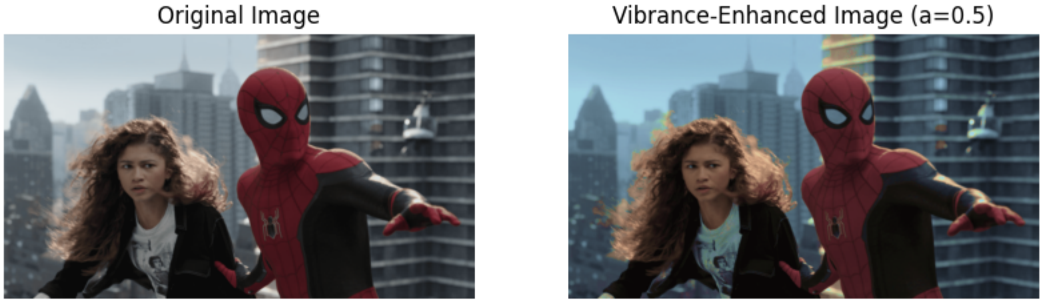
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

from google.colab import drive
drive.mount('/content/drive')

image = cv.imread("/content/drive/MyDrive/images/spider.png", cv.IMREAD_COLOR)

assert image is not None

# Convert the image from BGR to HSV color space
hsv_image = cv.cvtColor(image, cv.COLOR_BGR2HSV)

# Split the image into hue, saturation, and value planes
hue, saturation, value = cv.split(hsv_image)

# Define the intensity transformation function
def intensity_transformation(x, a=0.5, sigma=70):
    return np.clip(a * x + a * 128 * np.exp(-((x - 128) ** 2) / (2 * sigma ** 2)), 0, 255).astype(np.uint8)

# Apply the intensity transformation to the saturation plane
a = 0.5  # You can adjust this value as needed
enhanced_saturation = intensity_transformation(saturation, a=a)

# Recombine the channels
enhanced_hsv_image = cv.merge([hue, enhanced_saturation, value])

# Convert the enhanced HSV image back to BGR color space
enhanced_image = cv.cvtColor(enhanced_hsv_image, cv.COLOR_HSV2BGR)

# Display the original and enhanced images
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(cv.cvtColor(image, cv.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cv.cvtColor(enhanced_image, cv.COLOR_BGR2RGB))
plt.title('Vibrance-Enhanced Image (a={})'.format(a))
plt.axis('off')

plt.show()
```

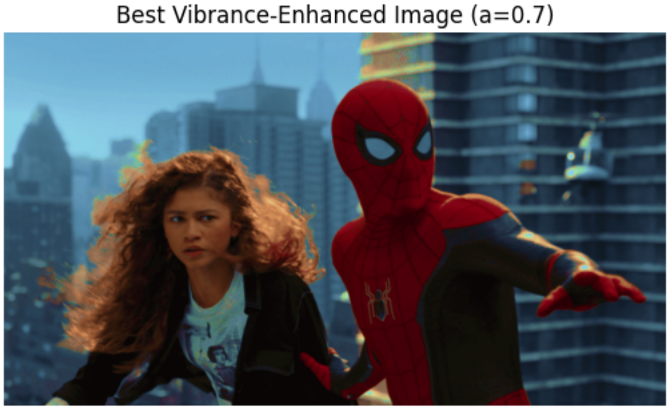
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```



Part c

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

from google.colab import drive
drive.mount('/content/drive')

image = cv.imread("/content/drive/MyDrive/images/spider.png", cv.IMREAD_COLOR)

assert image is not None

# Convert the image from BGR to HSV color space
hsv_image = cv.cvtColor(image, cv.COLOR_BGR2HSV)

# Split the image into hue, saturation, and value planes
hue, saturation, value = cv.split(hsv_image)

# Define the intensity transformation function
def intensity_transformation(x, a=0.5, sigma=70):
    return np.clip(x + a * 128 * np.exp(-((x - 128) ** 2) / (2 * sigma ** 2)), 0, 255).astype(np.uint8)

# Adjust 'a' to find a visually pleasing output
a_values = [0.7, 0.8, 0.9]
best_a = None
best_score = float('-inf')
best_enhanced_image = None

for a in a_values:
    # Apply the intensity transformation to the saturation plane
    enhanced_saturation = intensity_transformation(saturation, a=a)

    # Recombine the enhanced saturation with the original hue and value channels
    enhanced_hsv_image = cv.merge([hue, enhanced_saturation, value])

    # Convert the enhanced HSV image back to BGR color space
    enhanced_image = cv.cvtColor(enhanced_hsv_image, cv.COLOR_HSV2BGR)



    # Update best_a if the current score is better
    if score > best_score:
        best_score = score
        best_a = a
        best_enhanced_image = enhanced_image

# Report the best value of 'a'
print("Best value of 'a':", best_a)


# Display the best enhanced image


plt.imshow(cv.cvtColor(best_enhanced_image, cv.COLOR_BGR2RGB))
plt.title('Best Vibrance-Enhanced Image (a={})'.format(best_a))
plt.axis('off')
plt.show()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
Best value of 'a': 0.7
```



