Color Models

Author Name: e-Yantra Team
ERTS Lab, IIT Bombay

Prerequisites: Fundamentals of Image Processing, Python Basics

Software required: Python 2.7.15, NumPy 1.14.5, OpenCV 2.4.13.6

Lecture Notes:

1. Introduction:

The color space conversion employs a significant function in preprocessing phase of digital image processing. Color conversion can improve the quality of images. The color space conversion is used in various applications such as commercial, multimedia, computer vision, visual tracking systems etc.

Some basic technical terms in image processing are:

- **Visible Light:**

  In the electromagnetic spectrum, it is range of wavelengths (about **390 to 700 nm**) that are visible to human eye.

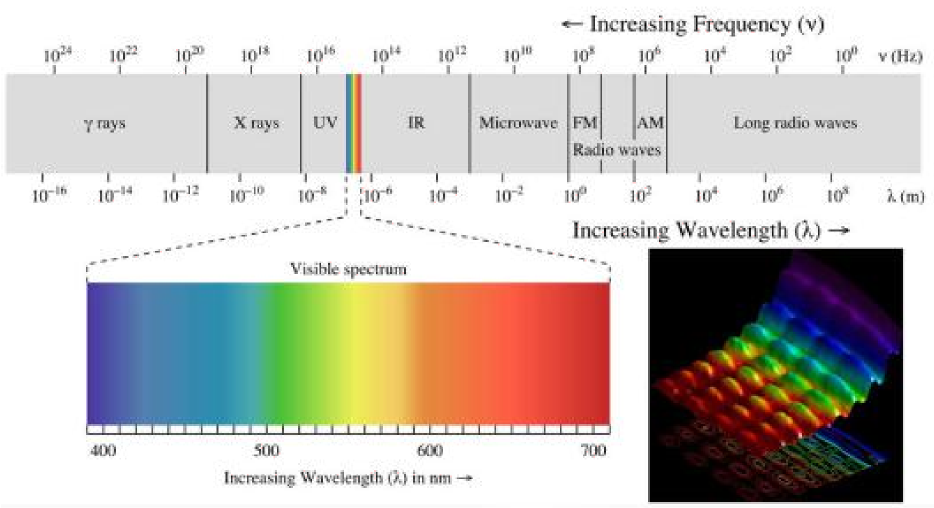

Figure 1: Spectrum for visible light (courtesy: factmyth.com)

- **Color Mixing:**

  There are 2 types of mixing:

  (a) Additive mixing

  (b) Subtractive mixing

  All colors can be created by mixing three primary colors.

(a) Additive color mixing:

In additive color mixing, colors are created using the three primary colors; **Red (R)**, **Green (G)**, and **Blue(B)**. Yellow can be formed by mixing Red and Green; Cyan by mixing Blue and Green and so on. On mixing all the three primary colors, White color is formed.

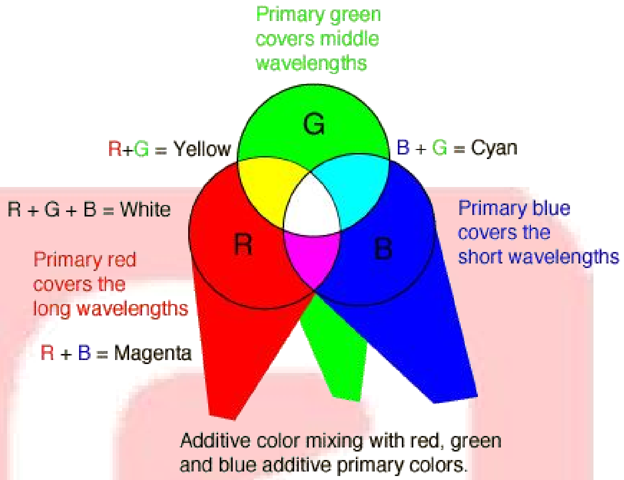

Figure 2: Additive color mixing (courtesy: hyperphysics.phy-astr.gsu.edu)

(b) Subtractive color mixing:

The primary colors in subtractive mixing are **Cyan (C)**, **Magenta (M)**, and **Yellow (Y)**. Subtractive color mixing in contrast with additive color mixing are created by subtracting parts of light spectrum present in white light. When all the three are overlapped in equal amount, light is subtracted and thus gives Black.

