# Color Image Enhancement and Sharpening

Govt. Graduate College Burewala

**Subject name:**

Image Processing.

**Submitted To:**

Prof. Rimsha.

**Submitted By:**

Syeda Ayesha Jamshed.
Javeria Asghar.

**Roll No:**

100039
100073

**Semester:**

BSCS (sem-5)

# INTRODUCTION

Digital image processing is a field of computer science and technology that deals with manipulating images using computers. It allows us to improve the quality of images, extract useful information, and prepare them for further analysis or presentation. Images captured through cameras or other devices may not always be perfect. They can appear blurry, have poor contrast, dull colors, or other defects due to lighting, motion, or camera limitations. Image processing provides tools and techniques to correct these problems and make images visually appealing and more informative.

One of the key aspects of image processing is image enhancement. Enhancement refers to improving the visual quality of an image to make it clearer and more appealing to the human eye. It can involve increasing brightness, adjusting contrast, improving sharpness, and enhancing colors. In this project, color images were specifically enhanced to make them vibrant, rather than converting them to grayscale, because color plays an important role in how humans perceive and interpret images.

The difference between the original and enhanced images is clearly noticeable. The original image may look dull, with faded colors and slightly blurred edges, while the enhanced image has brighter, richer colors and sharper details. By improving contrast and sharpness, the image becomes more visually striking and easier to analyze. Color enhancement is particularly important in applications where color conveys critical information, such as photography, medical imaging, and satellite imagery. Maintaining the color integrity of the image ensures that the final result is not only visually attractive but also accurate in representing the real-world scene.

Through this project, the process of image enhancement was implemented using Python and OpenCV. Techniques such as contrast adjustment using CLAHE (Contrast Limited Adaptive Histogram Equalization) and image sharpening were applied to improve the overall quality of the images. This project demonstrates how simple yet effective digital image processing methods can transform ordinary images into high-quality visuals suitable for various applications.

## PROBLEM STATEMENT & OBJECTIVES

In many real-world scenarios, digital images often suffer from quality issues. Images captured by cameras or other devices can appear blurry due to motion, focus errors, or low-resolution sensors. In addition, the colors in such images may look dull or washed out, and poor contrast can make important details less visible.

These problems reduce the overall quality of the image and make it less useful for analysis, presentation, or practical applications. Enhancing images is therefore necessary to improve their visual quality and ensure that they convey information more effectively.

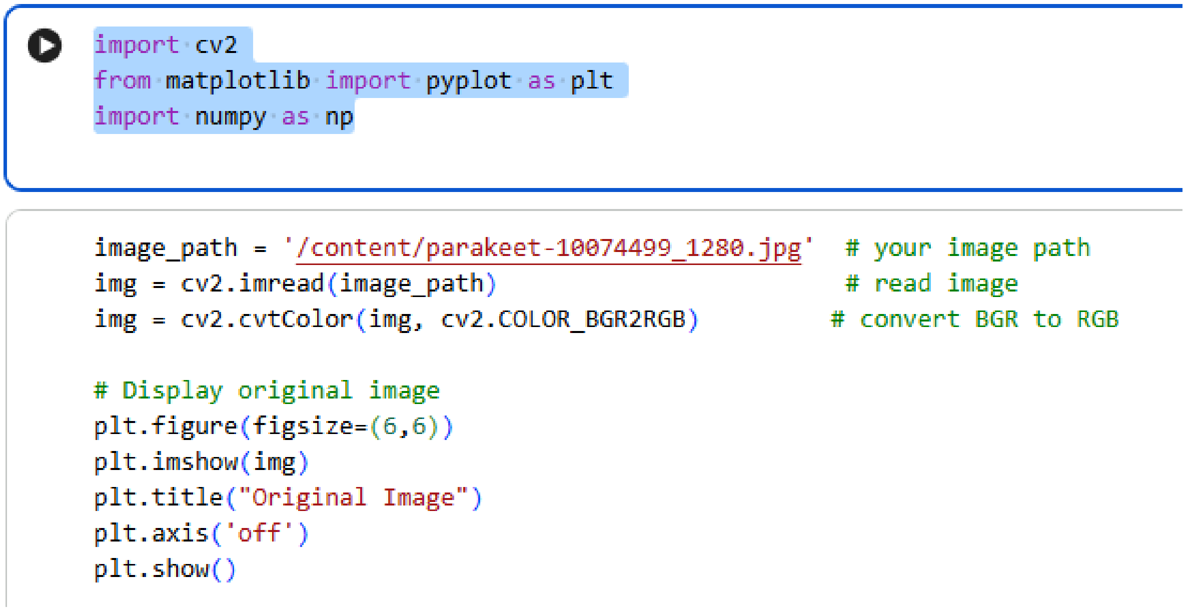
The main objectives of this project are to improve the clarity of images, enhance the color quality, and reduce blur by applying sharpening techniques. By enhancing the contrast and vibrancy of colors, the processed image becomes more visually appealing and informative. Another key objective is to compare the original image with the enhanced version to clearly demonstrate the improvements made.

