PROJET PYTHON

Nom du projet : Traitement d'image

Présenté par : CHAMI Younes
CAMARA FAMAKAN
DJAENFAR ABLEZE
BENKHLIFA Yousra
DAMMARI Hafsa

Encadré par : Ms. SAADI Mostafa

Plan du Projet

- Remerciement
- Introduction

Partie I:
Les opérations d'entrée/sortie sur les images

Partie II:
Les images Noir et blanc

Partie III:
Les images en niveau de gris

Partie IV:
Opérations élémentaires sur les images en mode gris

Partie V:
Les images RGB

Remerciement :

Nous tenons à remercier notre Prof Mr SAADI Mostafa
pour sa guidance et ses précieux conseils tout au long
de ce mini projet de traitement d'images en Python.
nous sommes persuadés que ses enseignements
nous aideront à poursuivre nos études et nos carrières
avec succès. Nous sommes très reconnaissants d'avoir
eu l'opportunité de travailler dans ce projet
et avec un Professeur aussi dévoué et passionné.

Introduction :

Notre monde d’aujourd’hui regorge de données et les images en constituent une partie importante. Cependant, pour pouvoir être utilisées, ces images doivent être traitées. Le traitement d’images consiste donc à analyser et à manipuler une image numérique principalement dans le but d’en améliorer la qualité ou d’en extraire des informations qui pourraient ensuite être utilisées. Les tâches courantes du traitement d’images incluent l’affichage des images, les manipulations basiques comme le recadrage, le retournement, la rotation, ou encore la segmentation, la classification et les extractions de caractéristiques, la restauration et la reconnaissance d’images. Python devient un choix judicieux pour de telles tâches de traitement d’images. Cela est dû à sa popularité croissante en tant que langage de programmation scientifique et à la disponibilité gratuite de nombreux outils de traitement d’images de pointe dans son écosystème

Partie I:

Les opérations d'entrée/sortie sur les images

- AfficherImg
- ouvrirImage
- saveImage

Code Source

```python
from matplotlib import pyplot as plt
#********** \\ Question 1 :AfficherImg // *********
def AfficherImg(img,m=255,n=0):
    plt.axis("off") # pour desactiver les axes
    plt.imshow(img, interpolation="nearest")
# on a choisi m,n comme valeurs pour assurer l'accer au choix vmax et vmin
    plt.show()
```

Explication

La fonction AfficherImg affiche l'image associée à la matrice donnée en argument sur l'écran.

Code Source

```
#********** \\ Question 2: ouvrir Image //*********
def ouvrirImage(chemin):
    img=plt.imread(chemin)
    return img
```

Explication

La fonction ouvrirImage prend en argument un chemin d'image et retourne une matrice contenant son codage.

Code Source

```
#**********\\ Question 3 : save Image //**************
def saveImage(img):
    plt.imsave("image1.png",img)
```

Explication

La fonction saveImage prend en argument une matrice représentant une image et enregistre cette image sur le disque dur.

Partie II:

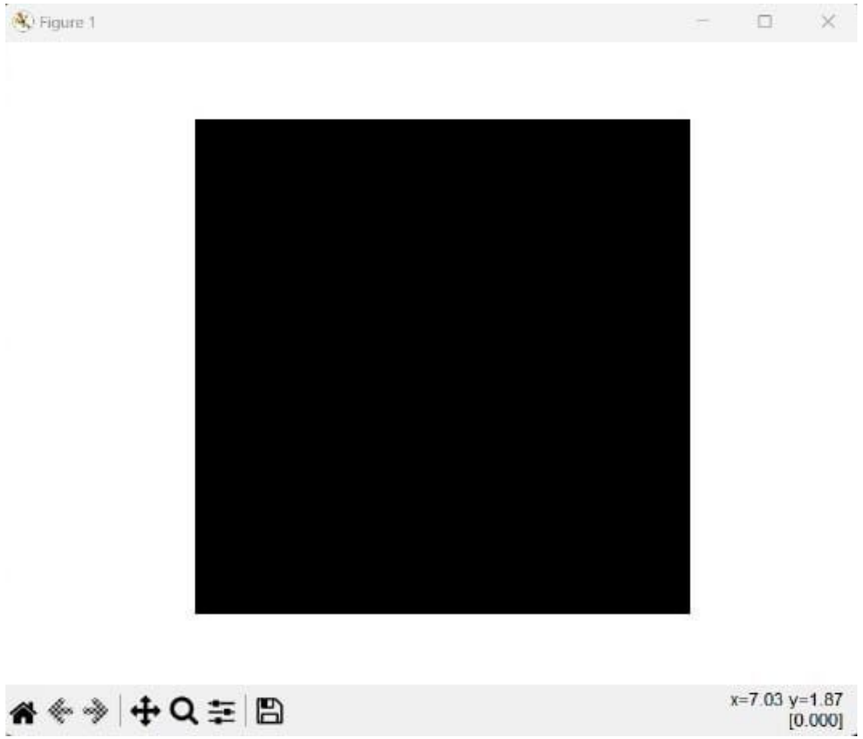
Les images Noir et blanc

- image_noire
- image_blanche
- creerImgBlancNoir
- negatif

Code Source

```python
#**************\\  Création  d'une image noir //**************#
def image_noire(h, l):
    img=np.zeros((h,l))
    return img
```