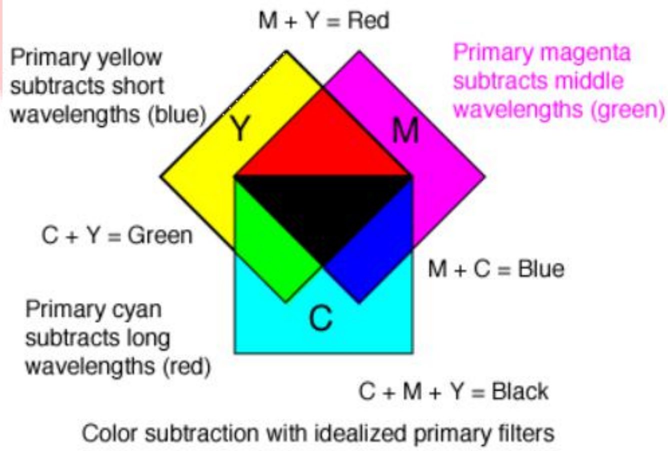

Figure 3: Subtractive color mixing (courtesy: hyperphysics.phy-astr.gsu.edu)

2. Color Models:

It is also called as color space or color system. Human eye has 3 different kinds of color sensors; each corresponding to RGB. Color Models uses mathematical formula in order to describe any color. Different color models are as described below:

(a) RGB Color Model:

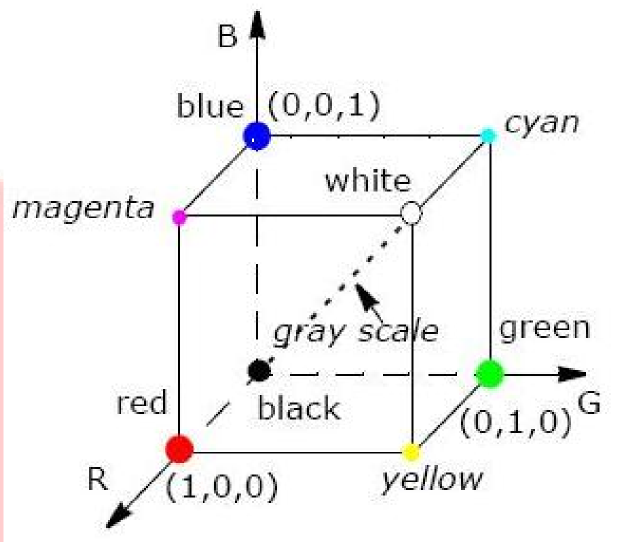

Figure 4: RGB Color Model (courtesy: scc.ustc.edu.cn)

The name of this model comes from the initials of the 3 additive primary colors; **Red (R)**, **Green (G)**, **Blue(B)**. Grayscale image is formed using only single matrix but color image is formed using 3 different images; Red, Green and Blue image.

It is basically additive color system in which primary colors: Red, Green and Blue light are added to form new colors. Origin **(0,0,0)** represents **black** while diagonally opposite vertex **(1,1,1)** is **white**. Vertices of the cube along the axes represent primary colors while the other vertices represents complementary colors for each primary color.

This model is mostly used in image analysis, computer graphics, image processing, storing images, etc.

(b) Grayscale Model:

Grayscale image consists of no color except the shades of gray. The darkest shade possible is black and lightest is white.

In this model, 255 means white color and 0 means black. All the other in between numbers are shades of gray that range from black to white.

