GAGUA Oussama

YOVODEVI Zaide

Date : 30/11/2021

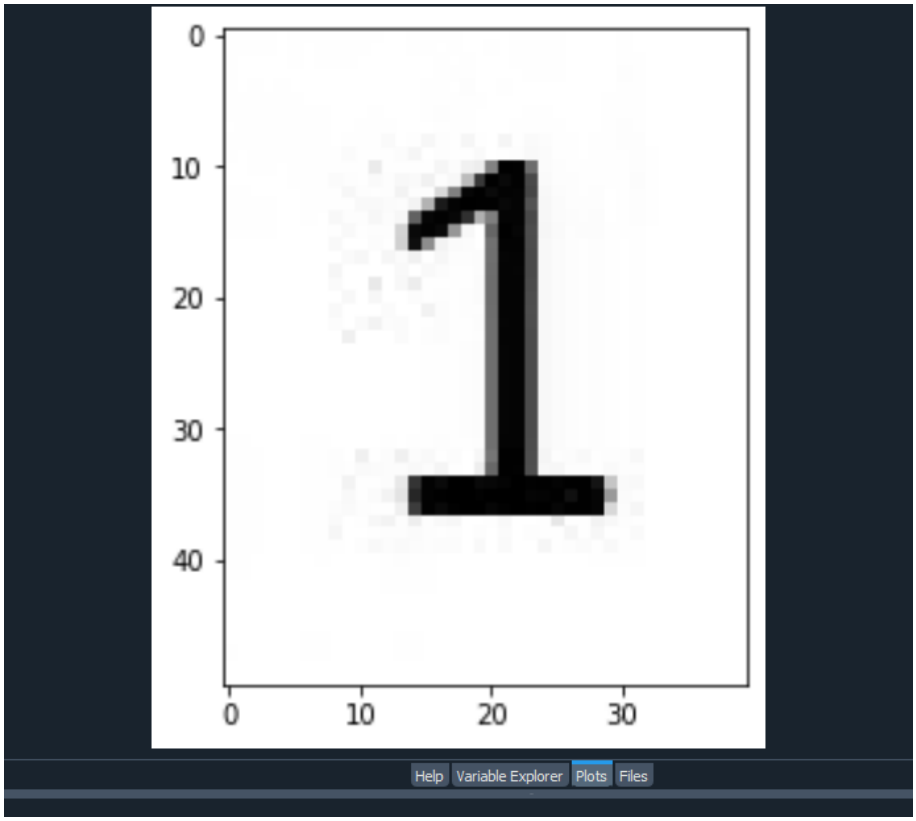
TP2 - Lecture automatique de chiffres par analyse d'image

I-Présentation du TP :

La première partie nous a permis de comprendre le TP. L'objectif est d'analyser une image représentant un chiffre et de pouvoir faire la reconnaissance de ce dernier. Un TP intéressant et utile nécessitant de travailler avec les pixels, de binariser ces derniers et de réfléchir sur la reconnaissance des chiffres. Nous avons utilisé la librairie Numpy car une image est un tableau du style np.array

II - Prise en main de l'environnement :

L'exécution du fichier main.py montre bien qu'on a une image qui est chargée :



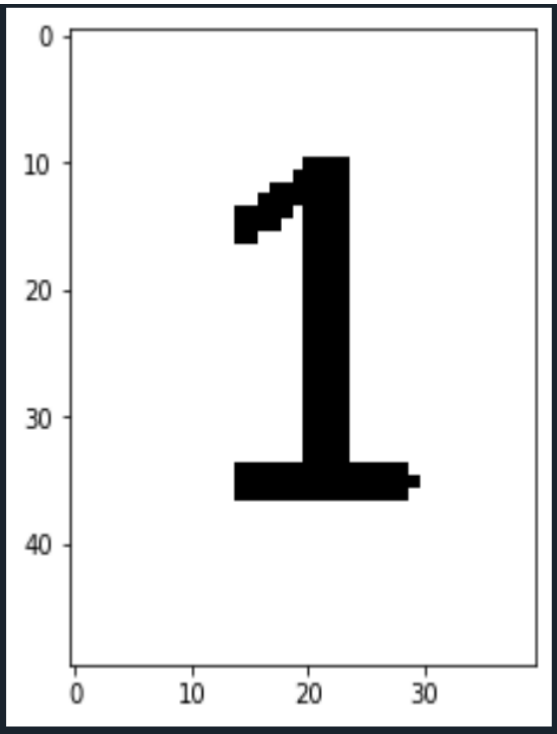
III – Travail préparatoire :

