Бурмистрова Екатерина,

Горбунова Валерия

381808-1

Отчет по Лабораторной Работе

"Точечные фильтры"

В нашей лабораторной работе мы реализовали несколько фильтров для преобразования изображений.

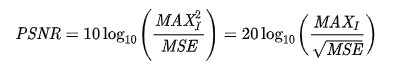
**1.Метрика сходства двух изображений psnr (def psnr(img1,img2)**

PSNR наиболее часто используется для измерения уровня искажений при сжатии изображений.

В случае использования MSE этот показатель для двух монохромных изображений I и K размера m×n, одно из которых считается зашумленным приближением другого, вычисляется по формуле:



PSNR определяется так:



Пример работы:



Сходство 35,7



Сходство 28,44 %

**2.Перевод изображения в оттенки серого.**

В данной лабораторной работе нами была выбрана реализация данного фильтра по модели ITU-R, BT. В процессе работы каждый пиксель изображения меняет свой цвет с r, g, b на 0.2126\*r + 0.7152\*g + 0.0722\*b.

Пример работы:

 оригинал

 результат

Данное исполнение занимает: ~4.3129 сек

 результат OpenCV

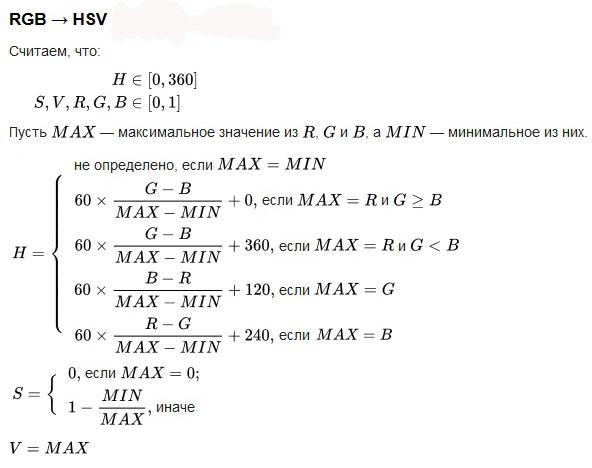
Исполнение с использованием OpenCV (grayscale): ~0.032 сек

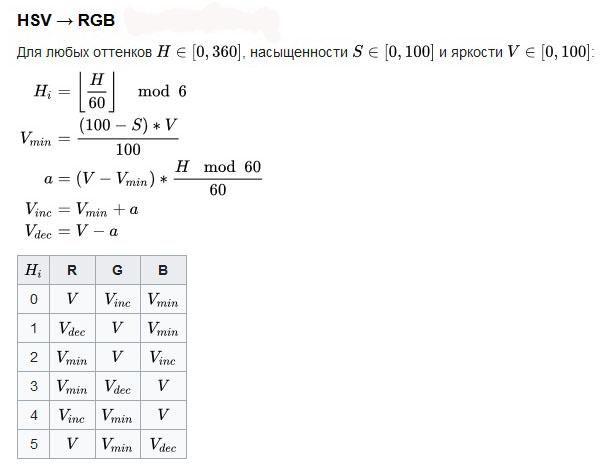
**3.Конвертация м/у цветовыми моделями**

**Конвертация RGB <-> HSV**

**а)**

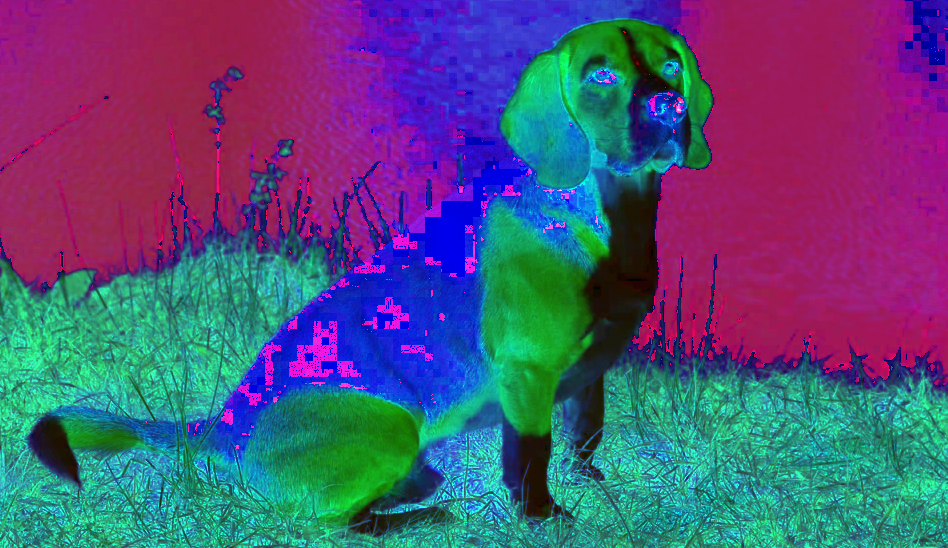
Проводится по следующим правилам:





В результате получаются значения параметров RGB в %. Чтобы привести их к стандартному представлению нужно каждый из параметров умножить на 255/100.

При целочисленном кодировании для каждого цвета в HSV есть соответствующий цвет в RGB. Однако обратное утверждение не является верным: некоторые цвета в RGB нельзя выразить в HSV так, чтобы значение каждого компонента было целым.

hsv с помощью matplotlib

**b)**

hsv с помощью OpenCV

**с)Сравнение результатов работы:**

**1.Pillow + matplotlib**

Т.к. изображение при использовании этого способа считывается как Image библиотеки pillow, оно может быть представлено только значениями типа uint8.

Скорость работы: ~0.4689 сек

**2.OpenCV**

Изображение может быть представлено значениями типа float. (при этом RGB хранится в формате BGR)

Скорость работы: ~0.015 сек

**d)Увеличение яркости - RGB представление (brightRGB(img1, brightness)**

Суть работы фильтра заключается в нахождении значений R,G и B для каждого пикселя и модификации каждого значения отдельно.

Пример работы:

(оригинал)





**Увеличение яркости - HSV представление (brightHSV(img1, brightness)**

Суть работы фильтра заключается в переводе изображения в HSV формат,получении значений H,S и V и увеличении пункта V на необходимую нам величину.

Затем мы возвращаем изображение в RGB формат.

Пример работы:



**e) Cравнение работы.**

Реализованная метрика показала 19% сходства с оригинальным изображением в случае работы с RGB,и 22% сходства в случае работы с HSV.

Скорость работы:

На выполнение RGB фильтра ушло 5,965 секунды

На выполнение HSV фильтра 0,002 секунды

Из этого можно сделать вывод,что использование hsv представления намного удобнее в работе с изменением яркости,так как не требует работы с каждым пикселем в отдельности.