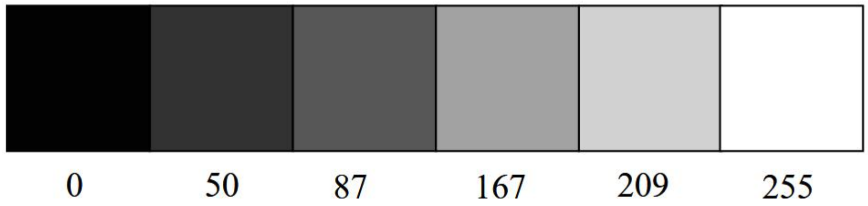

Figure 5: Grayscale Model (courtesy: processing.org)

Conversion from RGB to Grayscale color space:

To convert from RGB to Grayscale color space:

$$\boxed{Gray = 0.299 \times R + 0.587 \times G + 0.144 \times B}$$

(c) CMY and CMYK Color Model:

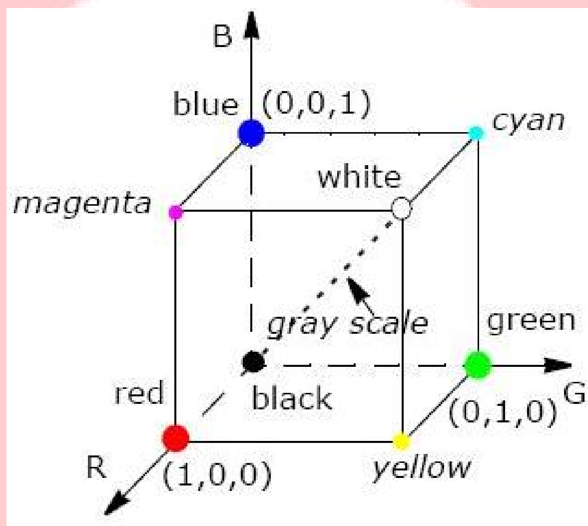

Figure 6: CMYK Color Model (courtesy: scc.ustc.edu.cn)

It is basically subtractive color model that stands for **Cyan (C)**, **Magenta (M)** and **Yellow (Y)**. It is mainly used in color printing. This model uses **CMY** as its primary colors and **RGB** as its secondary colors.

Printed color that looks Green absorbs other two components Red and Blue and reflects G. Thus, C-M-Y coordinates are simply the complements of R-G-B coordinates. The **CMYK** color model is variation of **CMY** color model. It adds one more color **K** (black).

When Cyan, Magenta and Yellow absorbs all Red, Green and Blue light; it should produce Black on paper. However, as Red, Green and Blue are not completely absorbed; there is no pure Black ink. To overcome this, printing industry added fourth ink i.e. black (K). Due to addition of fourth color, it could produce good black color on press.

Conversion from RGB to CMY color space and vice-versa:

To convert RGB to CMY color space:

$$\boxed{\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}}$$

To convert CMY to RGB color space:

$$\boxed{\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}}$$

(d) HSV or HSB Color Model:

Figure 7: HSV or HSB Color Model (courtesy: codeitdown.com)

Hue (H) refers to the purity of colour it resembles to. **Saturation (S)** simply describes how white the colour is. Pure red colour is fully saturated which means there is no addition of white **(S=1)**. **Value (V)** of a colour describes how dark the colour is. Black has Value = 0 that increases as it moves away from black.

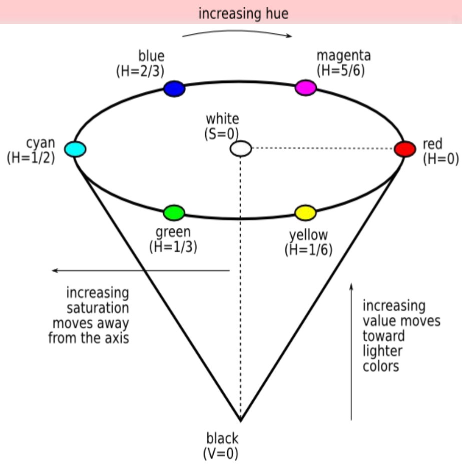

Figure 8: HSV or HSB Color Model values for different colors (courtesy: infohost.nmt.edu)

(e) HSL Color Model:

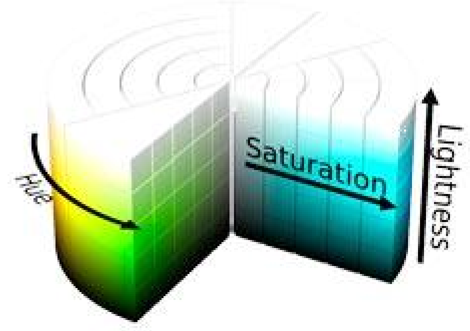

Figure 9: HSL Color Model (courtesy: codeitdown.com)

