**Assignment No.**

**Problem Statement:** Write a program to do following image format conversion Using different image processing function using Open CV.

1. RGB to GRAY.
2. GRAY to RGB.
3. RGB to HSV.
4. HSV to RGB.

**RGB to GRAY:**

* **Theory:** RGB color model stores individual values for red, green, and blue. With a color space based on the RGB color model, the three primaries are added together to create colors from completely white to completely black.

A grayscale (or gray level) image is simply one in which the only colors are shades of gray.

When converting an RGB image to grayscale, we have to take the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel. One such approach is to take the average of the contribution from each channel: (R+B+C)/3.

* **Algorithm:** 
  1. Read an image by using imread() and store it in img variable.
  2. Use cvtColor() and pass the img and cv2.COLOR\_RGB2GRAY and store the object in img\_converted
  3. Show the image using imshow() function
  4. Call the method waitKey() and pass 0 in it
  5. Destroy all the existing windows previously opened
  6. Save the image using imwrite() method by passing arguments “converted\_img\_name,jpg” and img\_converted
  7. End.
* **Source Code:**

import cv2

img = cv2.imread("./download.jpg")

img\_converted = cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)

cv2.imshow(“./download.jpg”, img\_converted)

cv2.waitKey(0)

cv2.destroyAllWindows()

cv2.imwrite(“download\_converted.jpg”, img\_converted)

* **Output:**

[[70 47 51 ... 39 40 40]

[69 49 52 ... 38 39 39]

[67 53 53 ... 36 37 37]

...

[62 61 64 ... 20 20 19]

[66 64 65 ... 18 17 17]

[70 68 68 ... 32 32 33]]

**GRAY to RGB:**

* **Theory:** Average method is the simplest one. You just have to take the average of three colors. Since its an RGB image, so it means that you have add r with g with b and then divide it by 3 to get your desired grayscale image. It’s done in this way. Grayscale = (R + G + B / 3)
* **Algorithm:** 
  1. Read an image in grayscale mode by passing an optional argument 0 in imread() and store it in img variable.
  2. Use cvtColor() and pass the img and cv2.COLOR\_GRAY2RGB and store the object in img\_converted
  3. Show the image using imshow() function
  4. Call the method waitKey() and pass 0 in it
  5. Destroy all the existing windows previously opened
  6. Save the image using imwrite() method by passing arguments “converted\_img\_name,jpg” and img\_converted
  7. End.
* **Source Code:**

import cv2

img = cv2.imread("./download.jpg", 0)

img\_converted = cv2.cvtColor(img, cv2.COLOR\_GRAY2RGB)

cv2.imshow(“./download.jpg”, img\_converted)

cv2.waitKey(0)

cv2.destroyAllWindows()

cv2.imwrite(“download\_converted.jpg”, img\_converted)

* **Output:**

[[[65 65 65]

[42 42 42]

[46 46 46]

...

[35 35 35]

[35 35 35]

[35 35 35]]

[[64 64 64]

[44 44 44]

[47 47 47]

...

[34 34 34]

[34 34 34]

[34 34 34]]

[[62 62 62]

[48 48 48]

[48 48 48]

...

[33 33 33]

[33 33 33]

[33 33 33]]

...

[[64 64 64]

[62 62 62]

[64 64 64]

...

[20 20 20]

[20 20 20]

[19 19 19]]

[[68 68 68]

[65 65 65]

[65 65 65]

...

[17 17 17]

[17 17 17]

[16 16 16]]

[[72 72 72]

[69 69 69]

[68 68 68]

...

[32 32 32]

[32 32 32]

[32 32 32]]]

**RGB to HSV:**

* **Theory:** HSV – (hue, saturation, value), also known as HSB (hue, saturation, brightness), is often used by artists because it is often more natural to think about a color in terms of hue and saturation than in terms of additive or subtractive color components. HSV is a transformation of an RGB color space, and its components and colorimetry are relative to the RGB color space from which it was derived.