Question 2 :

La méthode Binarisation est écrit de la manière suivante :

```
    def binarisation(self, S):
        #Image vide créee
        im_2=Image()
        #Initialisation d'un tableau de pixels de même taille que self
        #Utilisation d'une matrice zero(cases nulles) , uint8 :type unsigned int de 8 bytes
        im_2.set_pixels(np.zeros(self.H,self.W), dtype = np.uint8 )
        #Parcours des lignes:i et des colonnes: j
        #Et Affectation de nouvelles valeurs a notre nouvelle image en fonction des pixels et du seuil
        for i in range(self.H):
            for j in range (self.W):
                if (self.pixels[i][j]<S):
                    self.pixels[i][j]=0
                else:
                    self.pixels[i][j]=255
        return im_2
```

-->En faisant le **test**, nous remarquons qu'il n'y a pas de problèmes pour la méthode binarisation. Cependant, ce sont les méthodes localisation,similitude et reconnaissance_chiffre qui renvoient des erreurs : chose évidente car nous n'avons pas encore codé celles-ci.



Help Variable Explorer Plots Files

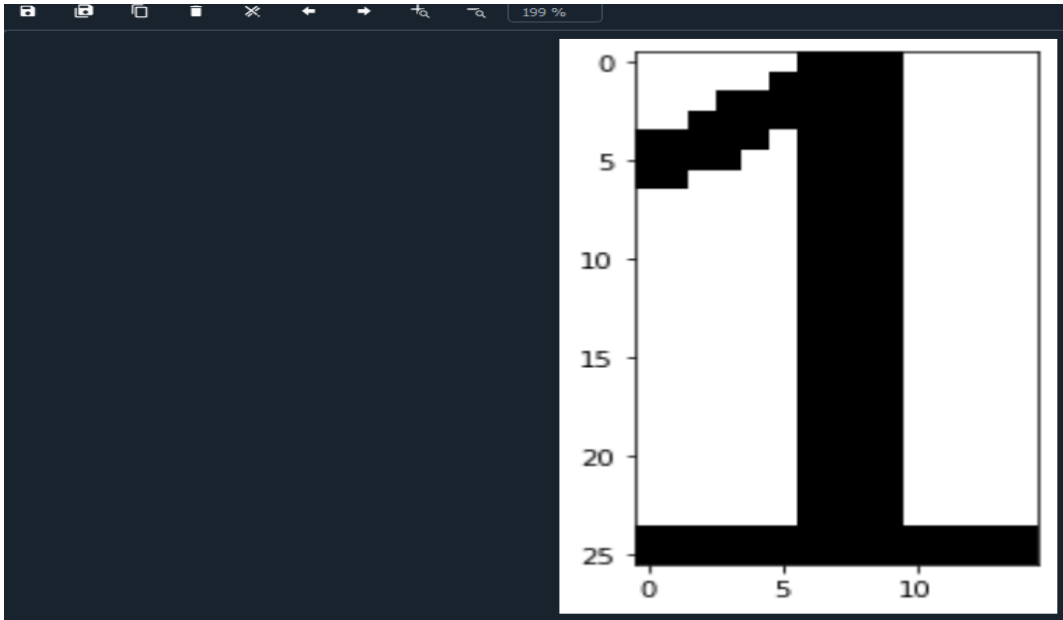
Console 1/A

```
AttributeError: 'NoneType' object has no attribute 'display'

In [21]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests/test_Image.py', wdir='E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-mai
.......Reloaded modules: image, reconnaissance
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test5.JPG (54x61)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test1.JPG (54x43)
.FFEEEEEEEEFFFFlecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)

======================================================================
ERROR: test_localisation_pixels_is_array (__main__.Test_Image_localisation)
Teste si les pixels de l'objet de type Image résultant de la méthode localisation
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 112, in test_localisation_pixels_is_array
    assert isinstance(image_localisee.pixels, np.ndarray)
AttributeError: 'NoneType' object has no attribute 'pixels'

======================================================================
```

