IVP LAB

HISTOGRAM EQUALIZATION

Krithika
22011101046

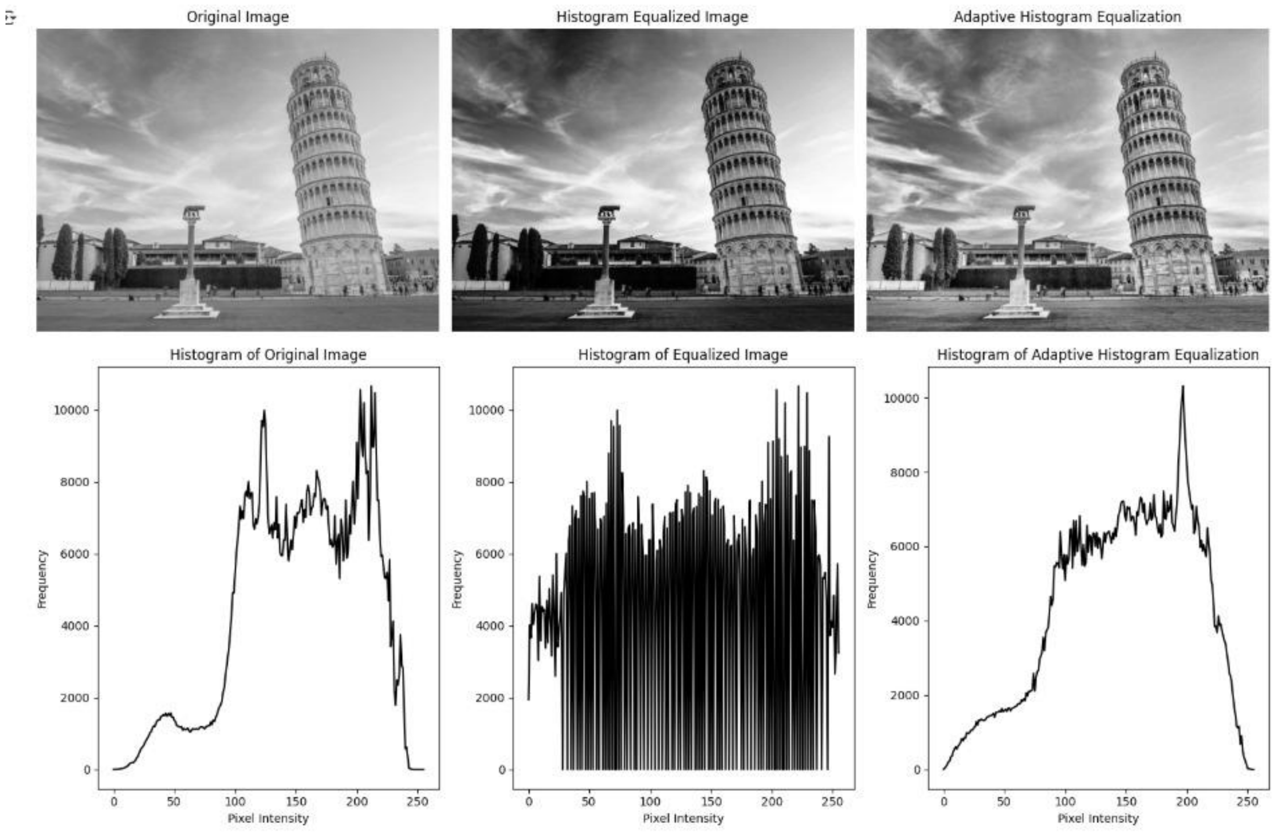
1. GRAYSCALE IMAGE

CODE:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files

# Step 1: Upload and Read Image
uploaded = files.upload()  # Upload your image
image = cv2.imread("/content/tower_of_pisa_image.jpg", cv2.IMREAD_GRAYSCALE)  # Read the image in
grayscale

# Step 2: Histogram Equalization
equalized_image = cv2.equalizeHist(image)

# Step 3: Adaptive Histogram Equalization (CLAHE)
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
adaptive_image = clahe.apply(image)

# Step 4: Display Images
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(image, cmap='gray')
plt.title("Original Image")
plt.axis("off")

plt.subplot(1, 3, 2)
plt.imshow(equalized_image, cmap='gray')
plt.title("Histogram Equalized Image")
plt.axis("off")

plt.subplot(1, 3, 3)
plt.imshow(adaptive_image, cmap='gray')
plt.title("Adaptive Histogram Equalization")
plt.axis("off")

plt.tight_layout()
plt.show()

# Step 5: Histogram Plots
plt.figure(figsize=(15, 6))

# Histogram of Original Image
plt.subplot(1, 3, 1)
hist = cv2.calcHist([image], [0], None, [256], [0, 256])
```

```
plt.plot(hist, color='black')
plt.title('Histogram of Original Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

# Histogram of Equalized Image
plt.subplot(1, 3, 2)
hist = cv2.calcHist([equalized_image], [0], None, [256], [0, 256])
plt.plot(hist, color='black')
plt.title('Histogram of Equalized Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

# Histogram of Adaptive Histogram Equalized Image
plt.subplot(1, 3, 3)
hist = cv2.calcHist([adaptive_image], [0], None, [256], [0, 256])
plt.plot(hist, color='black')
plt.title('Histogram of Adaptive Histogram Equalization')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

