

Gray Level Slicing

It is used to highlight a specific range of intensities in an image that might be of interest.

Two common approaches:

- Set all the pixel values with a range of interest to one value (white) and all others to another value (black) produces a binary image.
- Brighten (or darken) pixel values in a range of interest and leave all others unchanged.

It is useful highlighting features in an image.

```
In [ ]: #import Libraries
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
```

```
In [ ]: #read the image
# reading image and converting to gray scale
img = Image.open('../images/tiger.jpg').convert('L')
# display image
img
```

Out[]:



```
In [ ]: img = img.resize((200,200), Image.Resampling.LANCZOS)
# convert to numpy array
numpy_image = np.array(img)
```

```
In [ ]: numpy_image.shape
row = numpy_image.shape[0]
```

```
column = numpy_image.shape[1]
```

Set all the pixel values with a range of interest to one value (white) and all others to another value (black) produces a binary image.

```
In [ ]: new_array = np.zeros(shape=(row,column))
```

```
In [ ]: for i in range(row):  
        for j in range(column):  
            new_array[i][j] = numpy_image[i][j]
```

```
In [ ]: new_array
```

```
Out[ ]: array([[ 86.,  86.,  88., ..., 138., 138., 136.],  
               [ 93.,  93.,  94., ..., 148., 147., 146.],  
               [ 98.,  99., 100., ..., 158., 156., 155.],  
               ...,  
               [ 49.,  49.,  50., ...,  72.,  71.,  71.],  
               [ 49.,  49.,  49., ...,  73.,  73.,  72.],  
               [ 48.,  49.,  49., ...,  72.,  72.,  72.]], shape=(200, 200))
```

```
In [ ]: numpy_image
```

```
Out[ ]: array([[ 86,  86,  88, ..., 138, 138, 136],  
               [ 93,  93,  94, ..., 148, 147, 146],  
               [ 98,  99, 100, ..., 158, 156, 155],  
               ...,  
               [ 49,  49,  50, ...,  72,  71,  71],  
               [ 49,  49,  49, ...,  73,  73,  72],  
               [ 48,  49,  49, ...,  72,  72,  72]], shape=(200, 200), dtype=uint8)
```

```
In [ ]: for i in range(row):  
        for j in range(column):  
            if((numpy_image[i][j]>100)&(numpy_image[i][j]<150)):  
                new_array[i][j] = 255  
            else:  
                new_array[i][j] = 0
```

```
In [ ]: new_array
```

```
Out[ ]: array([[ 0.,  0.,  0., ..., 255., 255., 255.],  
               [ 0.,  0.,  0., ..., 255., 255., 255.],  
               [ 0.,  0.,  0., ...,  0.,  0.,  0.],  
               ...,  
               [ 0.,  0.,  0., ...,  0.,  0.,  0.],  
               [ 0.,  0.,  0., ...,  0.,  0.,  0.],  
               [ 0.,  0.,  0., ...,  0.,  0.,  0.]], shape=(200, 200))
```

```
In [ ]: gray_level_slicing_image = Image.fromarray(new_array)  
        gray_level_slicing_image = gray_level_slicing_image.convert("L")  
        gray_level_slicing_image
```

Out[]:



Brighten(or darken) pixel values in a range of interest and leave all others unchanged.

```
In [ ]: new_array2 = np.zeros(shape=(row,column))
```

```
In [ ]: for i in range(row):
        for j in range(column):
            if((numpy_image[i][j]>100)&(numpy_image[i][j]<150)):
                new_array2[i][j] = 0
            else:
                new_array2[i][j] = numpy_image[i][j]
```

```
In [ ]: gray_level_slicing_image2 = Image.fromarray(new_array2)
        gray_level_slicing_image2 = gray_level_slicing_image2.convert("L")
        gray_level_slicing_image2
```

Out[]:

