SUPERIOR UNIVERSITY

Programming For Artificial Intelligence (Lab)

Assignment - 5

Name:

Ali Maqsood.

Roll no:

SU92-BSAIM-F23-050.

Department:

Software Engineering Department.

Program:

Artificial Intelligence.

Section:

BSAI-4A

Question # 1:

Implement the main OpenCV functions in a Jupyter Notebook (.ipynb) (main topics, can skip others)

Code:

```
# Importing Libaries
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Reading image
image_path = 'img1.jpg'
image = cv2.imread(image_path)

### Showing image
plt.imshow(image)
plt.title('Original Image')
plt.axis('off')
plt.show()

### Blurring image
bilateral = cv2.bilateralFilter(image, 15, 150, 150)
bilateral_rgb = cv2.cvtColor(bilateral, cv2.COLOR_BGR2RGB)
plt.imshow(bilateral_rgb)
plt.title('Bilateral Blurred Image')
plt.axis('off')
plt.show()
```

```
### Gray scaling image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow('Grayscale', gray_image)
cv2.waitKey(0)
cv2.destroyAllWindows()

### Edge detection
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
edges = cv2.Canny(image_rgb, 100, 700)
fig, axs = plt.subplots(1, 2, figsize=(7, 4))
axs[0].imshow(image_rgb), axs[0].set_title('Original Image')
axs[1].imshow(edges), axs[1].set_title('Image Edges')
for ax in axs:
    ax.set_xticks([]), ax.set_yticks([])
plt.tight_layout()
plt.show()

### Transformaton into gamma_corrected
gamma_corrected = np.array(255*(image / 255) ** 1.2, dtype = 'uint8')
plt.imshow(gamma_corrected)
plt.title('gamma_corrected Image')
plt.axis('off')
plt.show()
cv2.imshow(gamma_corrected)

### Equalizing image histogram
img = cv2.imread('img1.jpg', 0)
```

```
equ = cv2.equalizeHist(img)
res = np.hstack((img, equ))
plt.figure(figsize=(10, 5))
plt.imshow(res, cmap='gray')
plt.title("Original vs Equalized Image")
plt.axis('off')
plt.show()

### CV2 color method
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("Grayscale Image", gray_image)
cv2.waitKey(0)
cv2.destroyAllWindows()

### Looking images in different color spaces
img = cv2.cvtColor(image, cv2.COLOR_BGR2YCrCb)
cv2.imshow('image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Documentation:

Following is the explanation of each part of code that is shown above:

1. We import **OpenCV** for image processing, **NumPy** for a few mathematical operations, and **Matplotlib** for displaying images.
2. Reads the image from the given path (img1.jpg) into an array format using OpenCV.
3. Displays the original image using Matplotlib.
4. Applies a bilateral filter to smooth the image while keeping edges sharp.
5. Converts the colour image into a grayscale version and displays it in an OpenCV window.
6. Uses the Canny edge detector to identify strong edges and boundaries in the image. Both the original and edge-detected images are displayed side by side.
7. Performs gamma correction to adjust brightness and contrast.
8. Uses histogram equalization to improve image contrast.
9. Again, converts the image to grayscale using OpenCV's color conversion method.
10. Transforms the image from BGR to YCrCb color space.