Compilation



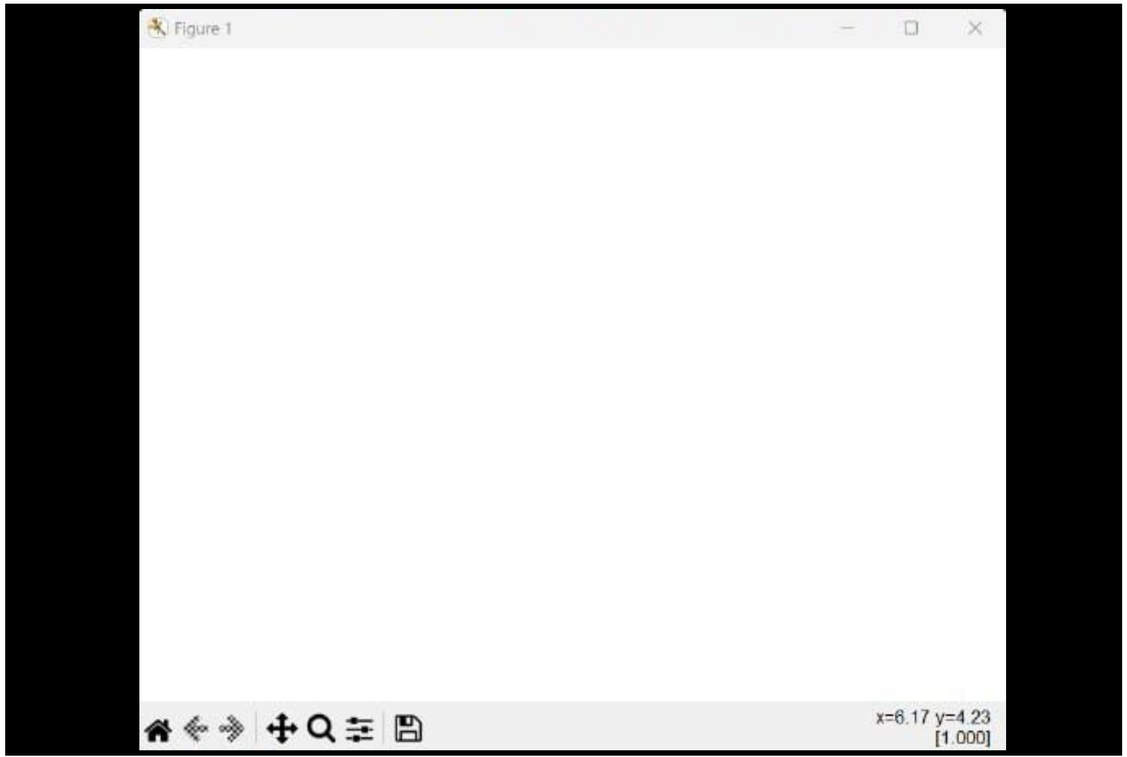
Explication

La fonction image_noire crée une image noire de dimensions spécifiées en entrée (hauteur h et largeur l). Elle retourne une matrice l lignes * h colonnes, dans laquelle chaque pixel est initialisé à la valeur 0.

Code Source

```python
#**************\\ Création  d'une image blanche //**************#
def image_blanche(h, l):
    img=np.ones((h,l))
    return img
```

