**SANJEEV AGRAWAL GLOBAL EDUCATIONAL (SAGE) UNIVERSITY, BHOPAL**

****

**Bachelor of Computer Application (BCA) - Cloud Computing**

**With iNurture**

**Lab File**

**High Performance Computing**

**CA21B402**

**Name:**

**Enrollment No:**

**List of Experiments**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Name of Experiments** | **Date of Experiments** | **Signature** |
| **1** | **Implement the conversion of color image to a gray scale** | **20-03-2023** |  |
| **2** | **Demonstrate a Reduction technique on 3D image** | **26-03-2023** |  |
| **3** | **To deploy Chef Cluster using cook book** | **13-04-2023** |  |
| **4** | **Demonstrate the deployment of Puppet Cluster** | **27-04-2023** |  |
| **5** | **Installing MRG grid on Redhat Enterprise Linux** | **4-05-2023** |  |
| **6** | **Installation and configuration of Alchemi Grid** | **11-05-2023** |  |
| **7** | **Running a sample application on Alchemi Grid and analyzing it** | **18-05-2023** |  |
| **8** | **To study a Grid Simulation Toolkit** | **25-6-2023** |  |
| **9** | **To run two sample programs using GridSim Toolkit.** | **1-6-2023** |  |
| **10** | **To study a Cloud Simulation Toolkit** | **8-6-2023** |  |

**1. implement the conversion of color image to a gray grayscale image**

**Aim: conversion color image to gray image**

**1.1. Overview**

In this tutorial, we’ll talk about how we can convert an RGB image to grayscale. First, we’ll make a brief introduction to the color models, and then we’ll present three conversion methods along with an example.

**1.2. Color Models**

Representing colors as numerical values is a necessary step in many applications. To do this, we use models that are mathematical models that describe ways of mapping colors to a set of numbers. Usually, a color model defines three or four color components that are easily described through a coordinate system. Each color that the model can represent corresponds to a point in this coordinate system.

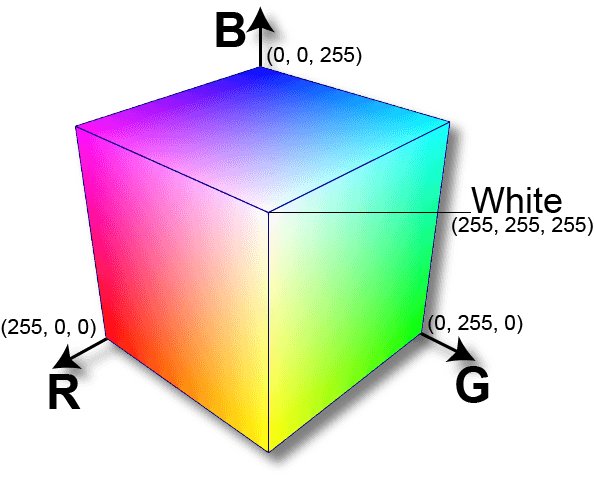
Along with these values, those models also provide a description of how to interpret these components in order to generate a color.

**1.2.1. RGB**

The most well-known color model is RGB which stands for Red-Green-Blue. As the name suggests, this model represents colors using individual values for red, green, and blue. The RGB model is used in almost all digital screens throughout the world.

Specifically, a color is defined using three integer values from 0 to 255 for red, green, and blue, where a zero value means dark and a value of 255 means bright. Given the values, the final color is defined when we mix these three basic colors weighted by their values.

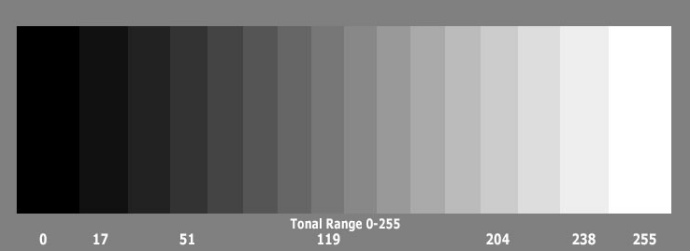
If we mix the three colors equally (RGB = (255, 255, 255)), we’ll get white while the absence of all colors (RGB = (0, 0, 0)) means black. Below there is the RGB coordinate system where we can see all the different colors that the model can describe:



**1.2.2. Grayscale**

Grayscale is the simplest model since it defines colors using only one component that is lightness. The amount of lightness is described using a value ranging from 0 (black) to 255 (white).

