ЛАБОРАТОРНАЯ РАБОТА №2.
Работа с цветными изображениями в Python

ЦЕЛЬ РАБОТЫ – Знакомство с представлением изображений в цветовых пространствах RGB и YCbCr и приобретение опыта обработки цветных изображений в Python.

Содержание

1 Теоретические сведения

1.1 Цветовые системы

1.1.1 Цветовая система RGB

В цветовом пространстве RGB пикселы цветоного изображения представляются с помощью трех чисел, указывающих относительное соотношение красного (*red*), зеленого (*green*), синего (*blue*) цветов. Поскольку красная, зелёная и синяя составляющие цвета элемента изображения задаются, численным значением от 0 до 255, для запоминания каждого элемента изображения требуется 24 бита. Такое изображение теоретически может содержать до 16,8 миллиона различных цветов.

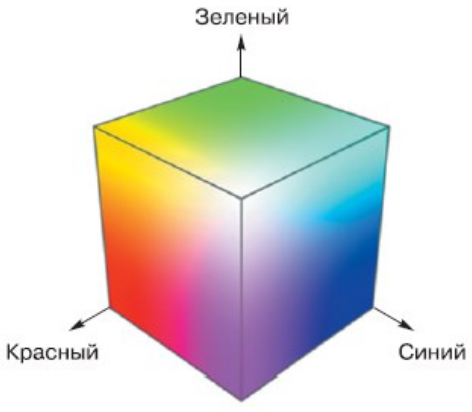

Рисунок 1 – Взаимосвязь цветов в цветовой модели RGB

Цветовую систему RGB можно представить в виде куба (рис. 1), для которого каждая его пространственная точка определяется координатами X , Y , Z. На главной диагонали куба расположены серые цвета.

В цветовом пространстве RGB все три цвета считаются одинаково важными, и они обычно сохраняются с одинаковым разрешением.

При практическом применении системы RGB встречаются две проблемы:

1) это зависимость от аппаратуры;

2) невозможность получать цвета путем аддитивного цветового синтеза.

1.1.2 Цветовая система YCbCr

В цветовом пространстве YCbCr (YUV) используется компонента яркости (Y) и две хроматические компоненты (Cb, Cr). Яркостная компонента вычисляется как взвешенное усреднение компонент R, G, B по формуле:

$$Y = k_r R + k_g G + k_b B, \tag{1.1}$$

где k_x – весовой множитель по компоненте $x \in \{r, g, b\}$.

По рекомендации стандарта ITU-T с идентификатором BT.601:

$$Y = 0{,}299 \cdot R + 0{,}587 \cdot G + 0{,}114 \cdot B, \tag{1.2}$$

$$Cb = 0{,}564 \cdot (B - Y), \tag{1.3}$$

$$Cr = 0{,}713 \cdot (R - Y), \tag{1.4}$$

Обратное преобразование определяется формулами

$$R = Y + 1{,}402Cr, \tag{1.5}$$

$$G = Y - 0{,}344Cb - 0{,}714Cr, \tag{1.6}$$

$$B = Y + 1{,}772Cb. \tag{1.7}$$

Преимущество пространства YCbCr по сравнению с RGB заключается в том, что компоненты Cb и Cr можно представить с меньшим разрешением, чем Y, так как глаз человека менее чувствителен к цвету предметов, чем к их яркости. Это позволяет сократить объем информации, требуемый для представления хроматических компонент, без заметного ухудшения качества передачи цветовых оттенков изображения.

Использование YCbCr позволяет передать информацию о яркости с полным разрешением, а для цветоразностных компонент произвести субдискретизацию, то есть выборку с уменьшением числа передаваемых элементов изображения, так как человеческий глаз менее чувствителен к перепадам цвета (рис. 2). Это повышает эффективность системы, позволяя уменьшить поток видеоданных.

Рассмотрим процесс субдискретизации (*subsampling*) более детально.

Схема субдискретизации обычно выражается в виде трех чисел $J{:}a{:}b$ (например, 4:2:2), которые описывают количество пикселей яркости и цветности

в концептуальной области, которая представляет из себя J пикселей в ширину и 2 пикселя в высоту. Тройка чисел (J:a:b) имеет следующий смысл:

J – ширина концептуальной области, обычно 4, но не обязательно.

a – количество выборок цветности (Cr, Cb) в первом ряду из J пикселей.

b – количество изменений выборок цветности (Cr, Cb) между первым и вторым рядом J пикселей. обычно значение b равно нулю или a (за исключением редких нестандартных случаев, таких как 4:4:1 и 4:2:1, которые не соответствуют этому соглашению).