This comparison helps in understanding the effectiveness of the enhancement techniques used in this project. Each objective in this project serves a specific purpose. Improving image clarity ensures that details are visible and the image can be effectively analyzed or appreciated.

Enhancing color quality makes the image more vibrant and visually appealing, which is particularly important in photography, education, and media applications. Reducing blur using sharpening techniques brings out the edges and fine details, making the image appear crisp and well-defined. Finally, comparing the original and enhanced images provides a clear demonstration of the effectiveness of the methods used, helping to understand how digital image processing can transform ordinary images into high-quality visuals.

## TOOLS & TECHNOLOGIES USED

This project was implemented using **Python**, a popular programming language for image processing. The **OpenCV** library was used to perform all image manipulation tasks, such as color enhancement and sharpening. **NumPy** was used for numerical operations, and **Matplotlib** was used to display images within the notebook. The project was executed on **Google Colab**, an online coding environment that allows running Python code without installing software on a local machine.



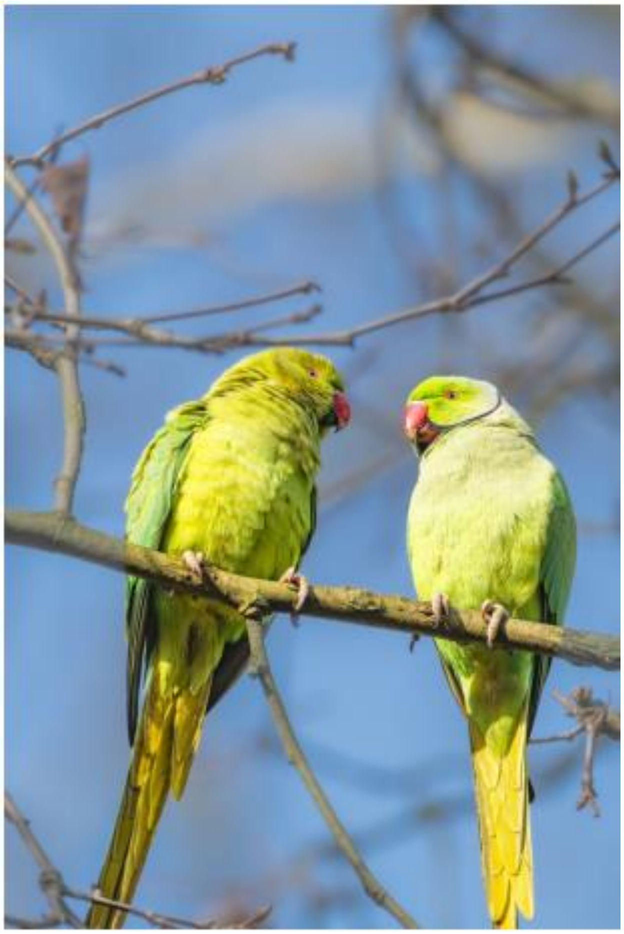
```python
import cv2
from matplotlib import pyplot as plt
import numpy as np
```

```python
image_path = '/content/parakeet-10074499_1280.jpg'  # your image path
img = cv2.imread(image_path)                        # read image
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)         # convert BGR to RGB

# Display original image
plt.figure(figsize=(6,6))
plt.imshow(img)
plt.title("Original Image")
plt.axis('off')
plt.show()
```

Google Colab was chosen because it allows running Python code online without installing anything locally. It also provides free access to powerful computing resources, making it convenient for processing images efficiently.