Compilation



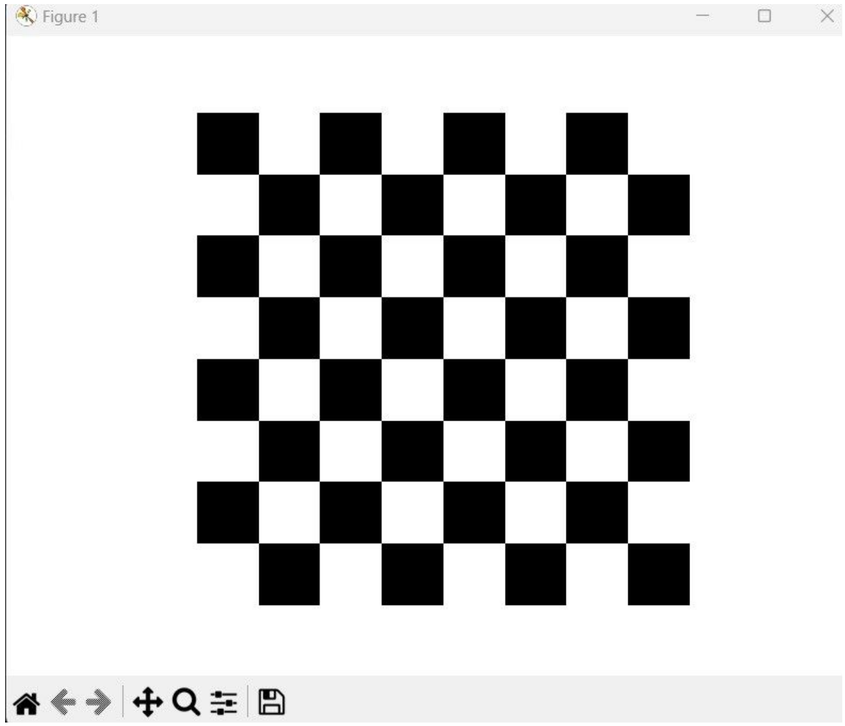
Explication

La fonction image_blanche crée une image blanche de dimensions spécifiées en entrée (hauteur h et largeur l). Elle retourne une matrice l lignes * h colonnes, dans laquelle chaque pixel est initialisé à la valeur 1.

Code Source

```
#**************\\  Création  d'une image noir et blanc //**************#
def creerImgBlancNoir(h,l):
    img=np.array([[(i+j)%2 for i in range(l)]for j in range(h)])
    return img
```

Compilation



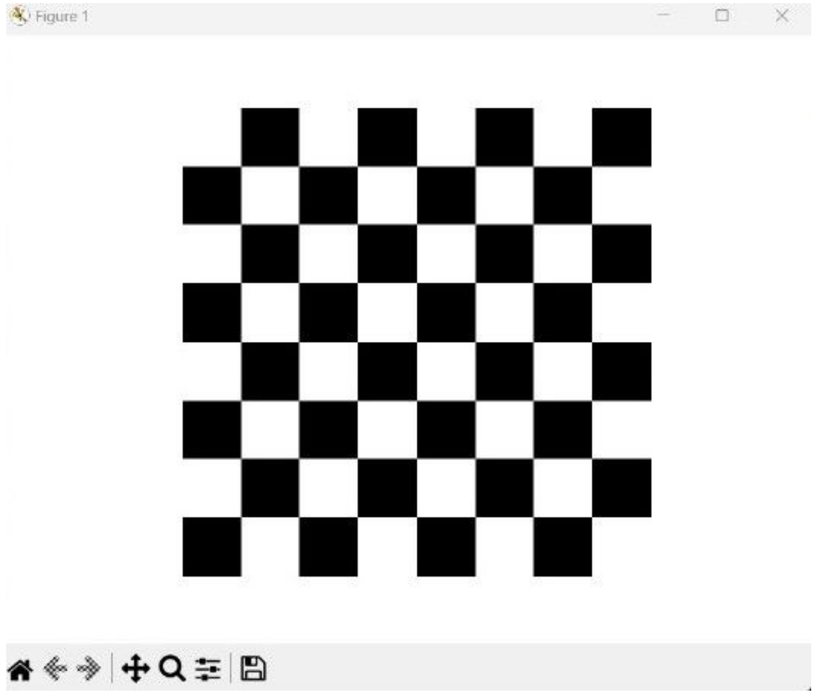
Explication

La fonction creerImgBlancNoir génère une image en noir et blanc. Cette image peut être représentée sous forme d'une matrice de pixels, chaque pixel ayant une valeur de 0 (noir) ou 1 (blanc).

Code Source

```python
#**************\\  Construction du negative d'une image //**************#
def negatif(Img):
    img=np.array([[1-j for j in i] for i in Img])
    return img
```

Compilation



Explication

La fonction "negatif" prend en entrée une matrice "Img" représentant une image et renvoie une image négative en inversant les valeurs de la matrice "Img" (les valeurs 0 deviennent 1 et les valeurs 1 deviennent 0).

Partie III:

Les images en niveau de gris

- luminance
- constrast
- profondeur
- ouvrir

Code Source

```
#*********************\\ Question1: luminance d'une image //*********************#
def luminance(Img):
    return (Img.max()+Img.min())/2
```

Explication

La fonction luminance calcule la luminance de l'image , qui est une mesure de la luminosité perçue par l'œil humain.Elle renvoie enfin une valeur numérique représentant la luminance de l'image.

