Alae Bouchiba

Groupe 2/2

Mini projet : Traitement d'image et vision

Exercice 1 : élimination de bruit :

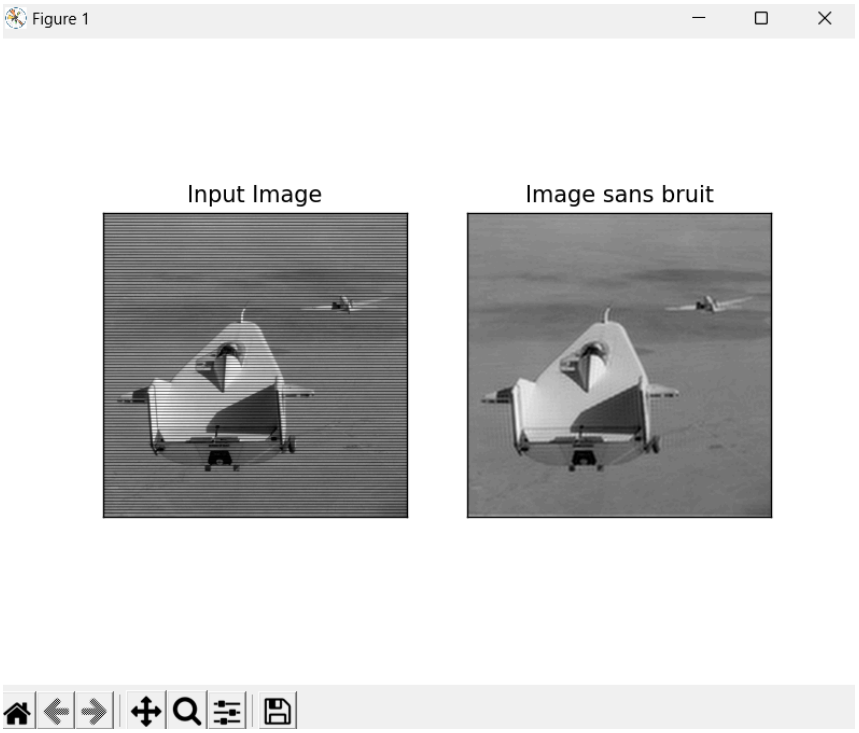
Dans cet exercice, on est demandé d'éliminer le bruit qui est sous forme de traits noirs :

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

# lecture de l'image
image = cv2.imread(r'C:\Users\Alae\Downloads\liftingbodybruite.png',0)

# transformation discrète de fourier de l'image
dft = cv2.dft(np.float32(image),flags = cv2.DFT_COMPLEX_OUTPUT)

# shift du zero au centre du spectrum
dft_shift = np.fft.fftshift(dft)
rows, cols = image.shape
crow,ccol = rows//2 , cols//2
mask = np.zeros((rows,cols,2),np.uint8)
mask[crow-70:crow+70, ccol-70:ccol+70] = 1

# application du mask et inverse DFT
fshift = dft_shift*mask
f_ishift = np.fft.ifftshift(fshift)
img_back = cv2.idft(f_ishift)
img_back = cv2.magnitude(img_back[:,:,0],img_back[:,:,1])