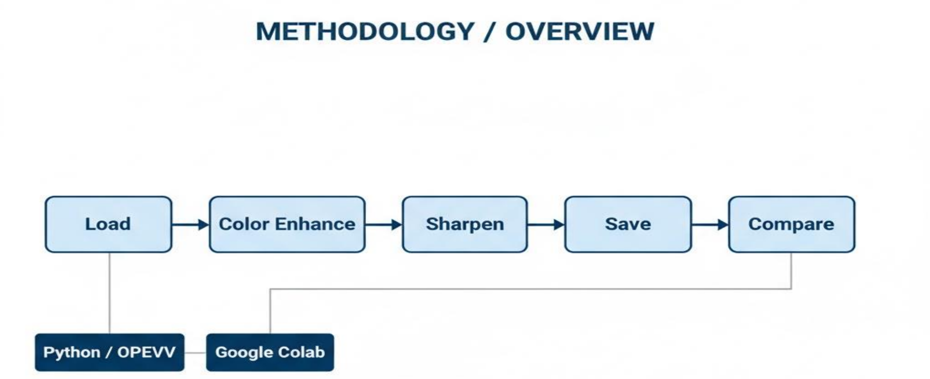
# IMAGE DESCRIPTION

The image used in this project shows **two colorful parrots sitting together**. It was downloaded from **Pixabay**, a free source of high-quality images. The image is in **JPG format** and has a resolution suitable for digital processing. This image was chosen because its vibrant colors and clear subject make it ideal for demonstrating color enhancement and sharpening techniques. Enhancing this image allows us to clearly observe the improvements in clarity, sharpness, and color vibrancy after processing.



# METHODOLOGY / OVERVIEW

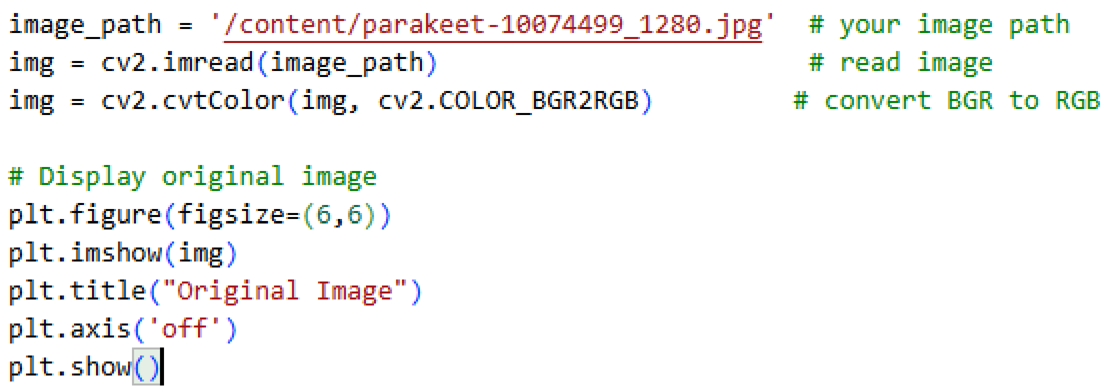
The project was carried out in a few systematic steps. First, the image was loaded into **Python** using **OpenCV** and displayed in **Google Colab**. Next, the color of the image was enhanced using the **LAB color space** and **CLAHE** (Contrast Limited Adaptive Histogram Equalization) to make the colors brighter and more vibrant. After color enhancement, the image was sharpened using a **Gaussian blur and unsharp masking technique** to make edges clear and details more prominent. Finally, the enhanced image was saved, and a side-by-side comparison with the original image was made to observe the improvements.



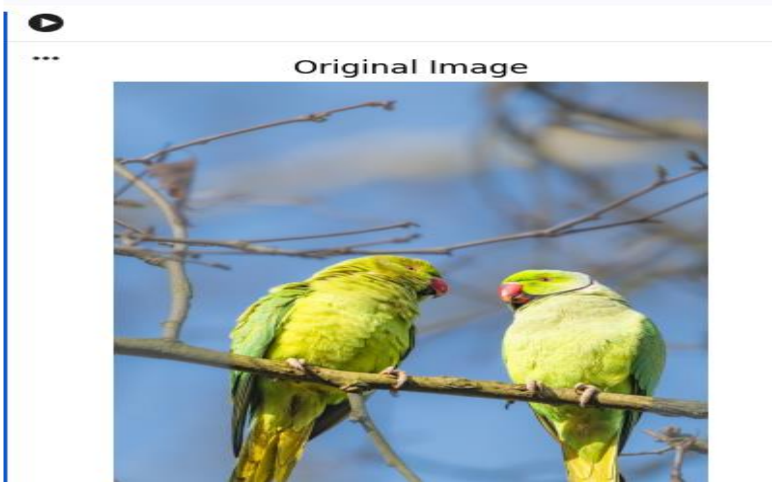
The diagram illustrates the sequence of operations applied to the image, from loading to final enhancement. It helps in understanding the workflow of the project clearly.