На рис. 2 показаны примеры семплирование 4:4:4, 4:2:2 и 4:2:0.

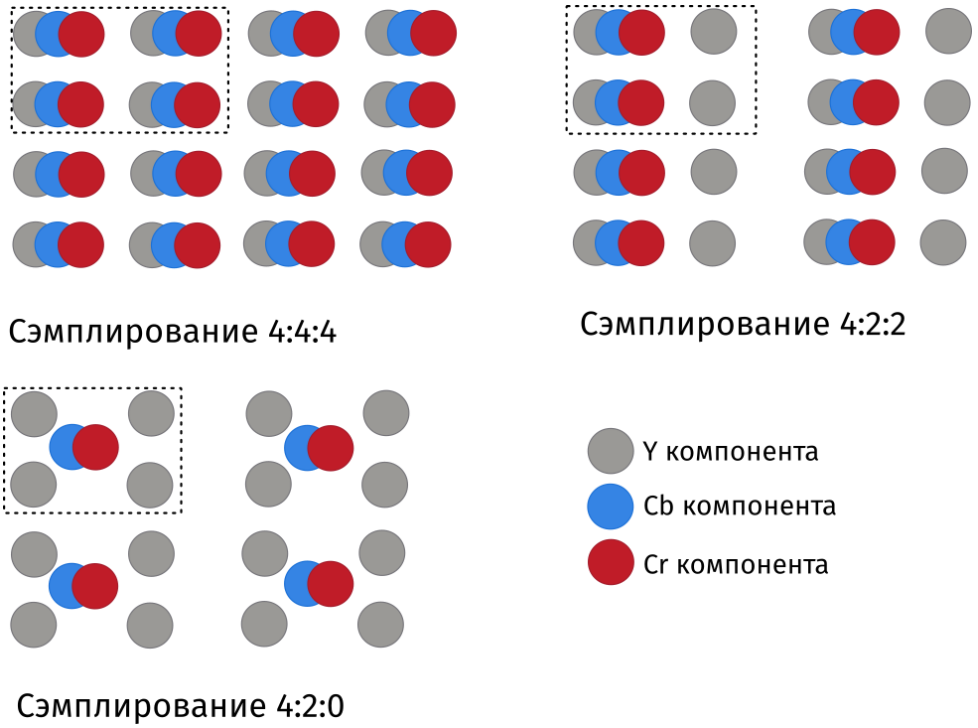

Рисунок 2 – Семплирование изображение в системе YCbCr

Следующий рисунок поясняет, как происходит кодирование цвета, при использовании субдискретизации компонент цветности.

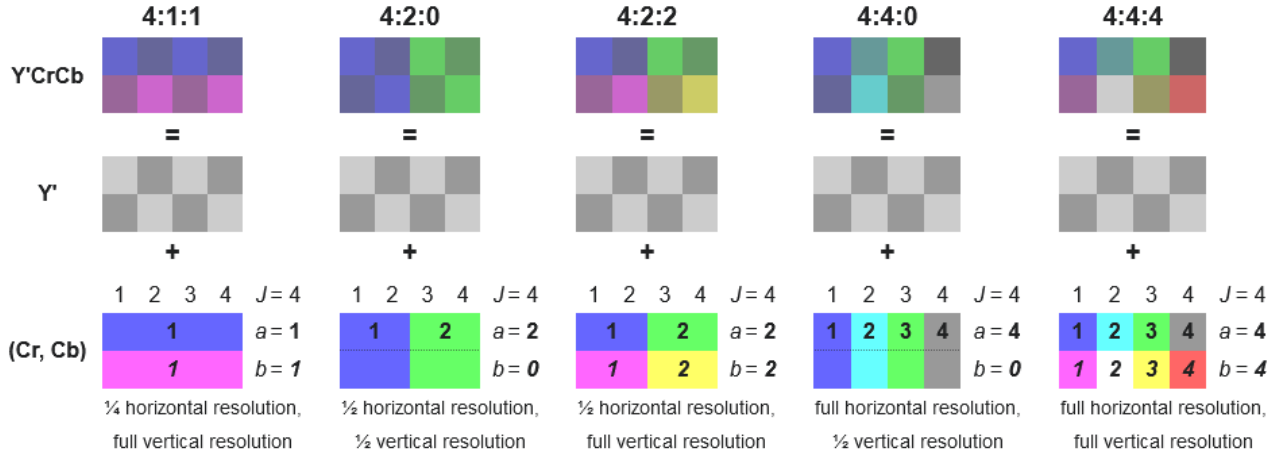

Рисунок 3 – Субдискретизация компонент цветности

1.2 Битовые плоскости

Рассмотрим произвольное изображение с L уровнями яркости. Эти уровни могут быть представлены с помощью $\log_2 L$ бит. Такое изображение можно разбить на **битовые плоскости.** Разбиение заключается в разделении одного изображения с L уровнями яркости на $\log_2 L$ бинарных изображений. При этом i-ое изображение получается путём выделения i-х битов из каждого пиксела исходного изображения. На следующем рисунке показан пример получения битовых плоскостей.

