

Image Bit Slicing

Lab4 Bit Slicing.ipynb • Lab4 Image Pyramid.ipynb • Lab4 Quantization.ipynb • Untitled.ipynb • Lab4 Sampling.ipynb

Lab 4 > Lab4_CSE457 > Lab4 Bit Slicing.ipynb > lst = []

+ Code + Markdown | ▶ Run All ↺ Restart ≡ Clear All Outputs | 📄 Variables ≡ Outline ... Python 3.12.3

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

[21] ✓ 0.0s Python

```
image = cv2.imread('pic.jpg',0)
window_name = 'img'
cv2.imshow(window_name,image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[22] ✓ 14.0s Python

```
lst = []
for i in range(image.shape[0]):
    for j in range(image.shape[1]):
        lst.append(np.binary_repr(image[i][j],width=8)) #width = no. of bits
```

[23] ✓ 0.1s Python

```
eight_bit_img = (np.array([int(i[0]) for i in lst],dtype = np.uint8)*128).reshape(image.shape[0],image.shape[1])
seven_bit_img = (np.array([int(i[1]) for i in lst],dtype = np.uint8)*64).reshape(image.shape[0],image.shape[1])
six_bit_img = (np.array([int(i[2]) for i in lst],dtype = np.uint8)*32).reshape(image.shape[0],image.shape[1])
five_bit_img = (np.array([int(i[3]) for i in lst],dtype = np.uint8)*16).reshape(image.shape[0],image.shape[1])
four_bit_img = (np.array([int(i[4]) for i in lst],dtype = np.uint8)*8).reshape(image.shape[0],image.shape[1])
three_bit_img = (np.array([int(i[5]) for i in lst],dtype = np.uint8)*4).reshape(image.shape[0],image.shape[1])
two_bit_img = (np.array([int(i[6]) for i in lst],dtype = np.uint8)*2).reshape(image.shape[0],image.shape[1])
one_bit_img = (np.array([int(i[7]) for i in lst],dtype = np.uint8)*1).reshape(image.shape[0],image.shape[1])
```

[24] ✓ 0.1s Python

```
finalr = cv2.hconcat([eight_bit_img,seven_bit_img,six_bit_img,five_bit_img])
finalv = cv2.hconcat([four_bit_img,three_bit_img,two_bit_img,one_bit_img])
final = cv2.vconcat([finalr,finalv])
```

[25] ✓ 0.0s Python