## STEP 1: IMAGE LOADING

The first step in this project was to load the original image into Python using the **OpenCV** library. OpenCV provides the function `cv2.imread()` to read images from a file path. Since OpenCV reads images in BGR format, the image was converted to **RGB** using `cv2.cvtColor()` to display the colors correctly. The image was then displayed using **Matplotlib** in Google Colab to ensure it was loaded properly before applying any enhancement techniques.

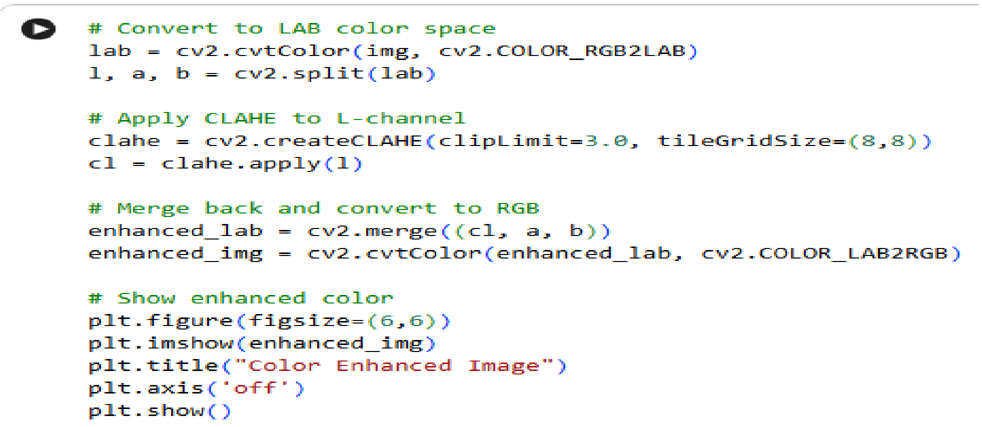


```
image_path = '/content/parakeet-10074499_1280.jpg'  # your image path
img = cv2.imread(image_path)                        # read image
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)         # convert BGR to RGB

# Display original image
plt.figure(figsize=(6,6))
plt.imshow(img)
plt.title("Original Image")
plt.axis('off')
plt.show()
```

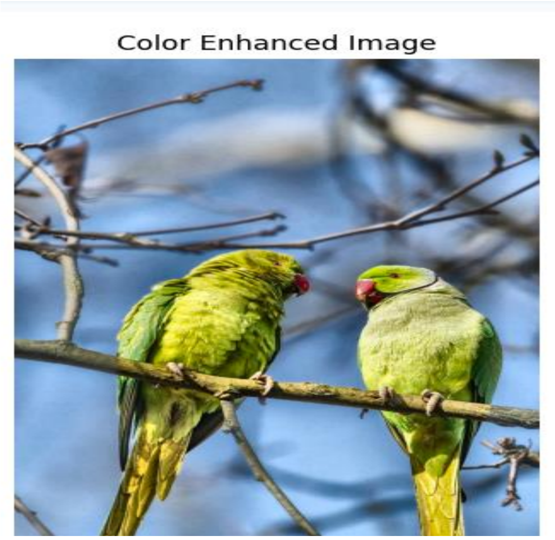


## STEP 2: COLOR ENHANCEMENT

After loading the image, the colors were enhanced to make the image more vibrant and visually appealing. The image was first converted to the **LAB color space**, which separates brightness from color information. **CLAHE (Contrast Limited Adaptive Histogram Equalization)** was applied to the L-channel to improve contrast without affecting colors. After merging the channels back and converting to RGB, the result was a brighter image with richer colors, while maintaining the original color balance.



