**Ishaan Mehta E18CSE069 EB02 LabWEEK4**

In [1]:
```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
from sklearn.metrics import mean_squared_error
from PIL import Image
```

In [2]:
```
img_path='./Bbiting.jpg'
```

In [3]:
```
img=cv2.imread(img_path)
```

In [4]:
```
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

In [5]:
```
img.shape
```

Out[5]:
```
(256, 256)
```

In [6]:
```
cv2_imshow(img)
```

In [7]:
```
def quantizer(g_img,level):
  g_img_flatten=g_img.reshape((1,-1))
  f_max=np.max(g_img_flatten)
  f_min=np.min(g_img_flatten)
  b=(f_max-f_min)/level
  res=[]
  for i in g_img_flatten:
    res.append(np.floor((i-f_min)/b)*b + b/2+ f_min)
  results=np.array(res).reshape((256,256))
  mse=mean_squared_error(g_img,results)
  return [results,mse]
```

In [8]:
```
levels=[5,16,32,64]
for level in levels:
  res,mse=quantizer(img,level)
  cv2_imshow(res)
  print('Mean squared Error for ',level,' Level is : ',mse,'using custom function')
  print()
```

```
Mean squared Error for  4  Level is :  307.36312103271484 using custom function
```

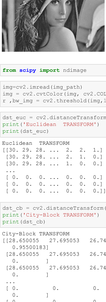
```
Mean squared Error for  16  Level is :  20.874886035931919 using custom function
```

```
Mean squared Error for  32  Level is :  5.1124079227447751 using custom function
```

```
Mean squared Error for  64  Level is :  1.2792555689811707 using custom function
```

In [9]:
```
for level in levels:
  im=Image.open(img_path)
  im1=im.quantize(colors=level)
  print('Displaying the result using im.quantize ')
  display(im1_img)
  print()
```

```
Displaying the result using im.quantize
```



```
Displaying the result using im.quantize
```

```
Displaying the result using im.quantize
```

```
Displaying the result using im.quantize
```

In [10]:
```
from scipy import ndimage
```

In [11]:
```
img=cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
r,bw_img = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
```

In [12]:
```
dst_euc = cv2.distanceTransform(bw_img,cv2.DIST_L1, 3)
print('Euclidean TRANSFORM')
print(dst_euc)
```

```
Euclidean  TRANSFORM
[[30. 29. 28. ...  2.  1.]
 [30. 29. 28. ...  2.  1.]
 [30. 29. 28. ...  1.  0.]
 ...
 [ 0.  0.  0. ...  0.  0.]
 [ 0.  0.  0. ...  0.  0.]
 [ 0.  0.  0. ...  0.  0.]]
```

In [13]:
```
dst_cb = cv2.distanceTransform(bw_img,cv2.DIST_L2, 3)
print('City-Block TRANSFORM')
print(dst_cb)
```

```
City-Block TRANSFORM
[[28.650055  27.695053  26.740051  ...  1.9100037   1.3692932
   0.95500183]
 [28.650055  27.695053  26.740051  ...  1.3692932   0.95500183
   0.        ]
 [28.650055  27.695053  26.740051  ...  0.95500183  0.
   0.        ]
 ...
 [ 0.         0.         0.        ...  0.          0.
   0.        ]
 [ 0.         0.         0.        ...  0.          0.
   0.        ]
 [ 0.         0.         0.        ...  0.          0.
   0.        ]]
```

In [14]:
```
dst_chess = cv2.distanceTransform(bw_img,cv2.DIST_C, 3)
print('CHESS TRANSFORM')
print(dst_chess)
```

```
CHESS TRANSFORM
[[28. 28. 28. ...  2.  1.  1.]
 [27. 27. 27. ...  1.  1.  0.]
 [26. 26. 26. ...  1.  0.  0.]
 ...
 [ 0.  0.  0. ...  0.  0.  0.]
 [ 0.  0.  0. ...  0.  0.  0.]
 [ 0.  0.  0. ...  0.  0.  0.]]
```

In [20]:
```
from PIL import Image, ImageFilter
import matplotlib.pyplot as plt

!matplotlib inline
image = cv2.imread(img_path) # reads the image
image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV) # convert to HSV
figure_size = 7 # the dimension of the x and y axis of the kernal.
new_image = cv2.blur(image,(figure_size, figure_size))
plt.figure(figsize=(11,6))
plt.subplot(121),plt.imshow(cv2.cvtColor(image, cv2.COLOR_HSV2RGB)),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_HSV2RGB)),plt.title('Mean filter')
plt.xticks([]), plt.yticks([])
plt.show()
```


In [21]:
```
from PIL import Image, ImageFilter
import matplotlib.pyplot as plt

!matplotlib inline
image = cv2.imread(img_path) # reads the image
image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV) # convert to HSV
figure_size = 9 # the dimension of the x and y axis of the kernal.
new_image = cv2.blur(image,(figure_size, figure_size))
plt.figure(figsize=(11,6))
plt.subplot(121),plt.imshow(cv2.cvtColor(image, cv2.COLOR_HSV2RGB)),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_HSV2RGB)),plt.title('Mean filter')
plt.xticks([]), plt.yticks([])
plt.show()
```




In [22]:
```
from PIL import Image, ImageFilter
import matplotlib.pyplot as plt

!matplotlib inline
image = cv2.imread(img_path) # reads the image
image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV) # convert to HSV
figure_size = 7 # the dimension of the x and y axis of the kernal.
new_image = cv2.blur(image,(figure_size, figure_size))
plt.figure(figsize=(11,6))
plt.subplot(121),plt.imshow(cv2.cvtColor(image, cv2.COLOR_HSV2RGB)),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_HSV2RGB)),plt.title('Mean filter')
plt.xticks([]), plt.yticks([])
plt.show()
```




In [ ]:
```
```