Every color is represented by three components: **Hue (H)**, **Saturation (S)** and **Lightness (L)**. HSV and HSB are exactly the same, but HSL is slightly different. H for Hue and S for Saturation is same for all cases. But, Lightness (L) is not same as Brightness/Value. At vertical axis where **L = 0.5** and **S = 1**, pure colors are found. **Black** and **White** is at **L = 0** and **L = 1** respectively.

HSV and HSL Color Models are used in human visual perception, computer graphics processing, face recognition etc.

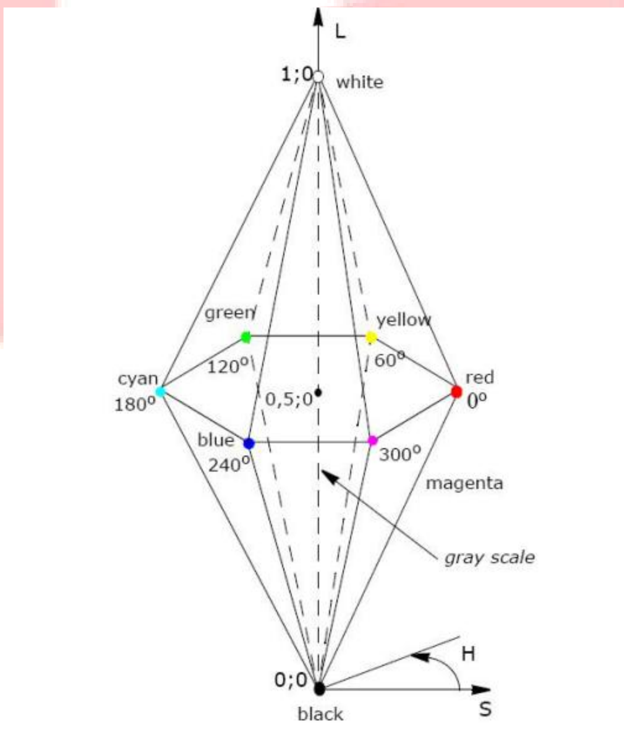

Figure 10: HSL Color Model values for different colors (courtesy: scc.ustc.edu.cn)

(f) HSI Color Model:

The Hue and Saturation components have exactly the same meaning for HSV/HSL/HSI. Like HSL color model, HSI is also a double hexagon model. Its bottom is similar to HSV color model. Top converges to a point that corresponds to white color. The line between top and bottom point consist of varying shades of gray. For this model, max saturation is available at medium gray intensity. This model is mostly used in developing image processing algorithms.

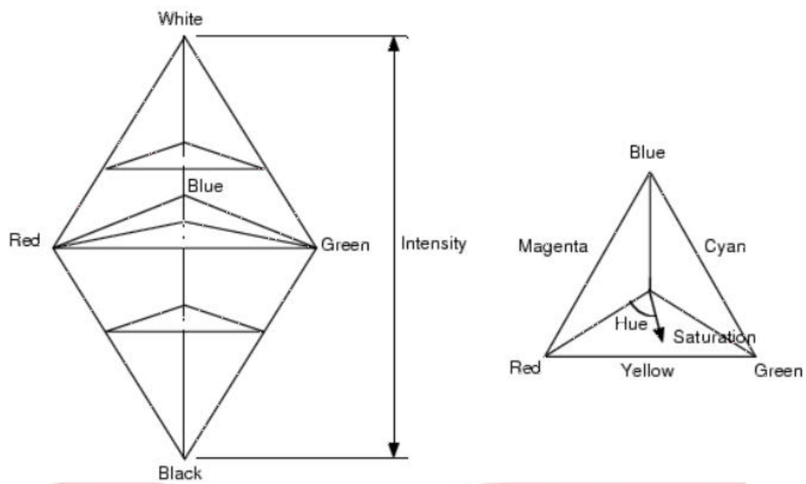

Figure 11: HSI Color Model (courtesy: homepages.inf.ed.ac.uk)

<u>Conversion from RGB to HSV/HSL/HSI color space</u>:

Note: In case of 8-bit images, **R, G** and **B** are converted to the floating-point format and scaled to fit in the **0 to 1** range.