Question 3 : La méthode localisation s'écrit de la manière suivante :

```
def localisation(self):
    l_min=self.H
    l_max=0
    c_min=self.W
    c_max=0
    #Parcours de l'image en fixant les lignes et en parcourant les colonnes
    #Cadrons l'image par suppression des lignes et des colonnes toutes blanches
    for i in range(self.H):
      for j in range (self.W):
        if self.pixels[i][j]==0:
            if j<c_min:
                c_min=j
            if j>c_max:
                c_max=j
            if i<l_min:
                l_min=i
            if i>l_max:
                l_max=i
    new_image=Image()
    #Affectation / supsets sauvegardant l'image
    #Cadrage
    new_image.set_pixels(self.pixels[l_min:l_max,c_min:c_max])
    return new_image
```

-->Les tests réalisés montrent bien que la méthode localisation marche. Seuls les méthodes resize, similitude et reconnaissance_chiffre affichent des erreurs : chose évidente car celles-ci ne sont pas codées.




```
Console 1/A
    image_resizee.display( image redimensionee )

AttributeError: 'NoneType' object has no attribute 'display'


In [26]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests/test_Image.py', wdir='E:/Numération-Tp/tp2-reconnaissance-
........Reloaded modules: reconnaissance, image
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test5.JPG (54x61)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
.....E.FFEFFFFlecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)

======================================================================
ERROR: test_resize_attributes (__main__.Test_Image_resize)
Teste si les attributs de l'objet de type Image résultat de la méthode resize
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 196, in test_resize_attributes
    assert image_resizee.H == new_H
AttributeError: 'NoneType' object has no attribute 'H'
```

IV - Reconnaissance automatique de chiffre :

Question 4-1 :

-Une image est définie par sa classe (classe Image) qui a comme attributs H et W pour présenter sa taille, aussi l'attribut pixel qui affiche sur un code binaire de 8bits l'intensité de couleur d'une partie d'image de ligne i entre [0, H-1] e colonne j entre [0-W-1].

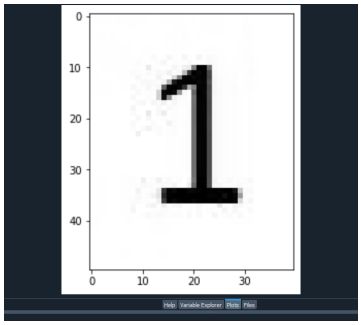
--> Les tests réalisés sur la binarisation n'ont posé aucun problème :

```
In [1]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests/test_Image.py', wdir='E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests')
.............lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test5.JPG (54x61)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)
E.FFEFFFFlecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)

======================================================================
ERROR: test_resize_attributes (__main__.Test_Image_resize)
Teste si les attributs de l'objet de type Image résultat de la méthode resize
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 196, in test_resize_attributes
    assert image_resizee.H == new_H
AttributeError: 'NoneType' object has no attribute 'H'

======================================================================
ERROR: test_resize_pixels_is_array (__main__.Test_Image_resize)
Teste si les pixels de l'objet de type Image résultant de la méthode resize
----------------------------------------------------------------------
```

Question 4-2 et 4-3 :

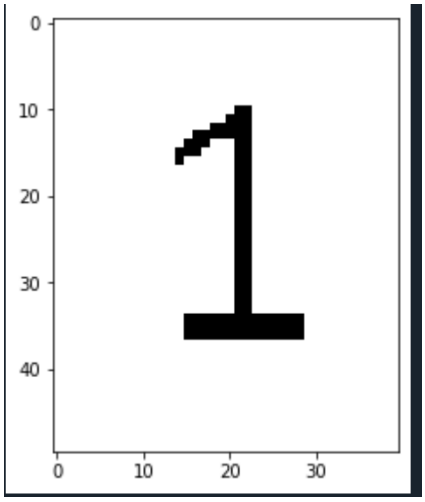
Les résultats obtenus avec différents seuils pour la binarisation et la localisation montrent que ces méthodes sont fonctionnelles.

