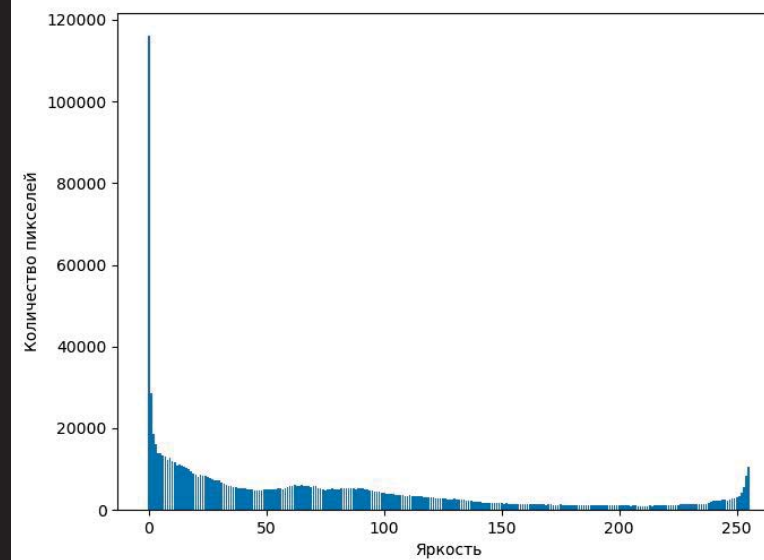


$(R + c, G + c, B + c)$
 c задается в третьей строке

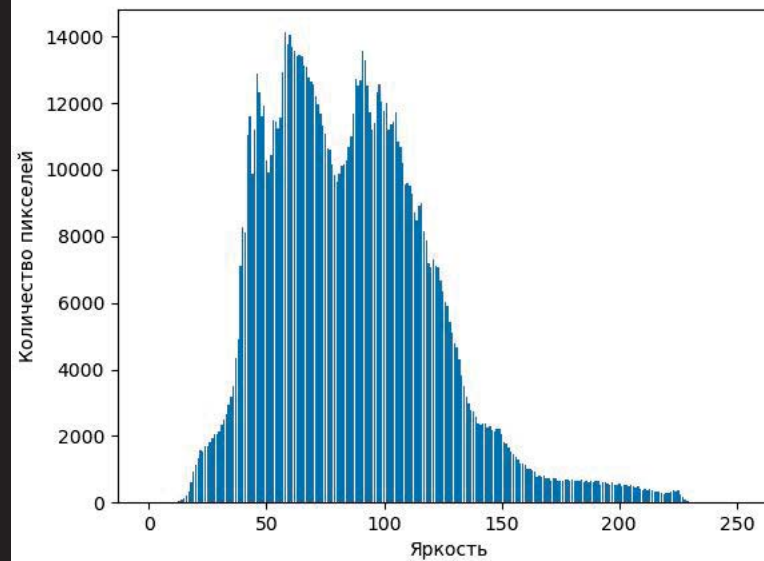
Контрастность



Контрастность



Контрастность



Задание 3: Изменение контрастности

Задача: изменить контрастность изображения с заданным коэффициентом, не меняя средней яркости

Параметры: c — коэффициент изменения контрастности

Идея: растянуть яркости относительно среднего значения

Как:

1. Посчитать среднюю яркость пикселей по всему изображению L_{avg}
2. Пересчитать новые значения по формулам:

$$R = L_{avg} + (R - L_{avg}) * c$$

$$G = L_{avg} + (G - L_{avg}) * c$$

$$B = L_{avg} + (B - L_{avg}) * c$$



Баланс белого



Задание 4: Серый мир

Задача: исправить баланс белого

Идея: сделать равными средние значения красной, синей и зелёной компонент

Как:

1. Посчитать средние значения компонент:

$$R_avg = \text{sum}(R(x, y)) / (W * H)$$

$$G_avg = \text{sum}(G(x, y)) / (W * H)$$

$$B_avg = \text{sum}(B(x, y)) / (W * H)$$

$$L_avg = (R_avg + G_avg + B_avg) / 3$$

2. Пересчитать новые значения:

$$R = R * L_avg / R_avg$$

$$G = G * L_avg / G_avg$$

$$B = B * L_avg / B_avg$$



Задание 5: Автоконтраст

Задача: увеличить контрастность изображения

Идея: в каждом из каналов изменить значения цвета так, чтобы самому тёмному соответствовало значение 0, а самому яркому — 255. Все остальные значения изменить равномерно

Как:

1. Найти минимальное и максимальное значения цвета для каждого из каналов
2. Пересчитать новые значения по формулам

$$R = (R - R_{\min}) / (R_{\max} - R_{\min}) * 255$$

$$G = (G - G_{\min}) / (G_{\max} - G_{\min}) * 255$$

$$B = (B - B_{\min}) / (B_{\max} - B_{\min}) * 255$$



Задание 6: Обрезание краев

Задача: обрезать **L** пикселей слева, **R** пикселей справа, **T** пикселей сверху и **B** пикселей снизу. Сохранить новое изображение меньшего размера в файл

Параметры: числа L, R, T, B через пробел

Как:

```
# Цветовая схема, размеры, изначальный цвет  
img = Image.new("RGB", (W, H), "white")
```

В качестве цветовой схемы можно передавать цветовую схему другого изображения. Её можно получить написав `img.mode`



Задания на выбор

Из следующих заданий можете выбирать и делать любые

Задание 7: Автоконтраст (5%)

Задача: увеличить контрастность изображения

Идея: в каждом из каналов изменить значения цвета так, чтобы 5% самых тёмных соответствовало значению 0, а 5% самых светлых — 255. Все остальные значения изменить равномерно

Как:

1. Для каждого канала и каждого значения цвета посчитать количество раз, которое он встречается в изображении
2. Найти максимальное среди 5% самых тёмных и минимальное среди 5% самых светлых значения цвета для каждого канала
3. Пересчитать новые значения по таким же формулам, как в задании про автоконтраст



Задание 8: Фильтр «Стекло»

Задача: в результирующем изображении каждый пиксель с координатами (x, y) является случайным пикселем из окрестности $(x-D \dots x+D, y-D \dots y+D)$ исходного изображения

Параметры: D — размер области

Как:

для выбора случайного смещения можно использовать функцию `randrange` из модуля `random`

```
from random import randrange
# delta - случайное целое число от -5 до 5
delta = randrange(-5, 6)
```



Задание 9: «256 оттенков серого»

Задача: сделать изображение черно-белым

Идея: для каждого пикселя присвоить компонентам R, G и B значение яркости этого пикселя



Задание 10: Фильтр «Сепия»

Задача: сделать фото устаревшим, черно-белым с коричневым оттенком

Параметры: k — степень коричневого

Идея: сделать фото черно-белым, добавим коричневого (красного и зелёного)

Как:

```
middle = (R + G + B) / 3
R = middle + 2 * k
G = middle + k
B = middle
```



Задание 11: Фильтр «Mayfair»

Задача: затемнить края, сделать центр более ярким

Идея: изменять яркость в зависимости от расстояния до центра изображения

Параметры: два числа — изменение яркости в центре и в самой удалённой точке



Задание 12: Бинаризация изображения

Задача: оставить в изображении только два цвета: черный и белый

Идея: все пиксели с яркостью меньше заданного порога сделать черными, больше порога — белыми

Параметры: порог яркости

Модификация*: определять порог автоматически как среднее значение яркости



Задание 13: Творите

Задача: вы — творцы своего фильтра!

Идея: нужно сотворить фильтр, который нам понравится

Как: возьмите и творите

Параметры: какие вытворите