Some common terms between conversions of RGB to HSV, HSL and HSI color space:

$$\boxed{R' = R/255} \quad ; \quad \boxed{G' = G/255} \quad ; \quad \boxed{B' = B/255}$$

$$\boxed{C_{\max} = \max(R', G', B')} \quad ; \quad \boxed{C_{\min} = \min(R', G', B')} \quad ; \quad \boxed{\Delta = C_{\max} - C_{\min}}$$

<u>To convert RGB to HSV color space</u>:

$$\boxed{H \leftarrow \begin{cases} 60^\circ \times \left(\dfrac{G' - B'}{\Delta}\right) & , \; if \; C_{\max} = R' \\ 120^\circ + 60^\circ \times \left(\dfrac{B' - R'}{\Delta}\right) & , \; if \; C_{\max} = G' \\ 240^\circ + 60^\circ \times \left(\dfrac{R' - G'}{\Delta}\right) & , \; if \; C_{\max} = B' \end{cases}} \qquad \boxed{S \leftarrow \begin{cases} 0 & , \quad C_{\max} = 0 \\ \dfrac{\Delta}{C_{\max}} & , \quad C_{\max} \neq 0 \end{cases}} \qquad \boxed{V \leftarrow C_{\max}}$$

If $\boxed{H < 0}$ then $\boxed{H \leftarrow H + 360}$

On output, $\boxed{0 \le V \le 1}$, $\boxed{0 \le S \le 1}$, $\boxed{0 \le H \le 360}$

These values have to be converted to the destination data type that is same as that of the source image.

Hence, for 8-bit images,

$$\boxed{V \leftarrow 255\,V} \quad , \quad \boxed{S \leftarrow 255\,S} \quad , \quad \boxed{H \leftarrow H/2 \text{ (to fit to 0 to 255)}}$$

<u>To convert RGB to HSL color space</u>:

$$\boxed{H \leftarrow \begin{cases} 60^\circ \times \left(\dfrac{G'-B'}{\Delta}\right) & , \; if\; C_{\max} = R' \\ 120^\circ + 60^\circ \times \left(\dfrac{B'-R'}{\Delta}\right) & , \; if\; C_{\max} = G' \\ 240^\circ + 60^\circ \times \left(\dfrac{R'-G'}{\Delta}\right) & , \; if\; C_{\max} = B' \end{cases}} \qquad \boxed{L \leftarrow \frac{(C_{\max} + C_{\min})}{2}}$$

$$\boxed{S \leftarrow \begin{cases} \dfrac{\Delta}{C_{\max} + C_{\min}} & , \; if\; L < 0.5 \\ \dfrac{\Delta}{2 - (C_{\max} + C_{\min})} & , \; if\; L \ge 0.5 \end{cases}}$$

If $\boxed{H < 0}$ then $\boxed{H \leftarrow H + 360}$

On output, $\boxed{0 \le L \le 1}$, $\boxed{0 \le S \le 1}$, $\boxed{0 \le H \le 360}$

These values have to be converted to the destination data type that is same as that of the source image.

Hence, for 8-bit images,

$$\boxed{L \leftarrow 255\,L} \quad , \quad \boxed{S \leftarrow 255\,S} \quad , \quad \boxed{H \leftarrow H/2 \text{ (to fit to 0 to 255)}}$$

<u>To convert RGB to HSI color space</u>:

$$\boxed{H \leftarrow \begin{cases} \theta & , \; if\; B' \le G' \\ 360^\circ - \theta & , \; if\; B' > G' \end{cases}} \quad \text{where} \quad \boxed{\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}\left[(R'-G') + (R'-B')\right]}{\left[(R'-G')^2 + (R'-B')(G'-B')\right]^{\frac{1}{2}}} \right\}}$$

$$\boxed{I \leftarrow \frac{1}{3}(R'+G'+B')} \qquad \boxed{S \leftarrow \begin{Bmatrix} 0 & , & I = 0 \\ 1-\dfrac{C_{min}}{I} & , & I \neq 0 \end{Bmatrix}}$$

On output, $\boxed{0 \leq I \leq 1}$, $\boxed{0 \leq S \leq 1}$, $\boxed{0 \leq H \leq 360}$

These values have to be converted to the destination data type that is same as that of the source image.