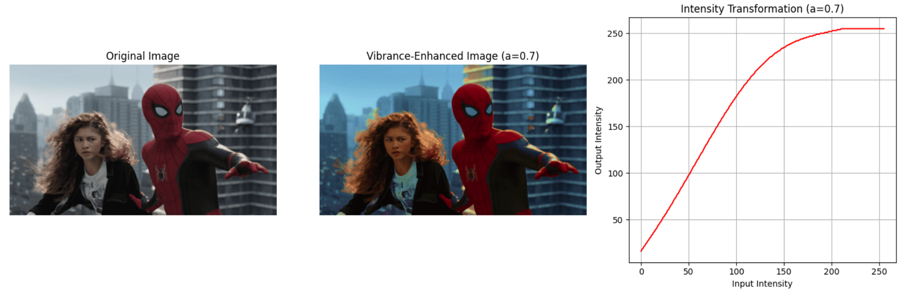
Best Vibrance-Enhanced Image (a=0.7)

Part d

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

from google.colab import drive
drive.mount('/content/drive')

image = cv.imread("/content/drive/MyDrive/images/spider.png", cv.IMREAD_COLOR)

assert image is not None

# Convert the image from BGR to HSV color space
hsv_image = cv.cvtColor(image, cv.COLOR_BGR2HSV)

# Split the image into hue, saturation, and value planes
hue, saturation, value = cv.split(hsv_image)

# Define the intensity transformation function
def intensity_transformation(x, a=0.5, sigma=70):
    return np.clip(x + a * 128 * np.exp(-((x - 128) ** 2) / (2 * sigma ** 2)), 0, 255).astype(np.uint8)

# Apply the intensity transformation to the saturation plane
a = 0.7
enhanced_saturation = intensity_transformation(saturation, a=a)

# Recombine the enhanced saturation with the original hue and value channels
enhanced_hsv_image = cv.merge([hue, enhanced_saturation, value])

# Convert the enhanced HSV image back to BGR color space
enhanced_image = cv.cvtColor(enhanced_hsv_image, cv.COLOR_HSV2BGR)

# Display the original image, the vibrance-enhanced image, and the intensity transformation
plt.figure(figsize=(15, 5))

# Original Image
plt.subplot(1, 3, 1)
plt.imshow(cv.cvtColor(image, cv.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

# Vibrance-Enhanced Image
plt.subplot(1, 3, 2)
plt.imshow(cv.cvtColor(enhanced_image, cv.COLOR_BGR2RGB))
plt.title('Vibrance-Enhanced Image (a={})'.format(a))
plt.axis('off')

# Intensity Transformation
x = np.arange(0, 256)
plt.subplot(1, 3, 3)
plt.plot(x, intensity_transformation(x, a=a), color='r')
plt.title('Intensity Transformation (a={})'.format(a))
plt.xlabel('Input Intensity')
```

```
    plt.ylabel('Output Intensity')
    plt.grid(True)

    plt.tight_layout()
    plt.show()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
```



Original Image

Vibrance-Enhanced Image (a=0.7)

Intensity Transformation (a=0.7)