The RGB color model is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

* **Algorithm:** 
  1. Read an image in imread() and store it in img variable.
  2. Use cvtColor() and pass the img and cv2.COLOR\_RGB2HSV and store the object in img\_converted
  3. Show the image using imshow() function
  4. Call the method waitKey() and pass 0 in it
  5. Destroy all the existing windows previously opened
  6. Save the image using imwrite() method by passing arguments “converted\_img\_name,jpg” and img\_converted
  7. End.
* **Source Code:**

import cv2

img = cv2.imread("./download.jpg")

img\_converted = cv2.cvtColor(img, cv2.COLOR\_RGB2HSV)

cv2.imshow(“./download.jpg”, img\_converted)

cv2.waitKey(0)

cv2.destroyAllWindows()

cv2.imwrite(“download\_converted.jpg”, img\_converted)

* **Output:**

[[[ 42 132 79]

[ 42 187 56]

[ 42 174 60]

...

[ 36 156 44]

[ 35 176 45]

[ 35 176 45]]

[[ 42 134 78]

[ 42 180 58]

[ 42 171 61]

...

[ 36 160 43]

[ 36 180 44]

[ 35 180 44]]

[[ 42 138 76]

[ 42 169 62]

[ 42 169 62]

...

[ 37 155 41]

[ 38 178 43]

[ 35 176 42]]

...

[[ 66 163 83]

[ 64 153 80]

[ 58 128 80]

...

[ 60 204 30]

[ 59 193 29]

[ 59 200 28]]

[[ 66 155 87]

[ 64 147 83]

[ 58 126 81]

...

[ 58 227 27]

[ 57 214 25]

[ 54 223 24]]

[[ 66 149 91]

[ 64 141 87]

[ 58 121 84]

...

[ 57 134 40]

[ 57 118 39]

[ 53 118 39]]]

**HSV to RGB:**

* **Theory:** HSV – (hue, saturation, value), also known as HSB (hue, saturation, brightness), is often used by artists because it is often more natural to think about a color in terms of hue and saturation than in terms of additive or subtractive color components. HSV is a transformation of an RGB color space, and its components and colorimetry are relative to the RGB color space from which it was derived.

The RGB color model is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

* **Algorithm:** 
  1. Read an image in imread() and store it in img variable.
  2. Use cvtColor() and pass the img and cv2.COLOR\_HSV2RGB and store the object in img\_converted
  3. Show the image using imshow() function
  4. Call the method waitKey() and pass 0 in it
  5. Destroy all the existing windows previously opened
  6. Save the image using imwrite() method by passing arguments “converted\_img\_name,jpg” and img\_converted
  7. End.
* **Source Code:**

import cv2

img = cv2.imread("./download.jpg")

img\_converted = cv2.cvtColor(img, cv2.COLOR\_HSV2RGB)

cv2.imshow(“./download.jpg”, img\_converted)

cv2.waitKey(0)

cv2.destroyAllWindows()

cv2.imwrite(“download\_converted.jpg”, img\_converted)

* **Output:**

[[[26 38 27]

[13 15 11]

[16 19 14]

...

[16 17 14]

[13 14 12]

[13 14 12]]

[[25 37 26]

[15 17 13]

[17 20 15]

...

[15 16 13]

[12 13 11]

[12 13 11]]

[[24 35 24]

[18 21 15]

[18 21 15]

...

[16 16 13]

[13 13 11]

[13 13 11]]

...

[[40 40 26]

[37 38 26]

[34 40 27]

...

[ 6 5 5]

[ 7 6 6]

[ 6 5 5]]

[[41 44 28]

[38 41 27]

[34 41 27]

...

[ 3 3 3]

[ 4 4 4]

[ 3 3 3]]

[[43 48 30]

[40 45 29]

[35 44 29]

...

[19 18 16]

[21 20 18]

[21 20 18]]]