Hence, for 8-bit images,

$$\boxed{I \leftarrow 255\ I} \quad , \quad \boxed{S \leftarrow 255\ S} \quad , \quad \boxed{H \leftarrow H/2 \text{ (to fit to 0 to 255)}}$$

<u>Conversion from HSV/HSL/HSI to RGB color space</u>:

Note: Before conversion, the values of **H → [0°, 360°)**, of **S → [0,1]** and of **V/L/I → [0,1]**.

<u>To convert HSV and HSL to RGB color space</u>:

For HSV, $\boxed{C = V \times S}$; For HSL, $\boxed{C = (1-|2L-1|)\times S}$

For both HSV and HSL, $\boxed{X = C \times (1-|(H/60°) \bmod 2 - 1|)}$

$$\boxed{(R', G', B') \leftarrow \begin{Bmatrix} (0,0,0) & \textit{if H is undefined} \\ (C, X, 0) & 0° \leq H < 60° \\ (X, C, 0) & 60° \leq H < 120° \\ (0, C, X) & 120° \leq H < 180° \\ (0, X, C) & 180° \leq H < 240° \\ (X, 0, C) & 240° \leq H < 300° \\ (C, 0, X) & 300° \leq H < 360° \end{Bmatrix}}$$

For HSV, $\boxed{m = V - C}$; For HSL, $\boxed{m = L - C/2}$

For both HSV and HSL, $\boxed{(R,G,B) \leftarrow (R'+m, G'+m, B'+m)}$

<u>To convert HSI to RGB color space</u>:

- If $\boxed{0° \leq H < 120°}$ (RG sector) then

$$\boxed{B' \leftarrow I(1-S)} \quad \boxed{R' \leftarrow I\left[1+\frac{S\cos(H)}{\cos(60°-H)}\right]} \quad \boxed{G' \leftarrow 3I - B' - R'}$$

- If $\boxed{120° \le H < 240°}$ (GB sector) then

$$\boxed{R' \leftarrow I(1-S)} \quad \boxed{G' \leftarrow I\left[1+\frac{S\cos(H-120°)}{\cos(180°-H)}\right]} \quad \boxed{B' \leftarrow 3I - R' - G'}$$

- If $\boxed{240° \le H < 360°}$ (BR sector) then

$$\boxed{G' \leftarrow I(1-S)} \quad \boxed{B' \leftarrow I\left[1+\frac{S\cos(H-240°)}{\cos(300°-H)}\right]} \quad \boxed{R' \leftarrow 3I - G' - B'}$$

On output, $\boxed{0 \le R' \le 1}$, $\boxed{0 \le G' \le 1}$, $\boxed{0 \le B' \le 1}$

These values have to be converted to the destination data type that is same as that of the source image.

Hence, for 8-bit images,

$$\boxed{R \leftarrow 255\ R'} \quad , \quad \boxed{G \leftarrow 255\ G'} \quad , \quad \boxed{B \leftarrow 255\ B'}$$

3. Commands / Functions:

(a) cv2.cvtColor(src, code) ⟶ dst

Use: To convert an image from one color space to another

Parameters:

- **src:** input image
- **dst:** output image of same size and depth as src
- **code:** color space conversion code

Codes for color space conversion:

- for RGB to grayscale conversion: **cv2.COLOR_BGR2GRAY**
- for grayscale to RGB conversion: **cv2.COLOR_GRAY2BGR**
- for RGB to CMY conversion and inverse: support unavailable in OpenCV, but possible using NumPy array computation
- for RGB to HSV conversion: **cv2.COLOR_BGR2HSV**
- for HSV to RGB conversion: **cv2.COLOR_HSV2BGR**
- for RGB to HSL conversion: **cv2.COLOR_BGR2HLS**
- for HSL to RGB conversion: **cv2.COLOR_HLS2BGR**
- for RGB to HSI conversion and inverse: support unavailable in OpenCV, but possible using NumPy array computation

4. Program:

Problem Statement:

Read a image (**'color_image.jpg'** is provided under Related Resources tab on portal) using an OpenCV function, check the type of image, if not grayscale, convert a multi-channel image to grayscale image using OpenCV, check its type and display it.

