Imports

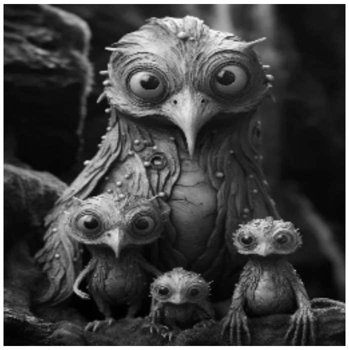
```
import numpy as np
import pandas as pd
from PIL import Image
import cv2
from google.colab.patches import cv2_imshow
```

```
image_path="/content/img.png"

image_color = Image.open(image_path)
gray_image = image_color.convert('L')
```

```
gray_image = cv2.imread('/content/img.png',0)
cv2_imshow(gray_image)
```

Sampling

```
values = np.array(gray_image)
i = 0
x = 256
y = 256

for i in range(5):
  img=cv2.resize(values,(x,y))
  cv2_imshow(img)
  print(x,y,sep='x')
  x=int(x/2)
  y=int(y/2)
```



256x256

128x128

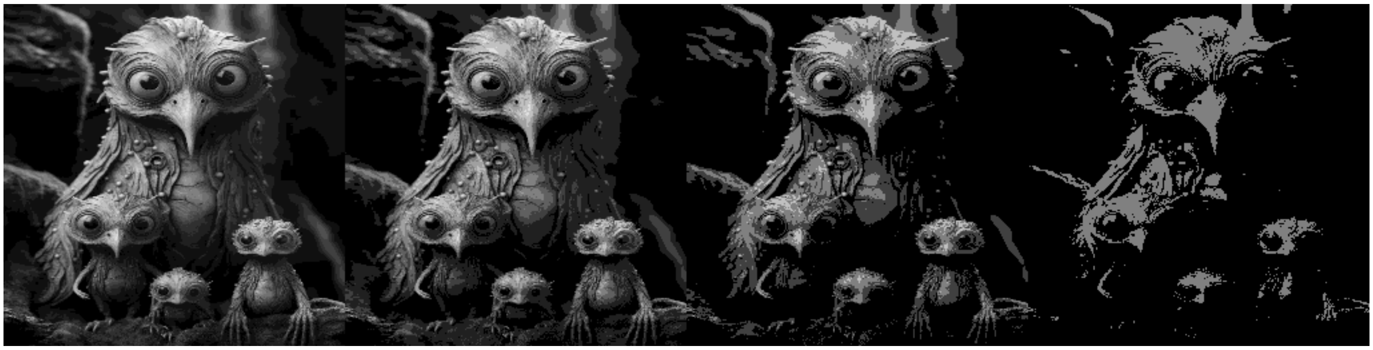
64x64

32x32

16x16

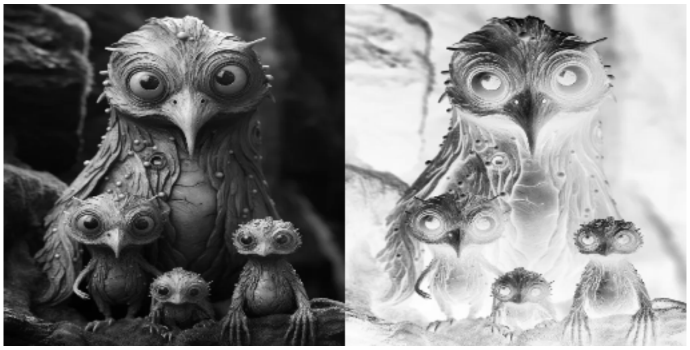
Quantization

```
img = cv2.resize(values,(256,256))

i = 0
x = 16

list1 = []

for i in range(4):
  img1=np.floor_divide(img,256//x)*(256//x)
  list1.append(img1)
  x=int(x/2)

horizontal_stack=np.concatenate(list1,axis=1)
cv2_imshow(horizontal_stack)
```



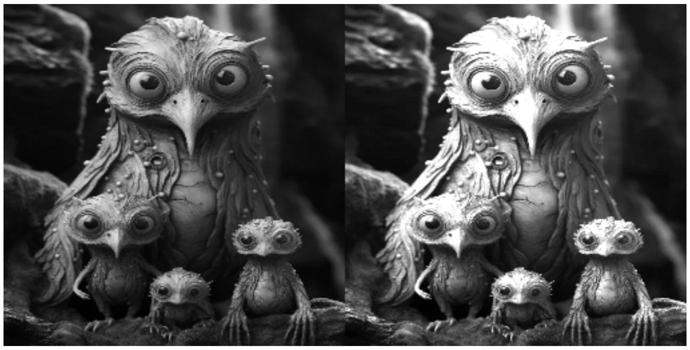
Negative

```
gray_image=cv2.imread('/content/img.png',0)
gray_image=cv2.resize(gray_image,(256,256))
negative=256-1-gray_image
cv2_imshow(np.concatenate((gray_image,negative),axis=1))
```



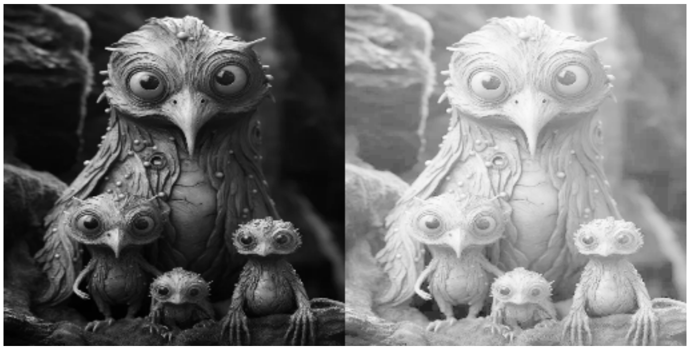
Scaling

```
gray_image=cv2.imread('/content/img.png',0)
gray_image=cv2.resize(gray_image,(256,256))
scaled=1.5*gray_image
cv2_imshow(np.concatenate((gray_image,scaled),axis=1))
```



Log Transformation

```
gray_image=cv2.imread('/content/img.png',0)
gray_image=cv2.resize(gray_image,(256,256))
c=2
log_trans=(np.log((gray_image)+1)/(np.log(1+np.max(gray_image))))*255
cv2_imshow(np.concatenate((gray_image,log_trans),axis=1))
```



Power Law Transformation

```
gray_image=cv2.imread('/content/img.png',0)
two_point_two = np.array(255*(gray_image/255)**2.2,dtype='uint8')
point_four = np.array(255*(gray_image/255)**0.4,dtype='uint8')
cv2_imshow(np.concatenate((gray_image,two_point_two,point_four),axis=1))
```