Compilation

```
la valeur de luminace est : 119.22241111111
```

Code Source

```python
#*********************\\ Question2: Contraste d'une image //*********************#
def constrast(Img):
    s=0
    for i in Img:
        for j in i:
            s+=(i[0]-luminance(Img))**2
    # au lieu de transformer à deux dimension j'ai pris un j[0]
    # puisque les trois en la meme valeur
    return (1/(len(Img)*len(Img[0])))*s
```

Explication

La fonction contraste calcule le contraste de l'image , qui est une mesure de la variance des niveaux de gris de l'image. Le contraste est donc déterminé en comparant les niveaux de gris de chaque pixel de l'image.

Compilation

```
la valeur de constrast est : 2522.5170332980347
```

Code Source

```python
#*********************\\ Question3: Valeur maximale d'un pixel dans l'image //*************#
def profondeur(Img):
    if Img.ndim==2:
        if Img.max()==1:
            return 1
        else: return 8
    else:
        return 24
    #par unité bit
```

Compilation

```
la valeur de la profondeur est :  24
```

Explication

La fonction profondeur renvoie la valeur maximale des pixels dans l'image "Img". Cette valeur est également appelée "profondeur de couleur" ou "profondeur de bits", et elle détermine le nombre de bits utilisés pour représenter chaque pixel de l'image.

```python
#****************\\ Question4: La matrice représentant l'image A.//***************
def ouvrir(chemin="C:\\Users\\camar\\OneDrive\\Pictures\\Image du projet1.PNG"):
    return ouvrirImage(chemin)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
  [0.14901961 0.14901961 0.14901961 1.        ]
  ...
  [0.11764706 0.11764706 0.11764706 1.        ]
  [0.09803922 0.09803922 0.09803922 1.        ]
  [0.10980392 0.10980392 0.10980392 1.        ]]

 [[0.09803922 0.09803922 0.09803922 1.        ]
  [0.12156863 0.12156863 0.12156863 1.        ]
  [0.12941177 0.12941177 0.12941177 1.        ]
  ...
  [0.12941177 0.12941177 0.12941177 1.        ]
  [0.11764706 0.11764706 0.11764706 1.        ]
  [0.14901961 0.14901961 0.14901961 1.        ]]

 [[0.09411765 0.09411765 0.09411765 1.        ]
  [0.11372549 0.11372549 0.11372549 1.        ]
  [0.11764706 0.11764706 0.11764706 1.        ]
  ...
  [0.13333334 0.13333334 0.13333334 1.        ]
  [0.16078432 0.16078432 0.16078432 1.        ]
  [0.21568628 0.21568628 0.21568628 1.        ]]]
PS C:\Users\camar\OneDrive\Bureau\projetpy>
```

```
PS C:\Users\camar\OneDrive\Bureau\projetpy> & C:/Us
mar/OneDrive/Bureau/projetpy/Partie3.py
[[[0.54901963 0.54901963 0.54901963 1.        ]
  [0.5568628  0.5568628  0.5568628  1.        ]
  [0.54509807 0.54509807 0.54509807 1.        ]
  ...
  [0.3882353  0.3882353  0.3882353  1.        ]
  [0.3882353  0.3882353  0.3882353  1.        ]
  [0.34509805 0.34509805 0.34509805 1.        ]]

 [[0.5372549  0.5372549  0.5372549  1.        ]
  [0.5529412  0.5529412  0.5529412  1.        ]
  [0.54509807 0.54509807 0.54509807 1.        ]
  ...
  [0.34117648 0.34117648 0.34117648 1.        ]
  [0.21568628 0.21568628 0.21568628 1.        ]
  [0.1254902  0.1254902  0.1254902  1.        ]]

 [[0.5254902  0.5254902  0.5254902  1.        ]
  [0.54509807 0.54509807 0.54509807 1.        ]
  [0.5411765  0.5411765  0.5411765  1.        ]
```

Partie IV:

Opérations élémentaires sur les images en mode gris

- inverser
- flipH
- poserV
- poserH

Code Source

```
#*********************\\ Question1: Inverse d'une image //*********************#
def inverser(img):
    inv=np.array([[255-j for j in i] for i in img])
    return inv
```

Compilation

Explication

La fonction inverser prend en entrée une matrice d'une image et renvoie une image inversée de cette image. Pour créer l'image inversée, la fonction calcule la valeur inverse de chaque pixel de l'image

Code Source :

```python
#*******\\ Question2: la transformée d'une image  par la symétrie d’axe vertical //*******#
def flipH(img):
    t=[]
    for i in range(len(img)):
        t=t+[img[i][::-1]]
    return t
```

Compilation :

Explication

La fonction flipH prend en entrée une matrice image et renvoie une version de cette image qui a été symétrique par rapport à un axe vertical passant par le milieu de l'image.

