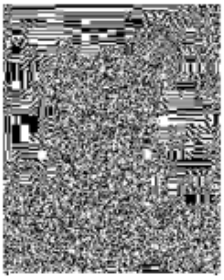
```
# -*- coding: utf-8 -*-
"""
Created on Thu Aug 22 09:17:09 2024

@author: dhanu
"""



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

# Load the image in grayscale
image = cv2.imread(r'C:/Users/dhanu/OneDrive/Pictures/1.jpg', cv2.IMREAD_GRAYSCALE)

# Define the intensity range to slice
lower_bound = 130
upper_bound = 230

# Create a copy of the image for slicing
sliced_image = np.zeros_like(image)

# Apply intensity level slicing
sliced_image[(image >= lower_bound) & (image <= upper_bound)] = image[(image >= lower_bound)
& (image <= upper_bound)]

# Alternatively, set pixels outside the range to a specific value, e.g., 0 or 255
# sliced_image[(image < lower_bound) | (image > upper_bound)] = 0  # set to black
# sliced_image[(image < lower_bound) | (image > upper_bound)] = 255  # set to white
```

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

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(image, cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title('Sliced Image')
plt.imshow(sliced_image, cmap='gray')
plt.axis('off')

plt.show()
```

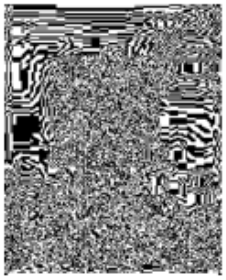
```
# -*- coding: utf-8 -*-
"""
Created on Thu Aug 22 20:56:53 2024

@author: dhanu
"""
import cv2
import numpy as np
import matplotlib.pyplot as plt
image = cv2.imread(r'C:\Users\dhanu\OneDrive\Desktop\dog.jpeg', cv2.IMREAD_GRAYSCALE)
rows, cols = image.shape
bit_planes = []
for i in range(8):
    bit_plane = (image & (1 << i)) >> i
    bit_plane = bit_plane * 255
    bit_planes.append(bit_plane)
plt.figure(figsize=(12, 6))
plt.subplot(2, 4, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.axis('off')
for i in range(8):
    plt.subplot(2, 4, i + 2)
    plt.imshow(bit_planes[i], cmap='gray')
    plt.title(f'Bit Plane {i}')
    plt.axis('off')
plt.tight_layout()
plt.show()
```

Original Image

Bit Plane 0



Bit Plane 1



Bit Plane 2

Bit Plane 3

Bit Plane 4

Bit Plane 5

Bit Plane 6