Image avant binarisation



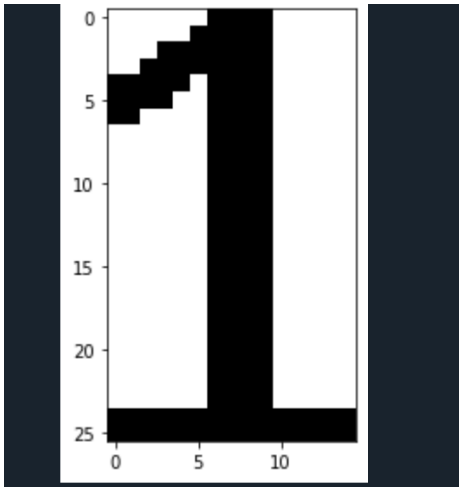
↓

Image avant binarisation



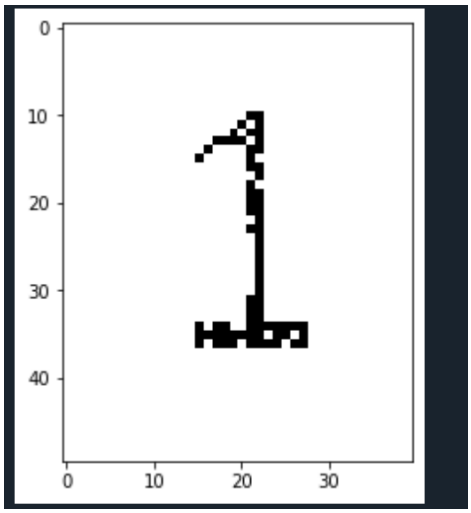
↓

Image après Localisation

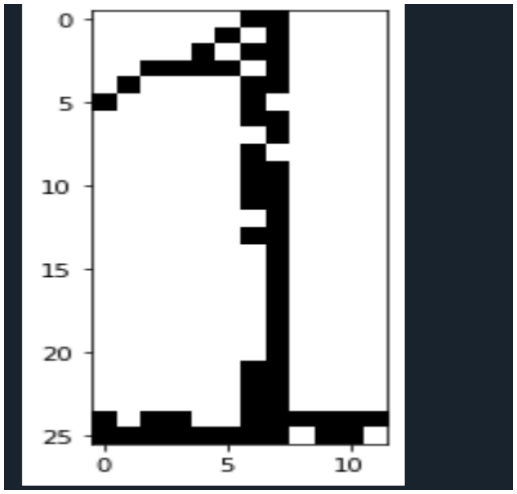


En changeant le Seuil :