On the one hand, grayscale images convey less information than RGB. However, they are common in image processing because using a grayscale image requires less available space and is faster, especially when we deal with complex computations.

Below, we can see the full range of colors that the grayscale model can describe:

1.2.3. Others

Depending on the use case, there have been numerous proposed models throughout the years. For example, the CMYK model is used in printing and describes colors using values for cyan, magenta, yellow and black. [HSL and HSV](https://en.wikipedia.org/wiki/HSL_and_HSV) are two colors used by artists since they’re more aligned with the way human vision perceives a color. The basic components are hue, saturation, and lightness or value, respectively.

1.3. Convert RGB to Grayscale

Now, let’s talk about the main goal of this tutorial which is how to convert an image from RGB to grayscale. Since these models are based on different interpretations of color perception (primary colors and lightness), there is not a perfect way for conversion. There are some commonly used methods like the ones below.

1.3.1. Lightness Method

A very simple method is to take the average value of the components with the highest and lowest value:

We can easily see that this method presents a very serious weakness since one RGB component is not used. This is definitely a problem because the amount of lightness that our eye perceives depends on all three basic colors.

1.3.2. Average Method

Another method is to take the average value of the three components (red, green, and blue) as the grayscale value:

Although we now take into account all components, the average method is also problematic since it assigns the same weight to each component. Based on research on human vision, we know that our eyes react to each color in a different manner. Specifically, our eyes are more sensitive to green, then to red, and finally to blue. Therefore, the weights in the above equation should change.