# visualization de l'image
plt.subplot(121),plt.imshow(image, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(img_back, cmap = 'gray')
plt.title('Image sans bruit'), plt.xticks([]), plt.yticks([])
plt.show()
cv2.waitKey(0)
```

- ★ Pour ce faire,Nous verrons d'abord comment trouver la transformée de Fourier en utilisant Numpy. Numpy a un package FFT pour le faire. np.fft.fft2() nous fournit la transformée de

fréquence qui sera un tableau complexe. Son premier argument est l'image d'entrée, qui est en niveaux de gris. Le deuxième argument est facultatif et décide de la taille du tableau de sortie.Maintenant, une fois que nous avons obtenu le résultat, le composant de fréquence zéro (composant DC) sera dans le coin supérieur gauche. Si nous voulons le centrer, nous devons décaler le résultat de \frac{N}{2} dans les deux sens. qui est fait par la fonction, np.fft.fftshift().

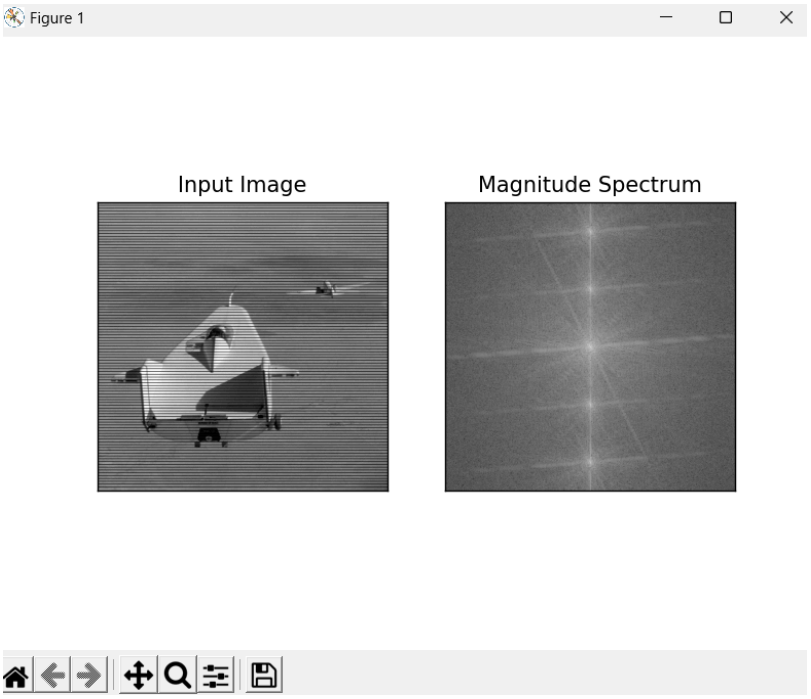
★ Après avoir trouvé la transformée de fréquence. On peut effectuer certaines opérations dans le domaine fréquentiel, comme le filtrage passe-haut et reconstruire l'image, c'est-à-dire trouver la DFT inverse. Pour cela il nous suffit de supprimer les basses fréquences en les masquant avec une fenêtre rectangulaire de taille 70x70(comme utilisé dans notre cas). On applique ensuite le décalage inverse à l'aide de np.fft.ifftshift() afin que la composante DC revienne dans le coin supérieur gauche.on Trouve ensuite la FFT inverse à l'aide de la fonction np.fft 2(). Ainsi, on obtient après affichage, notre image sans bruit.



Remarque : si on cherche à afficher le spectrum de magnitude, il suffit d'ajouter les lignes de code suivantes :

```
dft_shift = np.fft.fftshift(dft)
magnitude_spectrum = 20*np.log(cv2.magnitude(dft_shift[:,:,0],dft_shift[:,:,1]))
```

