# Lab-1: Image Enhancement Codes

Adjusting brightness and contrast

There are several ways to adjust the brightness and contrast of an image using OpenCV and Python. One common method is to use the cv2.addWeighted() function, which allows you to adjust the brightness by adding a scalar value to each pixel in the image, and the contrast by scaling the pixel values.

#Import the necessary libraries

import cv2

import matplotlib.pyplot as plt

import numpy as np

# Load the image

image = cv2.imread('GFG.jpeg')

#Plot the original image

# Adjust the brightness and contrast

# Adjusts the brightness by adding 10 to each pixel value

# Adjusts the contrast by scaling the pixel values by 2.3

image2 = cv2.addWeighted(image, contrast, np.zeros(image.shape, image.dtype), 0, brightness)

#Save the image

cv2.imwrite('modified\_image.jpg', image2)

#Plot the contrast image

Sharpening images

Sharpening is the process of enhancing the edges and fine details in an image to make it appear sharper and more defined. It is important because it can help to bring out the details and features in an image, making it more visually appealing and easier to understand.

One common method for sharpening images using OpenCV and Python is to use the cv2.filter2D() function, which convolves the image with a kernel. The kernel can be designed to enhance the edges in the image, resulting in a sharper image.

#Import the necessary libraries

import cv2

import matplotlib.pyplot as plt

import numpy as np

# Load the image

image = cv2.imread('GFG.jpeg')

#Plot the original image

# Create the sharpening kernel

# Sharpen the image

sharpened\_image = cv2.filter2D(image, -1, kernel)

#Save the image

cv2.imwrite('sharpened\_image.jpg', sharpened\_image)

#Plot the sharpened image

plt.title("Sharpening")

Enhancing color in images

There are several ways to enhance the colors in an image using OpenCV and Python. One common method is to use the cv2.cvtColor() function, which allows you to convert the image from one color space to another. This can be useful for adjusting the color balance or saturation of the image.  
Here is an example of how to enhance the colors in an image using the cv2.cvtColor() function:

#Import the necessary libraries

import cv2

import matplotlib.pyplot as plt

import numpy as np

# Load the image

image = cv2.imread('GFG.jpeg')

#Plot the original image

# Convert the image from BGR to HSV color space

image = cv2.cvtColor(image, cv2.COLOR\_RGB2HSV)

# Adjust the hue, saturation, and value of the image

# Adjusts the hue by multiplying it by 0.7

image[:, :, 0] = image[:, :, 0] \* 0.7

# Adjusts the saturation by multiplying it by 1.5

image[:, :, 1] = image[:, :, 1] \* 1.5

# Adjusts the value by multiplying it by 0.5

image[:, :, 2] = image[:, :, 2] \* 0.5

# Convert the image back to BGR color space

image2 = cv2.cvtColor(image, cv2.COLOR\_HSV2BGR)

#Save the image

cv2.imwrite('enhanced coloured.jpg', image2)

#Plot the enhanced image