```
window_name = 'Final'
cv2.imshow(window_name,final)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[26] ✓ 13.9s Python

```
newImage = eight_bit_img+seven_bit_img+six_bit_img+five_bit_img
window_name = 'Created Image'
cv2.imshow(window_name,newImage)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[27] ✓ 8.4s Python

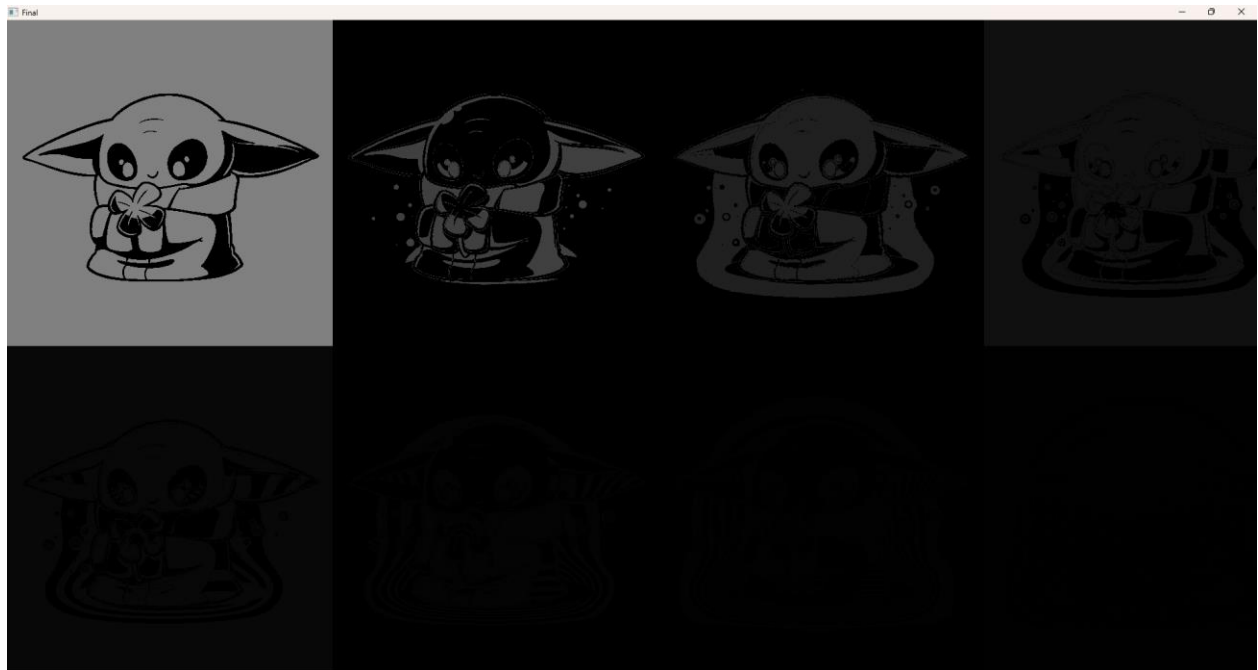
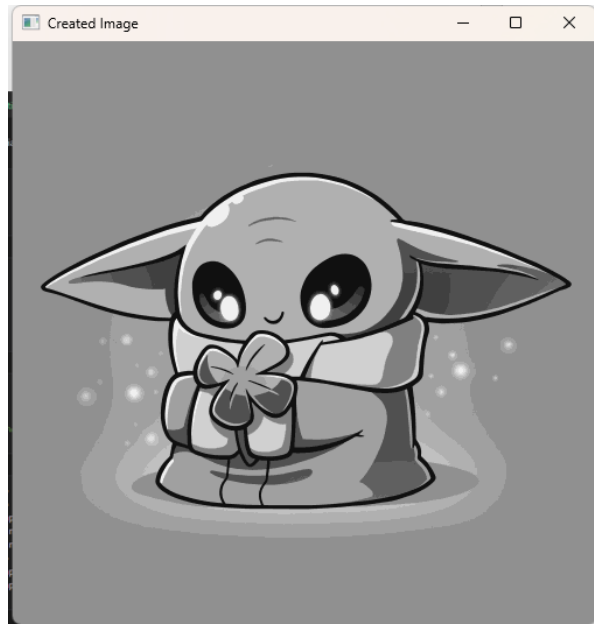
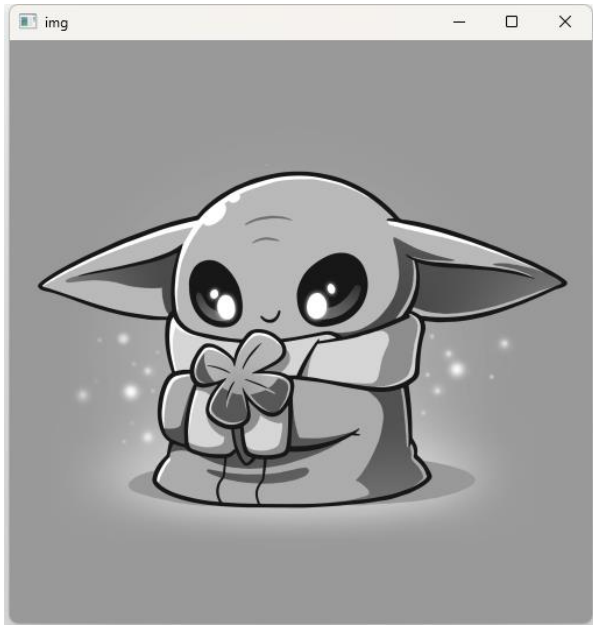


Image Pyramid

```
[7] import cv2
import matplotlib.pyplot as plt
import numpy as np
✓ 0.0s Python
```