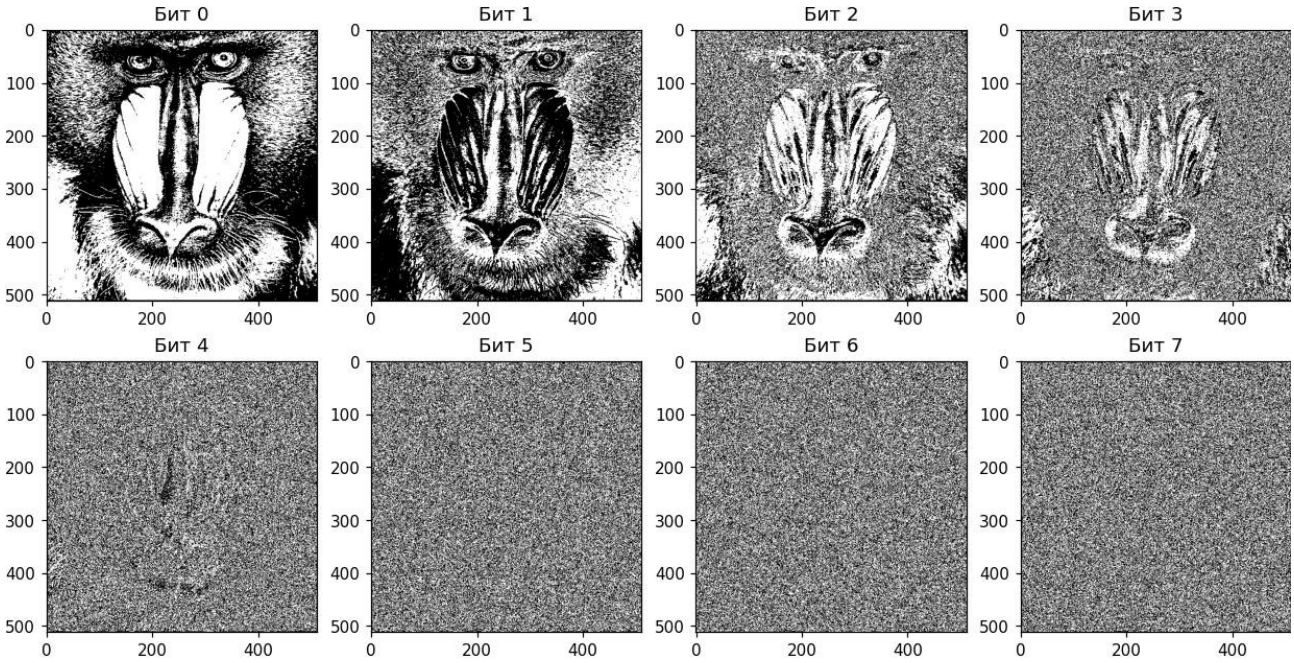

Рисунок 4 – Битовые плоскости

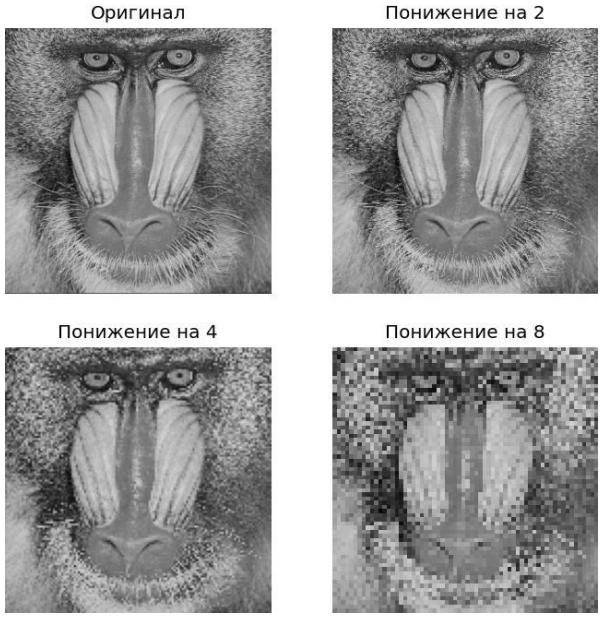

Рисунок 5 – Изображение с пространственным разрешением 512x512, 256x256, 128x128 и 64x64