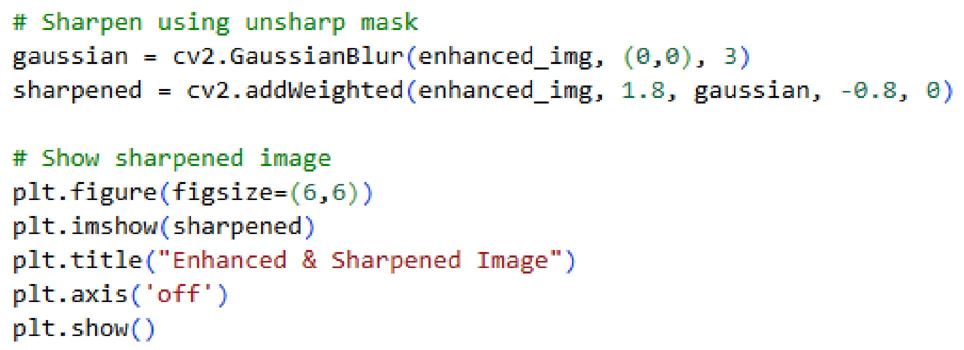
```
# Convert to LAB color space
lab = cv2.cvtColor(img, cv2.COLOR_RGB2LAB)
l, a, b = cv2.split(lab)

# Apply CLAHE to L-channel
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8,8))
cl = clahe.apply(l)

# Merge back and convert to RGB
enhanced_lab = cv2.merge((cl, a, b))
enhanced_img = cv2.cvtColor(enhanced_lab, cv2.COLOR_LAB2RGB)

# Show enhanced color
plt.figure(figsize=(6,6))
plt.imshow(enhanced_img)
plt.title("Color Enhanced Image")
plt.axis('off')
plt.show()
```



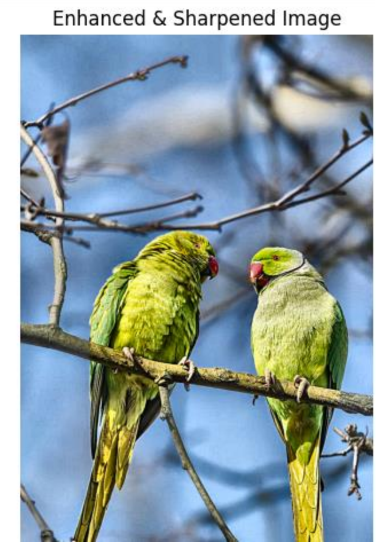
Color Enhanced Image

## STEP 3: IMAGE SHARPENING

After enhancing the colors, the image was sharpened to make the edges and details more clear. This was done using a **Gaussian blur** and **unsharp masking**. The blurred version of the image was subtracted from the enhanced image, and a weighted sum was applied to emphasize edges. This technique made the parrots' features crisp while preserving the enhanced colors. The sharpening process improves the visual quality and makes the details stand out more clearly.



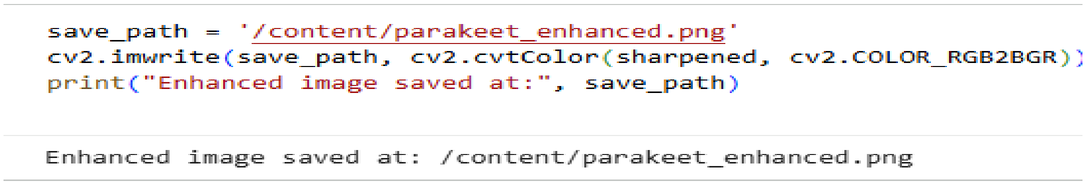
```
# Sharpen using unsharp mask
gaussian = cv2.GaussianBlur(enhanced_img, (0,0), 3)
sharpened = cv2.addWeighted(enhanced_img, 1.8, gaussian, -0.8, 0)

# Show sharpened image
plt.figure(figsize=(6,6))
plt.imshow(sharpened)
plt.title("Enhanced & Sharpened Image")
plt.axis('off')
plt.show()
```



## STEP 4: SAVING & COMPARISON

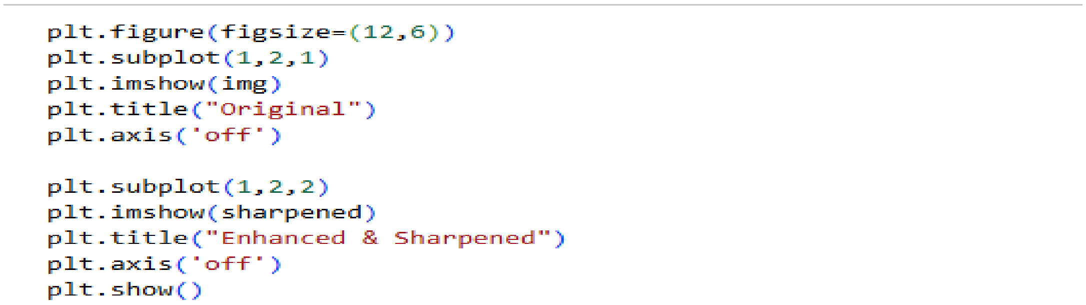
After the image was enhanced and sharpened, it was saved using the **`cv2.imwrite()`** function in OpenCV. Saving the output ensures that the enhanced image can be used later or shared without running the code again.