```
plt.subplot(121),plt.imshow(image, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(magnitude_spectrum, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()
```



****

Exercice 2 : élimination de bruit et sauvegarde résultat

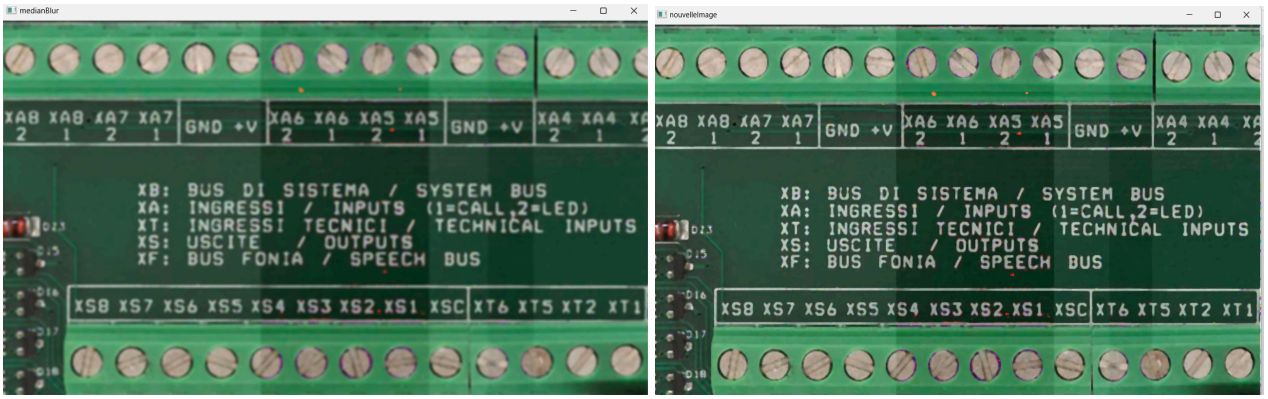
Dans cette partie, on est demandé d'éliminer le bruit et de sauvegarder le résultat dans un fichier

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
from scipy import ndimage
import matplotlib.image as mpimg
from PIL import Image
import PIL

ImageOriginal= cv.imread(r'C:\Users\Alae\Downloads\cartebruitee.png')
# filtre médian
```

```
img=cv.medianBlur(ImageOriginal, 5)
newImg = cv.GaussianBlur(img, (5, 5), 3)
cv.imshow("nouvelleImage",img)

# enregistrer l'image
imwrt = cv.imwrite(r'C:\Users\Alae\Downloads\medianBlur.jpg',newImg)
cv.imshow('medianBlur', newImg)
if imwrt:
 print('Image enregistrée avec succès.')

cv.waitKey(0)
```

- ★ Le filtre médian est celui qui remplace chaque valeur de pixel par la médiane de son pixel voisin. La méthode medianBlur() est idéale lorsqu'il s'agit d'une image avec un bruit poivre et se(comme trouvé dans l'image originale)l.On y ajoute le filtre Guassien, pour supprimer tout autre type de bruit

★



****

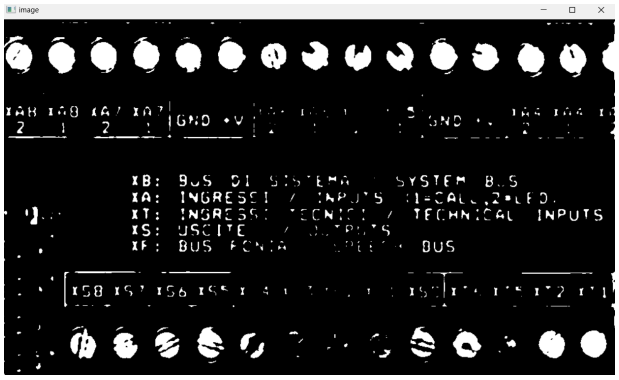
Dans l'autre partie, on veut obtenir des caractères blancs sur fond noir :

