IMAGE ENHANCEMENT AND MANIPULATION

1. Changing the brightness and contrast of an image.
   a. Task to enhance underexposed images
2. Changing the hue and saturation (HSV) color model
   a. Task Colour Filtering and Colour Pop Effects

Image Processing

A general image processing operator is a function that takes one or more input images and produces an output image.

Image transforms can be seen as:

- Point operators (pixel transforms)
- Neighborhood (area-based) operators

Brightness and contrast adjustments

1. Two commonly used point processes are multiplication and addition with a constant:

   $g(x)=\alpha f(x)+\beta$

2. The parameters $\alpha>0$ and β are often called the gain and bias parameters; sometimes these parameters are said to control contrast and brightness respectively.
3. You can think of $f(x)$ as the source image pixels and $g(x)$ as the output image pixels. Then, more conveniently we can write the expression as:

   $g(i,j)=\alpha\cdot f(i,j)+\beta$

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow


LBImage = cv2.resize(cv2.imread('/content/drive/MyDrive/Computer Vision And image Processing/Images/LowBrightness_ima


cv2_imshow(LBImage)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
print("You want to lower the contrast or higher the contrast")
ans = int(input("0 for lower and 1 for higher"))
if ans == 0:
  alpha = float(input("enter the value of alpha between [0-1]: "))
else:
  alpha = float(input("enter the value of alpha >1 : "))

beta = float(input("enter the value of beta between [-127 to 127] (Lower the value lower the brightness): "))

AdjustedImage = cv2.convertScaleAbs(LBImage, alpha=alpha, beta=beta)
```

```
You want to lower the contrast or higher the contrast
0 for lower and 1 for higher1
enter the value of alpha >1 : 1.5
enter the value of beta between [-127 to 127] (Lower the value lower the brightness): 100
```

```
cv2_imshow(AdjustedImage)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

HSV

In computer vision and image processing, HSV (Hue, Saturation, Value) color space is a representation of colors that separates the intensity information from the color information. This color space is often used in image processing tasks such as color segmentation, object tracking, and image enhancement.

The HSV color space is represented by three components:

1. Hue (H): This component represents the color type, such as red, green, or blue. It is measured in degrees on a color wheel, with red at 0°, green at 120°, and blue at 240°.
2. Saturation (S): This component represents the purity of the color. A saturation value of 0 corresponds to a grayscale image, while a saturation value of 1 corresponds to a fully saturated color.
3. Value (V): This component represents the intensity or brightness of the color. A value of 0 corresponds to black, while a value of 1 corresponds to the brightest possible color.

Conversion

The conversion from RGB (Red, Green, Blue) color space to HSV color space can be done using the following formulas:

Given an RGB color (R, G, B) normalized to the range [0, 1], the corresponding HSV values (H, S, V) are calculated as follows:

1. Calculate the maximum and minimum values among R, G, and B: $C_{max} = max(R, G, B)$ $C_{min} = min(R, G, B)$

2. Calculate the Hue (H): $H = \begin{cases} 0 & \text{if } C_{max} = C_{min} \\ 60(\frac{G-B}{C_{max}-C_{min}}) \mod 360 & \text{if } C_{max} = R \\ 60(\frac{B-R}{C_{max}-C_{min}} + 2) \mod 360 & \text{if } C_{max} = G \\ 60(\frac{R-G}{C_{max}-C_{min}} + 4) \mod 360 & \text{if } C_{max} = B \end{cases}$
3. Calculate the Saturation (S): $S = \begin{cases} 0 & \text{if } C_{max} = 0 \\ \frac{C_{max}-C_{min}}{C_{max}} & \text{otherwise} \end{cases}$
4. Calculate the Value (V): $V = C_{max}$

Color filtering

Color filtering in computer vision and image processing (CVIP) is a technique used to isolate specific colors or color ranges in an image for further analysis or processing. This process involves setting up a color filter to extract only the desired colors while removing or suppressing others.

