import numpy as np  
import cv2  
import matplotlib.pyplot as plt  
  
# Define the size of the grids and the whole image  
grid\_size = 100  
image\_size = 400  
  
# Separate list of 16 numbers representing pixel values for each grid  
pixel\_values = [125, 0, 125, 255, 0, 125, 255, 125, 125, 255, 125, 0, 255, 125, 0, 125]  
  
# Create a blank image  
image = np.zeros((image\_size, image\_size), dtype=np.uint8)  
  
# Iterate through each grid and assign the specified pixel value  
for i in range(4):  
 for j in range(4):  
 grid\_index = i \* 4 + j  
 pixel\_value = pixel\_values[grid\_index]  
 image[i\*grid\_size:(i+1)\*grid\_size, j\*grid\_size:(j+1)\*grid\_size] = pixel\_value  
  
# Create a new image with specific grid colors  
new\_image = np.zeros((image\_size, image\_size), dtype=np.uint8)  
  
for i in range(4):  
 for j in range(4):  
 grid\_index = i \* 4 + j  
 if grid\_index in [2, 8]: # Grids 3 and 9 are black  
 pixel\_value = 0  
 elif grid\_index in [5, 6, 9, 10]: # Grids 6, 7, 10, 11 are white  
 pixel\_value = 255  
 else: # Rest all are gray  
 pixel\_value = 125  
 new\_image[i\*grid\_size:(i+1)\*grid\_size, j\*grid\_size:(j+1)\*grid\_size] = pixel\_value  
  
# Calculate and plot histograms for old and new images  
hist\_old = cv2.calcHist([image], [0], None, [256], [0,256])  
hist\_new = cv2.calcHist([new\_image], [0], None, [256], [0,256])  
  
plt.figure(figsize=(12, 6))  
  
# Plot old image  
plt.subplot(2, 3, 1)  
plt.imshow(image, cmap='gray')  
plt.title('Old Image')  
plt.axis('off')  
  
# Plot histogram of old image  
plt.subplot(2, 3, 2)  
plt.plot(hist\_old, color='black')  
plt.title('Histogram (Old Image)')  
plt.xlabel('Pixel Value')  
plt.ylabel('Frequency')  
  
# Plot histogram equalization for old image  
plt.subplot(2, 3, 3)  
equalized\_old = cv2.equalizeHist(image)  
plt.imshow(equalized\_old, cmap='gray')  
plt.title('Equalized Old Image')  
plt.axis('off')  
  
# Plot new image  
plt.subplot(2, 3, 4)  
plt.imshow(new\_image, cmap='gray')  
plt.title('New Image')  
plt.axis('off')  
  
# Plot histogram of new image  
plt.subplot(2, 3, 5)  
plt.plot(hist\_new, color='black')  
plt.title('Histogram (New Image)')  
plt.xlabel('Pixel Value')  
plt.ylabel('Frequency')  
  
# Plot histogram equalization for new image  
plt.subplot(2, 3, 6)  
equalized\_new = cv2.equalizeHist(new\_image)  
plt.imshow(equalized\_new, cmap='gray')  
plt.title('Equalized New Image')  
plt.axis('off')  
  
plt.tight\_layout()  
plt.show()

import cv2  
import numpy as np  
from matplotlib import pyplot as plt  
  
# Create a 4x4 image with grayscale colors  
image = np.ones((4, 4), dtype=np.uint8) \* 255 # Initialize with white  
image[1:3, 2:4] = 0 # Set 6,7,10,11 boxes to black  
  
# Convert to OpenCV format  
image = cv2.cvtColor(image, cv2.COLOR\_GRAY2BGR)  
  
# Generate histogram for original image  
plt.subplot(121)  
plt.hist(image.ravel(), 256, [0, 256])  
plt.title('Original Image Histogram')  
plt.xlabel('Pixel Intensity')  
plt.ylabel('Frequency')  
  
# Convert specified boxes to black  
image[1:3, 2:4] = [0, 0, 0] # Set 6,7,10,11 boxes to black  
  
# Generate histogram for modified image  
plt.subplot(122)  
plt.hist(image.ravel(), 256, [0, 256])  
plt.title('Converted Image Histogram')  
plt.xlabel('Pixel Intensity')  
plt.ylabel('Frequency')  
  
# Display images and histograms  
plt.show()