To evaluate the improvements, a **side-by-side comparison** of the original and the enhanced image was made using Matplotlib. This comparison clearly shows the difference in color vibrancy, clarity, and sharpness between the original and processed image, highlighting the effectiveness of the techniques applied in this project.

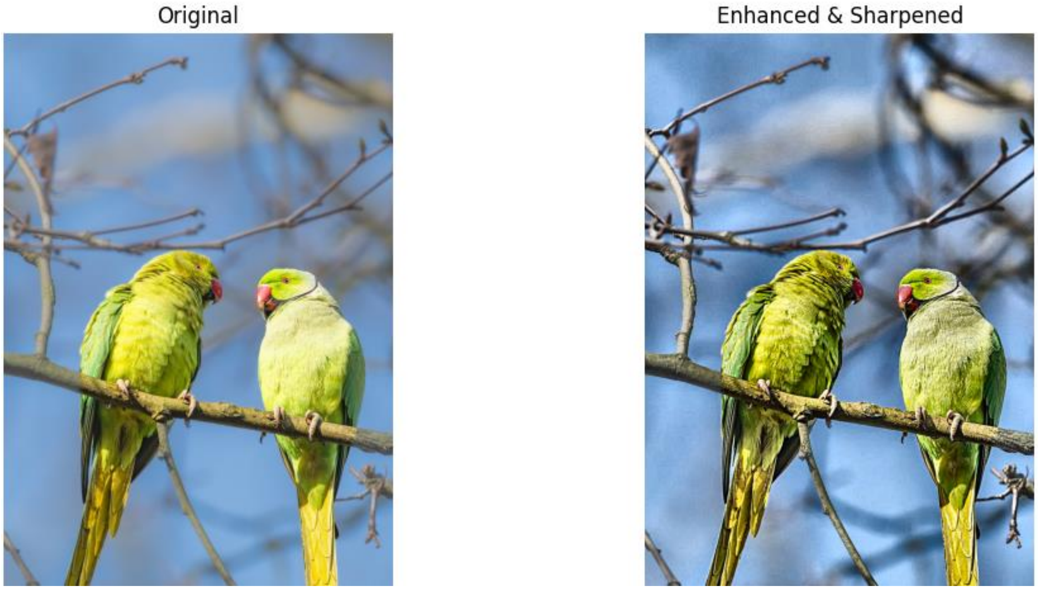


```
save_path = '/content/parakeet_enhanced.png'
cv2.imwrite(save_path, cv2.cvtColor(sharpened, cv2.COLOR_RGB2BGR))
print("Enhanced image saved at:", save_path)
```

```
Enhanced image saved at: /content/parakeet_enhanced.png
```



```
plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
plt.imshow(img)
plt.title("Original")
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(sharpened)
plt.title("Enhanced & Sharpened")
plt.axis('off')
plt.show()
```



Original

Enhanced & Sharpened

## RESULTS & DISCUSSION

The results of this project show a clear improvement in the visual quality of the image after applying enhancement techniques. The enhanced image appears brighter, with more vibrant and natural colors compared to the original image. The sharpening process significantly improved the clarity of the image by making edges and fine details more distinct.

The comparison between the original and enhanced images demonstrates that color enhancement and sharpening are effective methods for improving image quality. The dull appearance and slight blur present in the original image were reduced, resulting in a visually pleasing and clearer image. These results confirm that the techniques used in this project successfully achieved the intended objectives of improving clarity, color quality, and overall visual appearance.

## CONCLUSION

This project successfully demonstrated the application of digital image processing techniques for color image enhancement and sharpening. By using Python and the OpenCV library, the visual quality of the image was significantly improved. The enhanced image showed better color vibrancy, improved contrast, and sharper details compared to the original image.

Through this project, practical knowledge of image loading, color space conversion, contrast enhancement, and image sharpening was gained. The results confirm that simple image processing techniques can effectively enhance image quality and are useful in various real-world applications such as photography, media, and computer vision.