Code Source

```python
#********************\\ Question3: Poser vertical d'une image //********************#
def poserV(img1,img2):
    img3=[]
    if profondeur(img1)==profondeur(img2):
        if len(img1[0])==len(img2[0]):
            img3+=list(img1)+list(img2)
            np.array(img3)
    return img3
```

Compilation

Explication

La fonction poserV prend en entrée deux images de même largeur et profondeur, et renvoie une nouvelle image qui est obtenue en superposant verticalement les deux images.

Code Source

```python
#*********************\\ Question4: Poser horizontal d'une image //*********************#
def poserH(img1, img2):
    img3=[[[0] for i in range(len(img1[0])+len(img2[0]))]for j in range(len(img1))]
    if profondeur(img1)==profondeur(img2):
        if len(img1)==len(img2):
            for i in range(len(img1)):
                img3[i]= list(img1[i])+list(img2[i])
    return np.array(img3)
```

Compilation

Explication

La fonction poserH prend en entrée deux images de même hauteur et profondeur, et renvoie une nouvelle image qui est obtenue en superposant horizontalement 2eme image à droite de 1 ere image.

Partie V:

Les images RGB

- initImageRGB
- symetrie
- grayscale

Code Source

```
M=[[[210, 100, 255],[100, 50, 255],[90, 90, 255],[90, 90, 255],[90, 90, 255],[90, 80, 255]],
[[190, 255,89],[ 201, 255,29],[200, 255,100],[100, 255,90],[20, 255,200], [100, 255,80]],
[[255,0, 0],[ 255,0, 0],[255,0, 0],[255,0, 0],[255,0, 0], [255,0, 0]]]
#************\\ Question 1: Valeurs des expressions suivantes : //***********#
print(M[0][1][1])
print(M[1][0][1])
print(M[2][1][0])
```

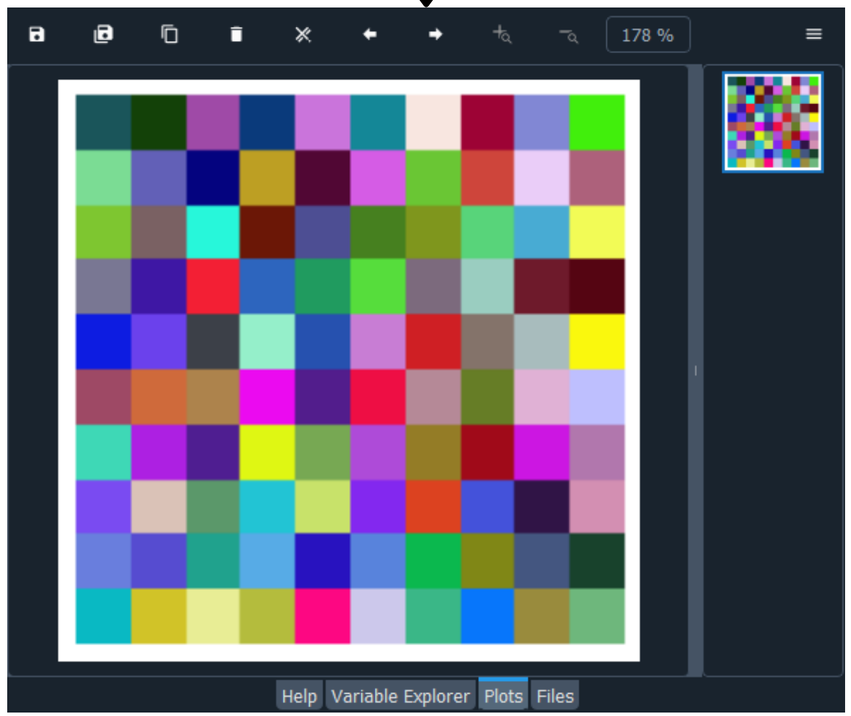
Compilation

```
l'element M[0][1][1]= 50
l'element M[1][0][1]= 255
l'element M[2][1][0]= 255
```

Code Source

```python
#*************\\ Question 3: Initialisation et renvoie du tab rgb en 3D //************#
def initImageRGB(h,l):
    img=[[[random.randrange(255) for i in range(3)] for j in range(l) ]for k in range(h)]
    return img
```

Compilation



Explication

La fonction initImageRGB initialise et renvoie une matrice à trois dimensions de manière aléatoire, cette matrice est représente une image couleur au format RGB (rouge, vert, bleu).

