**Assignment #1 Python Platform & Basic Image Processing**

P68131509 林均有

1. **Setup Python coding environment (Google Colab or your Laptop/PC).**

I chose VSCode as my code editor, Anaconda to set up my Python coding environment, and Jupyter Notebook for interactive computing, which offers interactive visualizations and real-time feedback.

1. **Explain and report the codes in Part A and the codes before Part A**

**Before Part A:**

Imports: Bring in tools for image processing (cv2), displaying images (matplotlib), and managing files (os).

os.chdir(): Sets the working folder to where my images are, making it easier to work with them in the script.

**PartA:**

cv2.imread("Lenna.png"): Reads the image file "Lenna.png" into the img variable. plt.figure(figsize=(4,4)) set the figure size in inches.

plt.imshow(img): Displays the image. Since OpenCV uses BGR. plt.title("Lenna orig"): Sets the title of the plot. plt.show(): Displays the plot.

cv2.cvtColor(img, cv2.COLOR\_BGR2RGB): Converts the image from BGR to RGB format so it displays correctly. cv2.COLOR\_BGR2RGB is a color conversion code which can convert an image from BGR color space to RGB color space. print(type(img)) and print(img.shape) shows the data type of current image and the dimensions of an image.

plt.subplots(1, 3, figsize=(12, 4)): Creates a figure and a grid of subplots with 1 row and 3 columns. img[:, :, 0]: Extracts the Red channel from the image array. In OpenCV, color images are represented in BGR format. cmap='gray': Displays the image in grayscale. This is used because each channel is a single grayscale image representing the intensity of that color channel. ax\_arr[0].set\_title("Red-channel image"): Sets the title for the first subplot, indicating that it shows the Red channel. vmin=0 and vmax=255 specify the minimum and maximum values for the color map scale. This ensures that the entire range of pixel intensities (0 to 255) is used.

1. **Implement the functions of image thresholding and image histogram by using  
   OpenCV in Part C with the aids of Germini or ChatGPT.**

img\_gray = cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)  # Convert to grayscale for thresholding

\_, img\_thresh = cv2.threshold(img\_gray, 127, 255, cv2.THRESH\_BINARY)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

plt.imshow(img\_gray, cmap='gray')

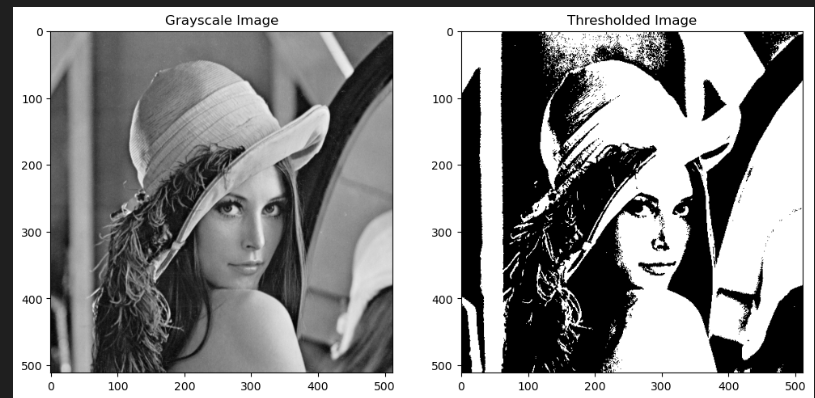
plt.title("Grayscale Image")

plt.subplot(1, 2, 2)

plt.imshow(img\_thresh, cmap='gray')

plt.title("Thresholded Image")

plt.show()



# image histogram

hist=cv2.calcHist([img\_gray], [0], None, [256], [0, 256])

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

plt.imshow(img\_gray, cmap='gray')

plt.title("Grayscale Image")

plt.subplot(1, 2, 2)

plt.plot(hist)

plt.title("Grayscale Histogram")

plt.xlabel("Bins")

plt.ylabel("# of Pixels")

plt.xlim([0, 256])

plt.tight\_layout()

plt.show()