Program Snippet:

```
# import OpenCV and NumPy
import cv2
import numpy as np

# read an image
rgb_img = cv2.imread('color_image.jpg')

# 'shape' attribute of NumPy array returns its dimensions
# in case of images, format is: (height, width, channels);
# if channels = 3, then the image is colored image having
# Red, Green and Blue channels; while grayscale image has
# only one channel, hence 'shape' will return height and width
print "RGB Image shape = ", rgb_img.shape

# convert the color space from RGB to Grayscale
gray_img = cv2.cvtColor(rgb_img, cv2.COLOR_BGR2GRAY)

# print the shape attribute of grayscale image
# to verify the channels in the grayscale image = 1
print "Grayscale Image shape = ", gray_img.shape

# display both images
cv2.imshow('original color image', rgb_img)
cv2.imshow('grayscale image', gray_img)

# close and destroy windows if any key is pressed
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:

```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: I:/eYRDC/Image Processing Lessons/Codes/color_models.py ================
RGB Image shape =  (300L, 480L, 3L)
Grayscale Image shape =  (300L, 480L)
>>> |
```

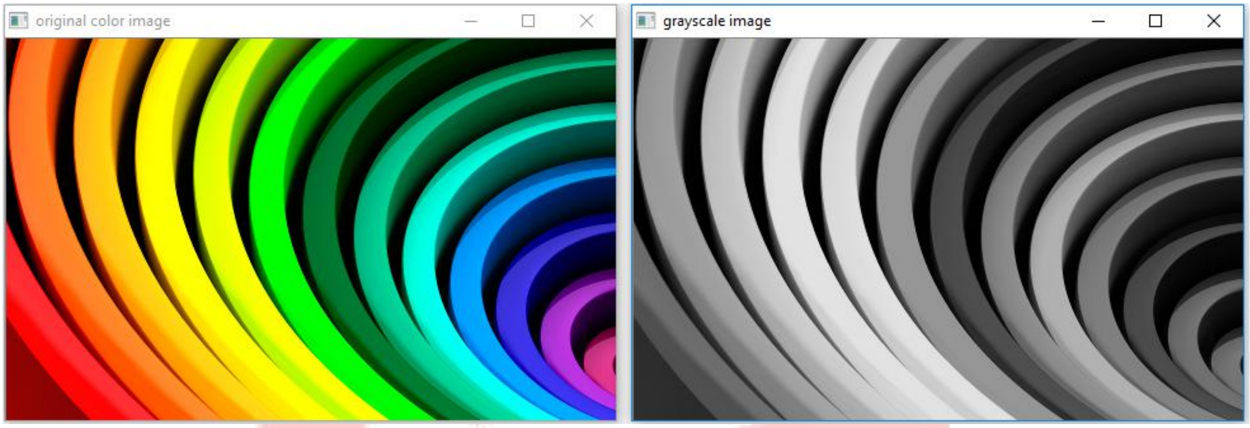

Figure 12: Output of the sample program

References:

[1] Digital Image Processing book by Rafael C. Gonzalez and Richard Eugene Woods
[2] http://www.tutorialspoint.com/dip/Introduction_to_Color_Spaces.htm
[3] https://www.slideshare.net/EngHaitham/color-models-42915934?next_slideshow=1
[4] https://www.slideshare.net/vishnurcvijayan/hsl-hsv-colour-models
[5] http://www.booksmartstudio.com/color_tutorial/colortheory4.html
[6] https://www.vocal.com/video/rgb-and-hsvhsihsl-color-space-conversion/