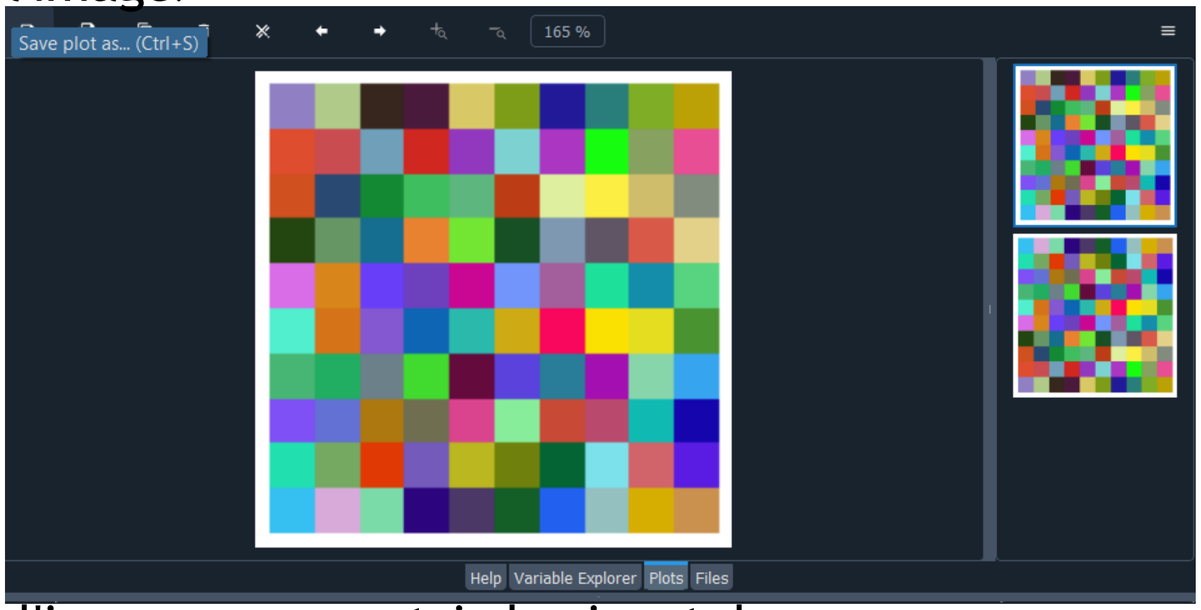
Code Source

```python
#*****************\\ Question 4: symetrie horisontale et verticale //*****************#
#Q4>>a
def symetrieH(img):
    return img[::-1]
```

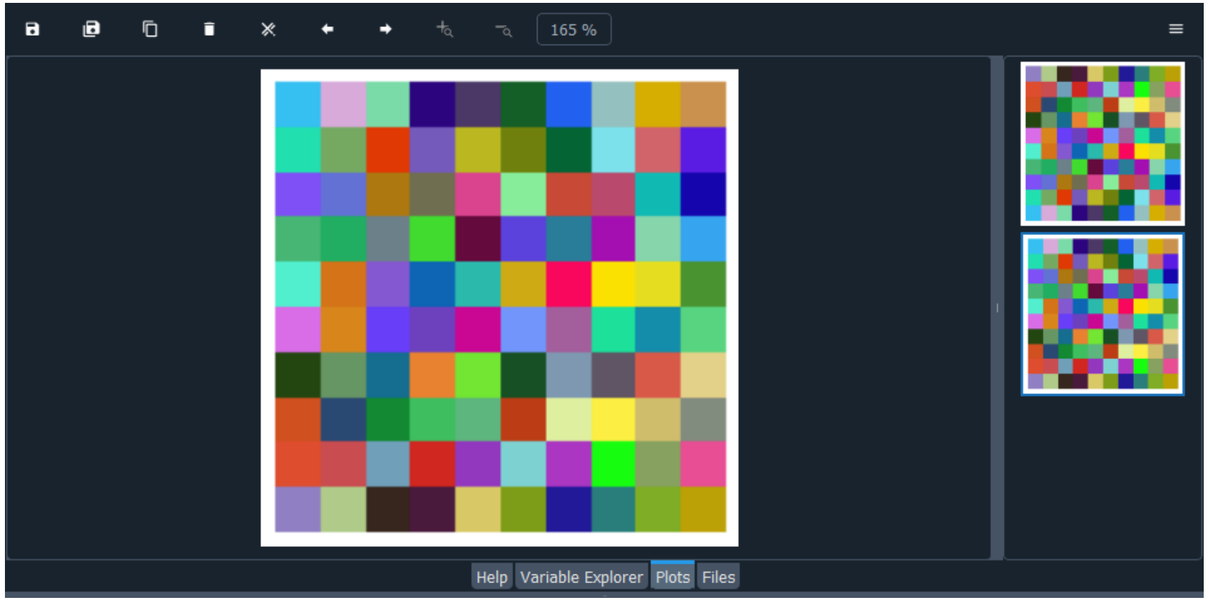
Explication

La fonction symetrie prend en entrée une matrice et renvoie une version symétrique de cette image par rapport à l'axe horizontal.

