

convolution_image

July 26, 2021

```
[1]: from PIL import Image, ImageOps, ImageFilter
import numpy as np
```

```
[2]: # Charger une image puis l'afficher
img = Image.open("20200102_154649.jpg")

# Afficher la taille de l'image chargée
img_size = np.asarray(img).shape
print("Taille de l'image :", img_size)
```

Taille de l'image : (2268, 4032, 3)

```
[3]: # Redimensionner l'image pour avoir une largeur de 512
new_width = 512
rescaling_ratio = new_width/img_size[1]
new_dims = (new_width, int(img_size[0] * rescaling_ratio))

# Utiliser une interpolation linéaire pour le redimensionnement
img = img.resize(new_dims)

print("Taille de l'image redimensionnée : ", np.asarray(img).shape)
```

Taille de l'image redimensionnée : (288, 512, 3)

```
[4]: # Afficher l'image redimensionnée
img.show()

# Pour afficher l'image dans le notebook
from IPython.display import Image
display(img)
```



```
[5]: # Convertir l'image en niveaux de gris  
gray_img = ImageOps.grayscale(img)  
#gray_img.show()  
display(gray_img)
```

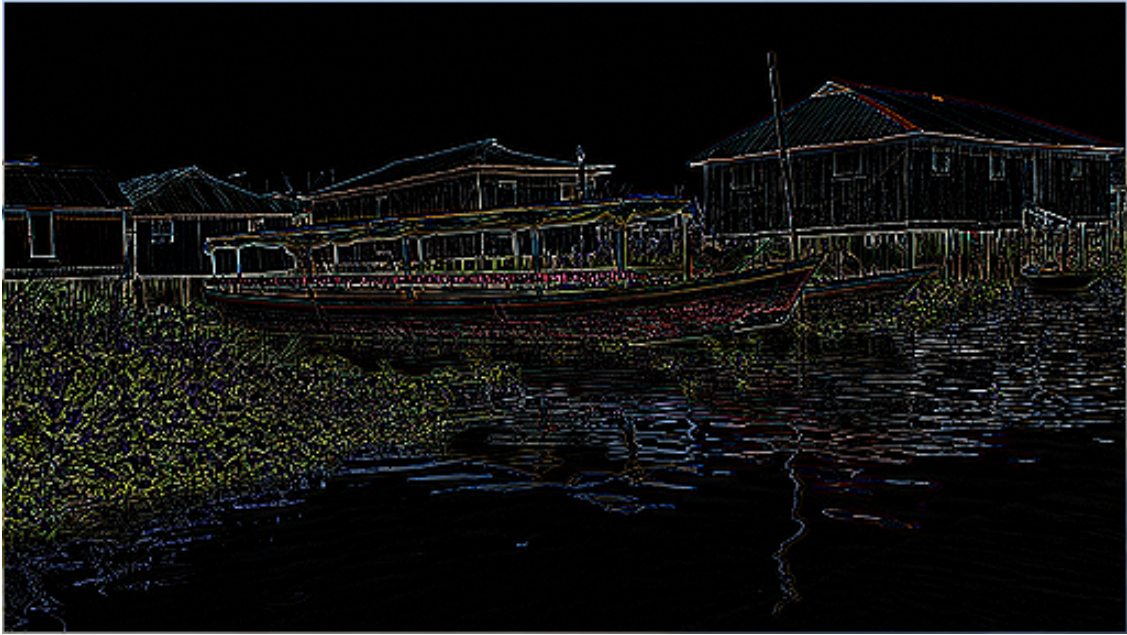


```
[6]: # Filtre laplacien sur l'image couleur
# [0 1 0
# 1 -4 1
# 0 1 0]

img_filtered = img.filter((ImageFilter.Kernel((3, 3),
                                              (0, 1, 0,
                                               1, -4, 1,
                                               0, 1, 0),
                                              1, 0)))



#img_filtered.show()
display(img_filtered)
```



```
[7]: # Filtre laplacien sur l'image en niveau de gris
# [0 1 0
# 1 -4 1
# 0 1 0]

img_filtered = gray_img.filter((ImageFilter.Kernel((3, 3),
                                              (0, 1, 0,
                                               1, -4, 1,
                                               0, 1, 0),
                                              1, 0)))

#img_filtered.show()
```

```
display(img_filtered)
```



```
[ ]:
```