**Assignment No.**

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

1. RGB to BINARY.
2. GRAY to BINARY.
3. RGB to YCbCr.
4. YCbCr to RGB.

**Actual Image:**



**RGB to BINARY:**

* **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 binary image is one that consists of pixels that can have one of exactly two colors, usually black and white. Binary images are also called bi-level or two-level.

* **Algorithm:** 
  1. Read an image by using imread() and store it in img variable.
  2. Use cv2.threshold() and pass img, 127, 256, and cv2.THRESH\_BINARY and store it in binary\_image variable
  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 binary\_image
  7. End.
* **Source Code:**

import cv2

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

ret, binary\_image = cv2.threshold(img, 127, 256, cv2.THRESH\_BINARY)

cv2.imshow("image", binary\_image)

cv2.imwrite("./download\_converted\_bin.jpg", binary\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

* **Output:**



**GRAY to BINARY:**

* **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). A binary image is one that consists of pixels that can have one of exactly two colors, usually black and white. Binary images are also called bi-level or two-level.
* **Algorithm:** 
  1. Read an image in grayscale mode by passing an optional argument 0 in imread() and store it in img variable.
  2. Use cv2.threshold() and pass img, 127, 256, and cv2.THRESH\_BINARY and store it in binary\_image variable
  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 binary\_image
  7. End.
* **Source Code:**

import cv2

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

ret, binary\_image = cv2.threshold(img, 127, 256, cv2.THRESH\_BINARY)

cv2.imshow("image", binary\_image)

cv2.imwrite("./download\_converted\_bin.jpg", binary\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

* **Output:**



**RGB to YCbCr:**

* **Theory:** Here Y is the luma component of the color. Luma component is the brightness of the color. That means the light intensity of the color. The human eye is more sensitive to this component.

Cb and Cr is the blue component and red component related to the Chroma component. That means “Cb is the blue component relative to the green component. Cr is the red component relative to the green component.” These components are less sensitive to the human eyes.

Since the Y component is more sensitive to the human eye, it needs to be more correct and Cb and Cr is less sensitive to the human eye. Therefore, it needs not to be more accurate. When in JPEG compression, it uses these sensitivities of the human eye and eliminate the unnecessary details of the image.

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\_RGB2YCrCb 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\_RGB2YCrCb)

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

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

cv2.waitKey(0)

cv2.destroyAllWindows()

* **Output:**



**YCbCr to RGB:**

* **Theory:** Here Y is the luma component of the color. Luma component is the brightness of the color. That means the light intensity of the color. The human eye is more sensitive to this component.

Cb and Cr is the blue component and red component related to the Chroma component. That means “Cb is the blue component relative to the green component. Cr is the red component relative to the green component.” These components are less sensitive to the human eyes.

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\_YCrCb2RGB 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\_YCrCb2RGB)

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

cv2.waitKey(0)

cv2.destroyAllWindows()

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

* **Output:**