Compilation l'image:



l'image par symetrie horizontale



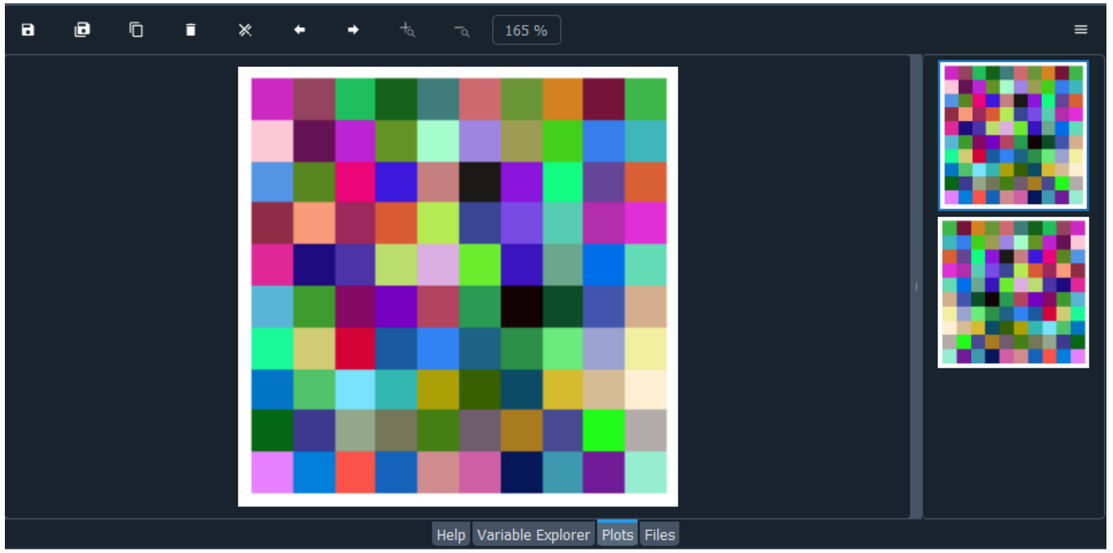
Code Source

```python
#Q4>>b
def symetrieV(img):
    img3=[[[0] for i in range(len(img[0]))]for j in range(len(img))]
    for i in range(len(img)):
        img3[i]=list(img[i][::-1])
    return np.array(img3)
```

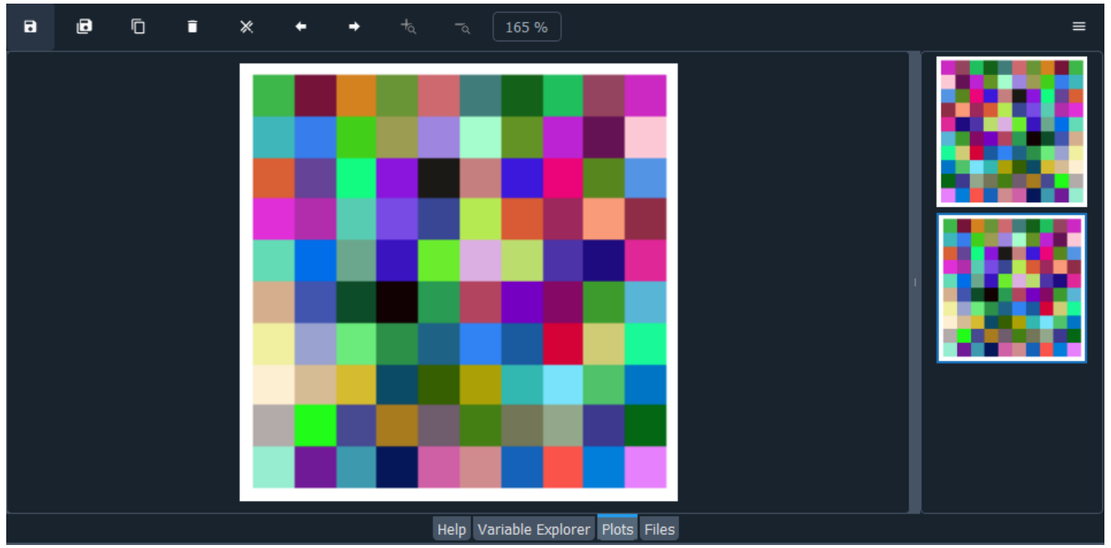
Explication

La fonction symetrie prend en entrée une matrice et renvoie une version symétrique de cette image par rapport à l'axe verticale.