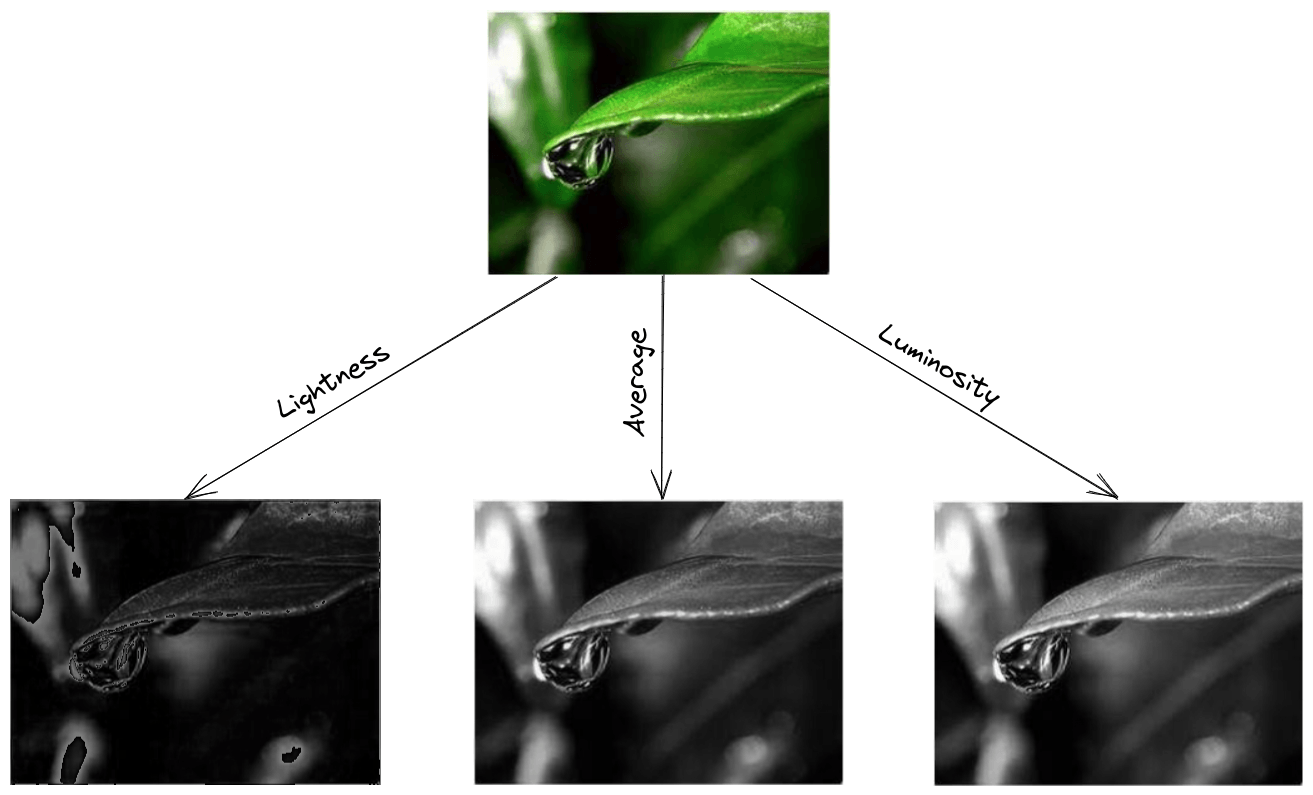
1.3.3. Luminosity Method

The best method is the luminosity method that successfully solves the problems of previous methods.

Based on the aforementioned observations, we should take a weighted average of the components. The contribution of blue to the final value should decrease, and the contribution of green should increase. After some experiments and more in-depth analysis, researchers have concluded in the equation below:

1.4. Example

Finally, let’s apply these three methods in an image to further illustrate our results. Below, we can see the results of applying the lightness, the average, and the luminosity method to convert an RGB image to grayscale:



As expected, the lightness method is the worst since it completely fails to capture the lightness of the input image. Among the other two, luminosity is the best since the average method produces a darker grayscale image.

1.5. Conclusion

In this tutorial, we discussed how we could convert an RGB image to grayscale. First, we talked about the different color models, and then we presented three conversion methods along with an example.

1.6 Example

Before showing you the coding demonstration you must install the [python image processing libraries](https://www.datasciencelearner.com/best-image-processing-library-in-python/). In our examples, I am using  Pillow, scikit-image, and the[Opencv python module](https://www.datasciencelearner.com/how-to-install-opencv-using-pip/). So install them. Secondly, the image used for explaining the examples is below.

[Convert an Image to Grayscale in Python Using the image.convert() Method of the pillow Library](https://www.delftstack.com/howto/python/convert-image-to-grayscale-python/#convert-an-image-to-grayscale-in-python-using-the-imageconvert-method-of-the--pillow-library)

The image.convert(mode, ..) method takes an image as input and converts it into the desired image type specified in the mode argument. The mode includes 1-bit and 8-bits pixel black and white images, RGB images etc.

As we want to convert our image to grayscale, we can pass 1 as mode argument for 1-bit black and white mode, L for 8-bits black and white image, and LA for alpha mode. The below example code demonstrates how to use the image.convert() method of the pillow library to convert an image to grayscale in Python:

Python PIL | Image.convert() Method

PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. The Image module provides a class with the same name which is used to represent a PIL image. The module also provides a number of factory functions, including functions to load images from files, and to create new images.

Image.convert() Returns a converted copy of this image. For the “P” mode, this method translates pixels through the palette. If mode is omitted, a mode is chosen so that all information in the image and the palette can be represented without a palette.

*Syntax: Image.convert(mode=None, matrix=None, dither=None, palette=0, colors=256)*

*Parameters:  
mode – The requested mode. See: Modes.  
matrix – An optional conversion matrix. If given, this should be 4- or 12-tuple containing floating point values.  
dither – Dithering method, used when converting from mode “RGB” to “P” or from “RGB” or “L” to “1”. Available methods are NONE or FLOYDSTEINBERG (default).  
palette – Palette to use when converting from mode “RGB” to “P”. Available palettes are WEB or ADAPTIVE.  
colors – Number of colors to use for the ADAPTIVE palette. Defaults to 256.*

*Returns: An Image object.*