The steps involved in color filtering in CVIP are as follows:

1. Color Space Conversion: The image is typically converted from the RGB color space to another color space that facilitates color manipulation, such as HSV or YUV.
2. Thresholding: Thresholding is applied to the color components (such as hue, saturation, value, or luminance) to define the range of colors to be filtered.
3. Filtering Operation: Pixels that fall within the specified color range are retained, while pixels outside the range are either set to a specific value or removed.
4. Post-Processing: Additional image processing techniques can be applied to the filtered image, such as morphological operations, contour detection, or object tracking.

Color pop effects

Color pop effects in computer vision and image processing (CVIP) are visual enhancements applied to images to selectively highlight or emphasize certain colors while desaturating or converting the rest of the image to grayscale. This effect creates a striking contrast between the highlighted colors and the background, making the selected colors stand out more prominently.

The steps involved in creating a color pop effect in CVIP are as follows:

1. **Color Selection**: Choose the color or colors that you want to emphasize in the image. This can be done manually by selecting specific hues or using color segmentation techniques to automatically identify the colors of interest.
2. **Color Isolation**: Isolate the selected colors by creating a mask that identifies pixels corresponding to the chosen colors. This mask will be used to preserve the selected colors while desaturating the rest of the image.
3. **Desaturation**: Convert the image to grayscale or reduce the saturation of all colors except the selected colors. This step helps in enhancing the contrast between the highlighted colors and the background.
4. **Blending**: Blend the desaturated image with the original image using the mask created in the color isolation step. This blending process ensures that the selected colors remain vibrant while the rest of the image appears in grayscale.
5. **Adjustments**: Fine-tune the color pop effect by adjusting the intensity of the highlighted colors, the level of desaturation, and other parameters to achieve the desired visual impact.

Color pop effects are commonly used in photography, graphic design, and image editing applications to draw attention to specific elements in an image, create artistic effects, or enhance the overall visual appeal. By selectively highlighting colors, color pop effects can make images more visually appealing and engaging to viewers.

```
BGRimage = cv2.resize(cv2.imread('/content/drive/MyDrive/Computer Vision And image Processing/Images/CoupleImage.jpg'
```

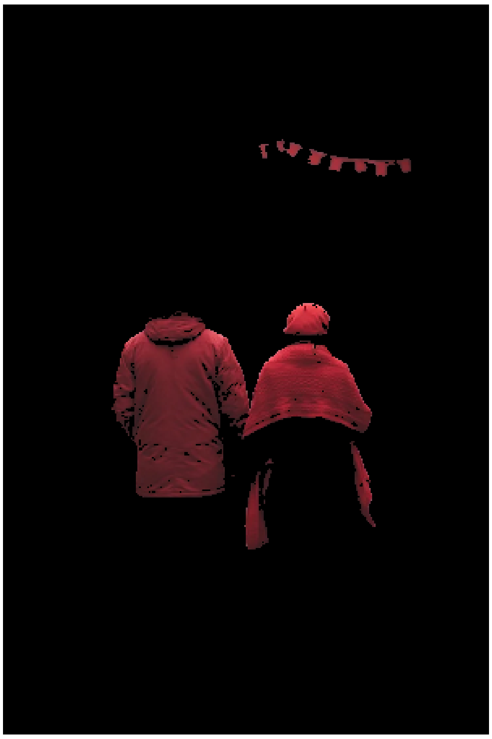
```
cv2_imshow(BGRimage)
```

```
hsv = cv2.cvtColor(BGRimage, cv2.COLOR_BGR2HSV)
gray = cv2.cvtColor(BGRimage, cv2.COLOR_BGR2GRAY)

lower_red = np.array([160,100,50])
upper_red = np.array([180,255,255])

mask = cv2.inRange(hsv, lower_red, upper_red)
mask_inverse = cv2.bitwise_not(mask)


res = cv2.bitwise_and(BGRimage, BGRimage, mask=mask)
cv2_imshow(res)
```



```
background = cv2.bitwise_and(gray, gray, mask = mask_inverse)

background = np.stack((background,)*3, axis=-1)

added_img = cv2.add(res, background)
cv2_imshow(added_img)
```