Compilation l'image:



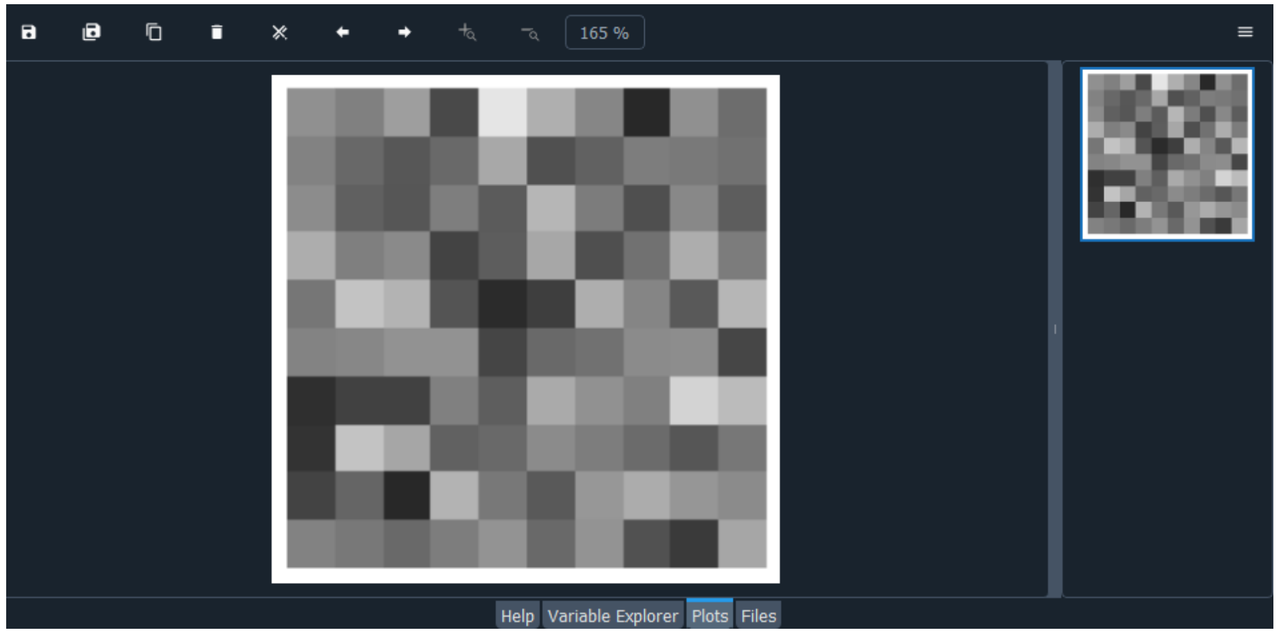
l'image par symetrie verticale



Code Source

```
#*********************\\ Question5:renvoie une image en niveaux degris... //************#
def grayscale(imageRGB):
    e=[[floor(((j.max()+j.min())/2))for j in i]for i in np.array(imageRGB)]
    return e
```

Compilation



Explication

La fonction Grayscale prend en entrée une matrice d'image couleur RGB, et renvoie une version en niveaux de gris de cette image.

CONCLUSION :

En conclusion, ce mini projet de traitement d'imagesen Python nous a permis de découvrir les différentes possibilités offertes par Python pour manipuler et traiter des images. Nous avons utilisé différentes bibliothèques et outils, tels que Pillow, et NumPy, pour charger, afficher, et manipuler des images.

Nous avons également appris à effectuer des opérations de base sur les images, telles que la redimensionne, la rotation. L'une des principalesdifficultés rencontrées au cours de ce projet a été de comprendre et de mettre en pratique les différentes techniques de traitement d'images. Il a fallu du temps pour apprendre à utiliser les bibliothèques et les outils nécessaires, et il a fallu de la patience et de la persévérance pour maîtriser les concepts et les techniques. Cependant, grâce à l'aide de Mr Saadi Mostafa et à notre propre travail acharné, nous avons réussi à surmonter ces difficultés et à réaliser un programme de traitement d'images fonctionnel.

En fin de compte, ce projet a été une expérience très enrichissante et nous espérons pouvoir continuer à explorer et à apprendre davantage sur le traitement d'images en Python dans le futur

Bilan de travail

Apres avoir reçu le travail nous avons décidé que chacun de nous va faire tout le travail on se qui concerne la création des fonctions après on se réunit pour les discuter et corrigé après choisir les codes qu'on a apprécié.

Pour la préparation du reste on a diviser les taches comme suit :

- Code main .
- préparation du rapport Word.
- Fichier exécutable.