```
[8] image = cv2.imread('pic.jpg',0)
r = cv2.pyrDown(image)
✓ 0.0s Python
```

```
[9] window_name = 'img'
cv2.imshow(window_name,r)
cv2.waitKey(0)
cv2.destroyAllWindows()
7.7s Python
```

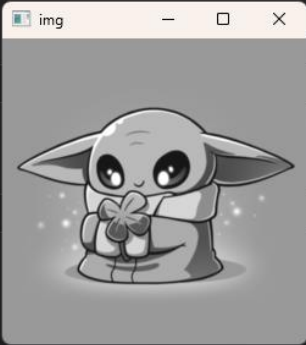


Image Sampling

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

✓ 0.2s Python

```
image = cv2.imread('pic.jpg',1)
window_name = 'img'
cv2.imshow(window_name,image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

✓ 4.6s Python

```
height = image.shape[0] #y-axis
width = image.shape[1] #x-axis
print(height)
print(width)
```

✓ 0.0s Python

2560
1706

```
numHeight = int(height/16)
numWidth = int(width/16)
print(numHeight)
print(numWidth)
```

✓ 0.0s Python

160
106

```
newImg = np.zeros((height,width,3),np.uint8)
for i in range(16):
    #obtain Y coordinates
    y = i*numHeight
    #obtain X coordinates
    x = i*numWidth
    b = image[y,x][0]
    g = image[y,x][1]
    r = image[y,x][2]
```

✓ 0.0s Python

```
window_name = 'New Image'
cv2.imshow(window_name,newImg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

✓ 7.2s Python

```
for i in range(numHeight):
    for j in range(numWidth):
        newImg[y+i,x+j][0] = np.uint8(b)
        newImg[y+i,x+j][1] = np.uint8(g)
        newImg[y+i,x+j][2] = np.uint8(r)
```

✓ 0.0s Python

```
window_name = 'New Image'
cv2.imshow(window_name,newImg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

✓ 3.2s Python

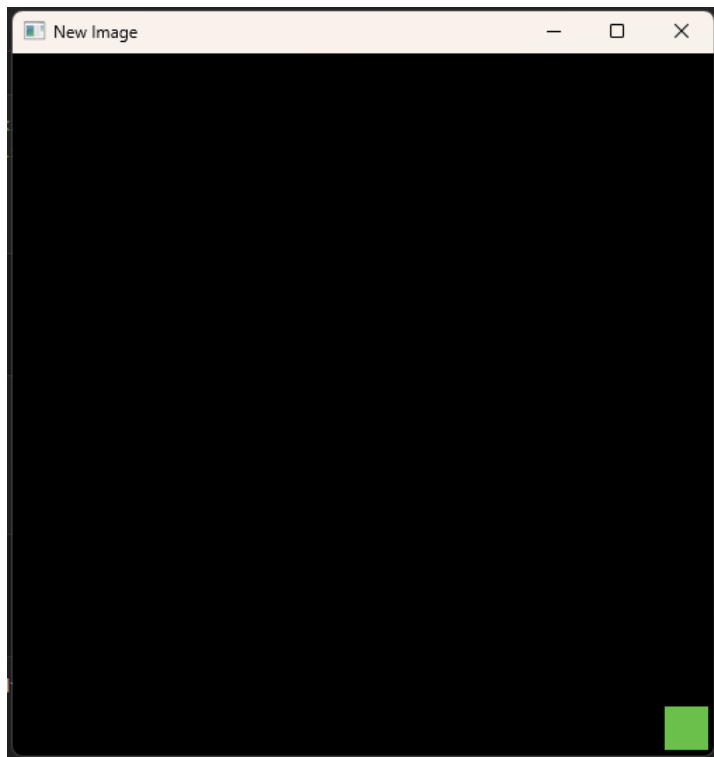
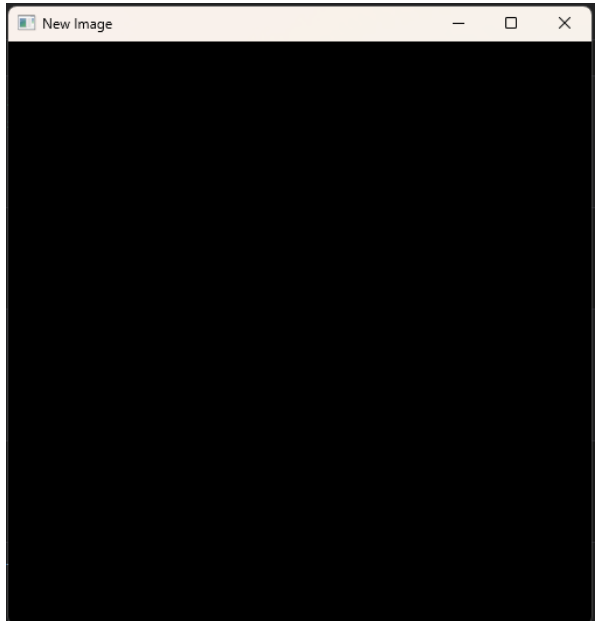
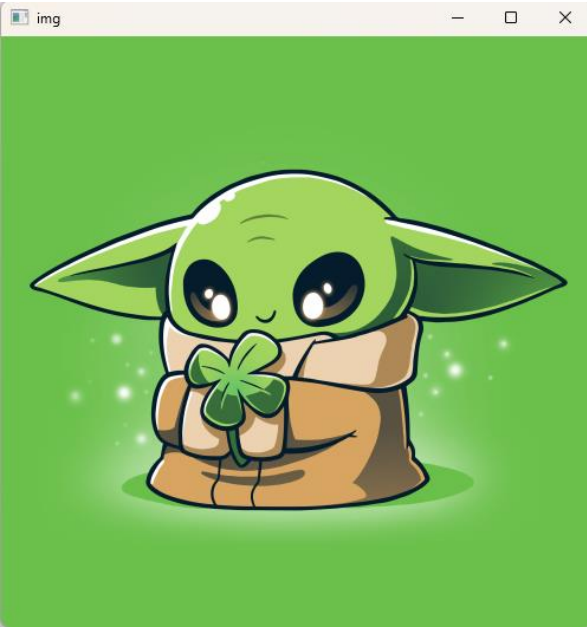