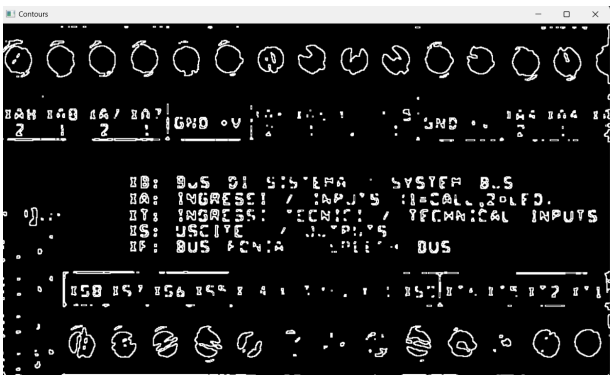
```
import numpy as np
import cv2
from matplotlib import pyplot as plt


image= cv2.imread(r'C:\Users\Alae\Downloads\medianBlur.jpg')
kernel = cv2.getStructuringElement(cv2.MORPH_OPEN,(3,3))

imagegray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

(retVal, I2) = cv2.threshold(imagegray, 150, 255, cv2.THRESH_BINARY)
cv2.imshow('image',I2)
#ouverture de l'image
ouverture = cv2.morphologyEx(I2, cv2.MORPH_OPEN, kernel)
cv2.imshow('ouverture',ouverture)
#détermier les contours
```

```
SobelC=np.array([[-1,-2,-1],[0,0,0],[1,2,1]])
SobelL=np.array([[-1,0,1],[-2,0,2],[-1,0,1]])
W1=abs(cv2.filter2D(I2,cv2.CV_64F,SobelC))
W2=abs(cv2.filter2D(I2,cv2.CV_64F,SobelL))
W=W1+W2
cv2.imshow('Contours ',W  )

cv2.waitKey(0)
# closing all open windows
cv2.destroyAllWindows()
```

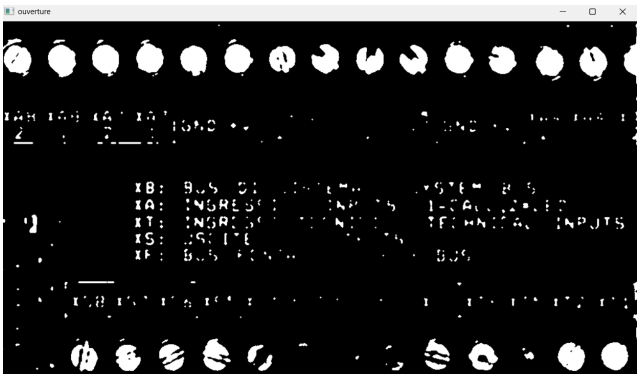
- ★ On utilise notre image medianBlur déjà enregistrée pour rendre ses pixels ou bien noir (0) ou blanc (255), étant donné que le fond sera noir.ceci est appelé seuillage, et c'est guarantie grace à la fonction THRESH_BINARY, les valeurs supérieures ou égales à 150 sont mises à 255, les autres à 0.
- ★ On utilise Sobel, qui est un filtre 3 x 3, pour détecter les contours, dans les directions x et y (d'où le 1 répété 2 fois) pour obtenir une image de type uint8.

★





★



****

Exercice 3 : Changement de couleur de drapeau :

```
import numpy as np
import cv2 as cv
```

```
#obtention path et lecture de l'image originale
path=r'C:\Users\Alae\Downloads\DrapeauAllemagne.png'
img = cv.imread(path)

#affichage image originale
cv.imshow('originale',img)

#determination de l'hauteur
h=np.size(img, 0)
print("hauteur =",h)

#determiner la longueurlongueur
l=np.size(img, 1)
print("longueur =",l)

#determiner l'hauteur d'une bande du drapeau
b=h/3
print("bande=",b)

#changer couleur 1ere bande
(cx1,cy1)=(96,480)
img[0:cx1,0:cy1]=(0,255,255)

#changer couleur 2eme bande
(cx2,cy2)=(192,480)
img[96:cx2,0:cy2]=(100,190,50)

##changer couleur 3eme bande
(cx3,cy3)=(288,480)
img[192:cx3,0:cy3]=(50,0,255)

#affichage image sortie
cv.imshow('sortie',img)

#savegarde image sortie
filename=r'C:\Users\Alae\Downloads\image_sortie.png'
cv.imwrite(filename,img)

cv.waitKey(0);
cv.destroyAllWindows();
cv.waitKey(1)
```

- ★ On commence par couper l'image en 3 pour obtenir 3 bandes, et identifier les hauteurs et les longueurs de chaque bande (en divisant par 3, car ils sont identiques) puis identifier la couleur de chaque bande, et la convertir en utilisant le code couleur convenable pour chaque bande.


originale
sortie