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

OUTPUT:

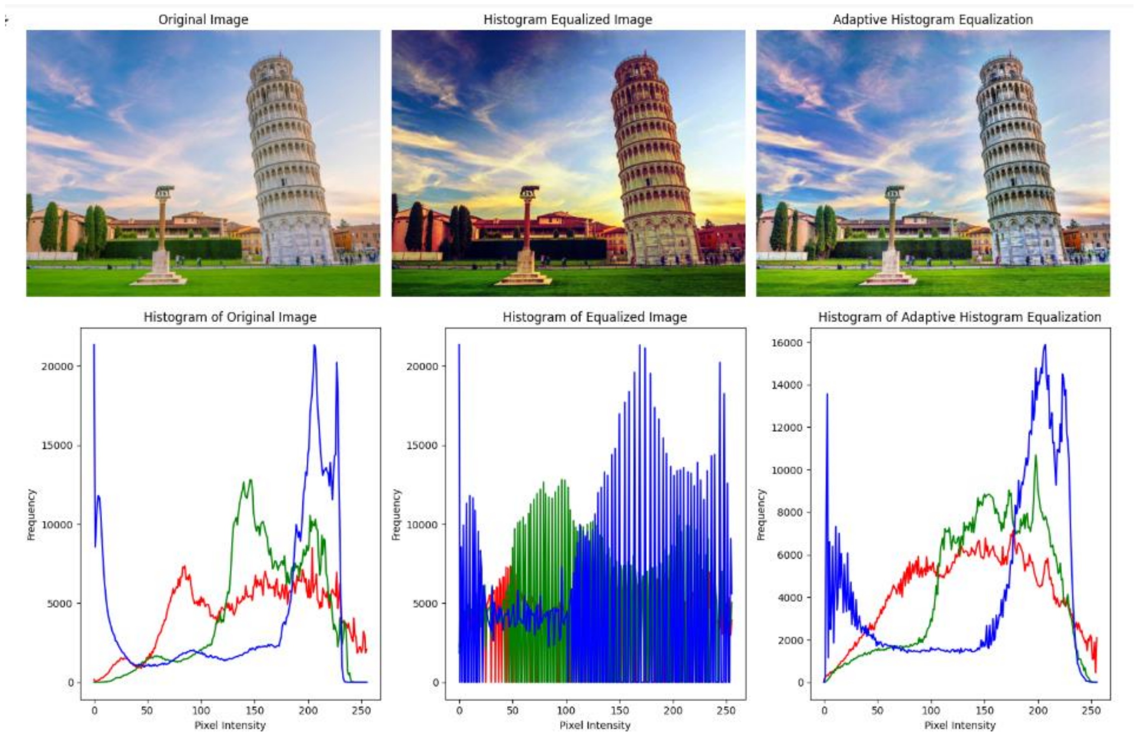


2. RGB IMAGE

CODE:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files

# Step 1: Upload and Read Image
uploaded = files.upload()  # Upload your image
image = cv2.imread("/content/tower_of_pisa_image.jpg")  # Read the image
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  # Convert BGR to RGB

# Step 2: Histogram Equalization
r, g, b = cv2.split(image)
equalized_r = cv2.equalizeHist(r)
equalized_g = cv2.equalizeHist(g)
equalized_b = cv2.equalizeHist(b)
equalized_image = cv2.merge([equalized_r, equalized_g, equalized_b])

# Step 3: Adaptive Histogram Equalization (CLAHE)
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
adaptive_r = clahe.apply(r)
adaptive_g = clahe.apply(g)
adaptive_b = clahe.apply(b)
adaptive_image = cv2.merge([adaptive_r, adaptive_g, adaptive_b])

# Step 4: Display Images
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(image)
plt.title("Original Image")
plt.axis("off")

plt.subplot(1, 3, 2)
plt.imshow(equalized_image)
plt.title("Histogram Equalized Image")
plt.axis("off")

plt.subplot(1, 3, 3)
plt.imshow(adaptive_image)
plt.title("Adaptive Histogram Equalization")
plt.axis("off")

plt.tight_layout()
plt.show()

# Step 5: Histogram Plots
colors = ('r', 'g', 'b')
plt.figure(figsize=(15, 6))

# Histogram of Original Image
plt.subplot(1, 3, 1)
for i, col in enumerate(colors):
  hist = cv2.calcHist([image], [i], None, [256], [0, 256])
  plt.plot(hist, color=col)
plt.title('Histogram of Original Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

# Histogram of Equalized Image
plt.subplot(1, 3, 2)
for i, col in enumerate(colors):
    hist = cv2.calcHist([equalized_image], [i], None, [256], [0, 256])
    plt.plot(hist, color=col)
plt.title('Histogram of Equalized Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

# Histogram of Adaptive Histogram Equalized Image
plt.subplot(1, 3, 3)
for i, col in enumerate(colors):
    hist = cv2.calcHist([adaptive_image], [i], None, [256], [0, 256])
    plt.plot(hist, color=col)
plt.title('Histogram of Adaptive Histogram Equalization')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')

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

OUTPUT:


Original Image
Histogram Equalized Image
Adaptive Histogram Equalization
Histogram of Original Image
Histogram of Equalized Image
Histogram of Adaptive Histogram Equalization
Frequency
Pixel Intensity