Image Quantization

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

[10] ✓ 0.0s Python

image = cv2.imread('pic.jpg',1)
window_name = 'img'
cv2.imshow(window_name,image)
cv2.waitKey(0)
cv2.destroyAllWindows()

[11] ✓ 7.4s Python

height = image.shape[0] #y-axis
width = image.shape[1] #x-axis
print(height)
print(width)

[12] ✓ 0.0s Python
...
2560
1706

newImg = np.zeros((height,width,3),np.uint8)
for i in range(height):
    for j in range(width):
        for k in range(3):
            if image[i,j][k] < 128:
                gray = 0
            else:
                gray = 128
            newImg[i,j][k] = np.uint8(gray)

[13] ✓ 19.0s Python

window_name = 'New Image Level 2'
cv2.imshow(window_name,newImg)
cv2.waitKey(0)
cv2.destroyAllWindows()

[14] ✓ 4.6s Python

newImg = np.zeros((height,width,3),np.uint8)
for i in range(height):
    for j in range(width):
        for k in range(3):
            if image[i,j][k] < 64:
                gray = 0
            elif image[i,j][k] < 128:
                gray = 64
            elif image[i,j][k] < 192:
                gray = 128
            elif image[i,j][k] >= 192:
                gray = 192
            newImg[i,j][k] = np.uint8(gray)

[15] ✓ 45.6s Python

window_name = 'New Image Level 4'
cv2.imshow(window_name,newImg)
cv2.waitKey(0)
cv2.destroyAllWindows()

[16] ✓ 4.8s Python
```

```
newImg = np.zeros((height,width,3),np.uint8)
for i in range(height):
    for j in range(width):
        for k in range(3):
            if image[i,j][k] < 32:
                gray = 0
            elif image[i,j][k] < 64:
                gray = 32
            elif image[i,j][k] < 96:
                gray = 64
            elif image[i,j][k] < 128:
                gray = 96
            elif image[i,j][k] < 160:
                gray = 128
            elif image[i,j][k] < 192:
                gray = 160
            elif image[i,j][k] < 224:
                gray = 192
            elif image[i,j][k] >= 224:
                gray = 224
            newImg[i,j][k] = np.uint8(gray)
```

[17] ✓ 1m 14.7s Python

```
window_name = 'New Image Level 8'
cv2.imshow(window_name,newImg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[18] ✓ 9.7s Python