1.3 Пространственное разрешение

Пространственное разрешение обычно приравнивается к числу пикселей на изображении, однако это не совсем корректно. Возможно иметь изображение, состоящее из блоков размером 2×2 пикселя с одинаковой интенсивностью. В этом случае число пикселей изображения будет равно $V \times H$ (где V – размер по вертикали, а H – по горизонтали), но пространственное разрешение такого изображения будет равно $\frac{V}{2} \times \frac{H}{2}$.

Пространственное разрешение может быть уменьшено при помощи оператора понижения дискретизации (англ. *downsample*) $\downarrow_n$. Индекс n определяет каким образом будет осуществляться понижение дискретизации. Например, оператор $\downarrow_2$ будет извлекать каждую вторую строку и каждый второй столбец для формирования нового изображения. Оператор $\downarrow_{n,m}$ будет извлекать каждую m-ю строку и n-й столбец.

На рисунке 5 приведен пример применения оператора понижения пространственного разрешения к изображению.

2 Порядок выполнения работы

1) Подготовить изображение своего лица в цифровом формате (цветное изображение с разрешением не более 800x800 пикселей в формате .jpeg или .png).

2) Используя средства Python загрузить изображение и отобразить его на экране.

3) Разбить загруженное RGB-изображение на три полутоновых изображения, соответствующих цветовым компонентам R (красный), G (зеленый) и B (синий). Отобразить полученные три изображения на экране.

4) Написать функцию Python для преобразования RGB-изображения в цветовое пространство YCbCr. Применить эту функцию к исходному RGB-изображению и отобразить на экране три полутоновых изображения, соответствующих яркостной компоненте Y и двум цветоразностным – Cb и Cr.

Выполнить субдискретизацию компонент цветности в соответствии с вариантом.

Таблица 1 – Варианты для задания 4

Номер варианта	Вид операции
1	4:1:1
2	4:2:0
3	8:2:0
4	8:1:1
5	8:2:2
6	16:4:0
7	16:2:2

8	32:4:0

5) Выполнить понижение пространственного разрешения изображения яркостной компоненты Y в соответствии с вариантом.

Таблица 1 – Варианты для задания 5 (S – исходное изображение, D – результирующее изображение)

Номер варианта	Вид операции
1	$D = \downarrow_{3} S$
2	$D = \downarrow_{3,2} S$
3	$D = \downarrow_{2,4} S$
4	$D = \downarrow_{3,4} S$
5	$D = \downarrow_{3,5} S$
6	$D = \downarrow_{5,3} S$
7	$D = \downarrow_{1,4} S$
8	$D = \downarrow_{5,2} S$

6) Представить полутоновое изображение, соответствующее яркостной компоненте Y (из задания 4) в виде набора из 8 двоичных изображений (битовых плоскостей).

7) Вычислить вероятности переходов между элементами изображений по горизонтали и по вертикали в каждой битовой плоскости одной цветовой компоненты и одной цветоразностной компоненты (Cb или Cr). Построить графики вероятности переходов от номера разряда цифрового полутонового изображения.

8) Оформление отчета в соответствии с СТП 01-2017.

3 Дополнительные задания

1. Восстановить изображение RGB из компонент Y и Cb.
2. Восстановить изображение RGB из компонент Y и Cr.
3. Восстановить изображение RGB из компонент Cb и Cr.
4. Взять только битовые плоскости старших битов компонент Y, Cb и Cr и восстановить RGB-изображение.
5. Восстановить изображение RGB из 4-х старших битовых плоскостей компонент Cb и Cr.
6. Восстановить RGB-изображение из 2-х старших битовых плоскостей компонент Y и Cr.

4 Контрольные вопросы

1. Какие цвета называются основными?
2. Что представляет собой цветовая модель RGB?
3. Что представляет собой цветовая модель YCbCr?

4. Изображение, представленное в какой цветовой системе, может иметь больший коэффициент сжатия. Почему?

5. Как определить статистическую избыточность в изображении?

6. Битовым плоскостям каких разрядов полутонового изображения будет присуща большая статистическая избыточность?

7. Яркостной или цветоразностным компонентам будет присуща большая статистическая избыточность? Как можно учесть это свойство при обработке изображения?