→ **S=5**

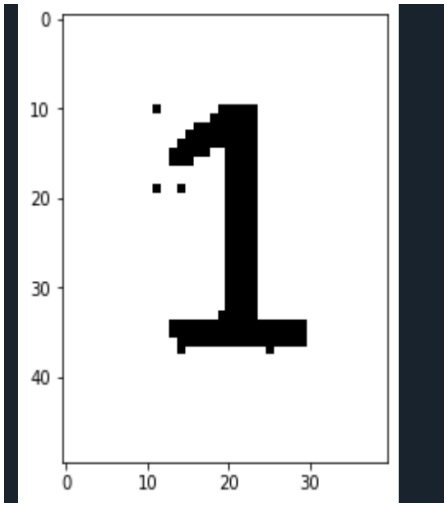


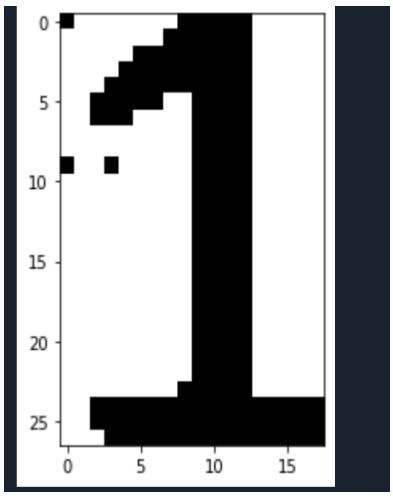
→



Binarisation **Localisation**

→ **S=240**





Binarisation **Localisation**

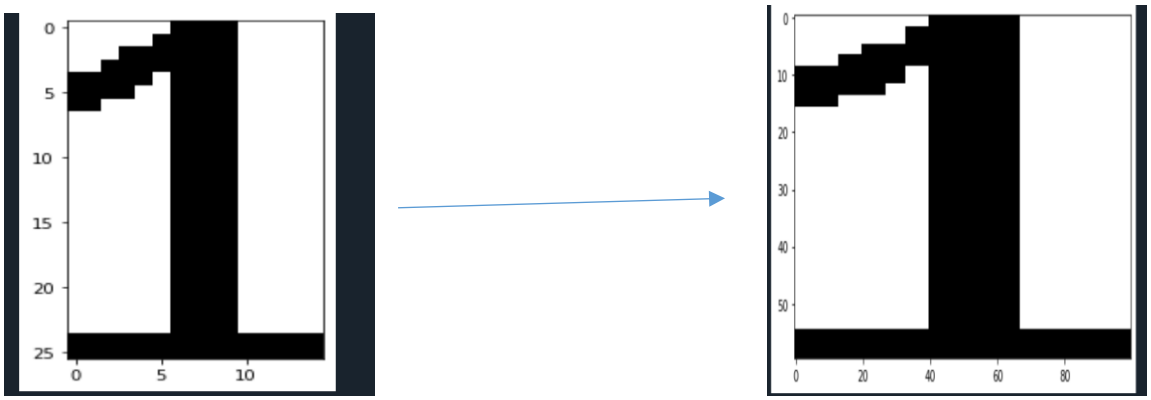
Question 4-2 : Les tests réalisés sur la méthode localisation n'ont posé aucun problème :

```
In [2]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests/test_Image.py', wdir='E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests')
.........Reloaded modules: image
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test5.JPG (54x61)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
....E.FFEFFFFlecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)

======================================================================
ERROR: test_resize_attributes (__main__.Test_Image_resize)
Teste si les attributs de l'objet de type Image résultat de la méthode resize
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 196, in test_resize_attributes
    assert image_resizee.H == new_H
AttributeError: 'NoneType' object has no attribute 'H'

======================================================================
ERROR: test_resize_pixels_is_array (__main__.Test_Image_resize)
Teste si les pixels de l'objet de type Image résultant de la méthode resize
```

Question 4-3 :

La méthode resize est écrit de la manière suivante. Les tests n'ont posé aucun souci majeur pour cette méthode :

```python
def resize(self, new_H, new_W):
    #Creation d'une nouvelle image
    new_image=Image()
    #Redimensionnement et multiplication par 255 pour convertir en int
    k=resize(self.pixels,(new_H,new_W),0)
    new_image.set_pixels(np.uint8(k*255))
    return new_image
```



-->Pour les tests on avait des problèmes au début avec les notations et avec a notion de ‘pixels’ mais après et avec les instructions données par le prof, on a pu résoudre le problème.

```
======================================================================
ERROR: test_resize_attributes (__main__.Test_Image_resize)
Teste si les attributs de l'objet de type Image résultat de la méthode resize
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 196, in test_resize_attributes
    assert image_resizee.H == new_H
AttributeError: 'NoneType' object has no attribute 'H'

======================================================================
ERROR: test_resize_pixels_is_array (__main__.Test_Image_resize)
Teste si les pixels de l'objet de type Image résultant de la méthode resize
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 184, in test_resize_pixels_is_array
    assert isinstance(image_resizee.pixels, np.ndarray)
AttributeError: 'NoneType' object has no attribute 'pixels'

======================================================================
FAIL: test_resize_is_Image (__main__.Test_Image_resize)
Teste si le résultat de resize est bien de la classe Image.
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 161, in test_resize_is_Image
    assert isinstance(image_resizee, Image)
AssertionError

======================================================================
FAIL: test_resize_is_not_none (__main__.Test_Image_resize)
Teste si le résultat de resize renvoie bien quelque chose.
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 150, in test_resize_is_not_none
    assert image_resizee is not None
AssertionError
```

```
In [5]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests/test_Image.py', wdir='E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua
........Reloaded modules: reconnaissance, image
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test5.JPG (54x61)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
.........FFFFlecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)

======================================================================
FAIL: test_similitude_is_not_none (__main__.Test_Image_similitude)
Teste la méthode similitude ne renvoie pas None (oublie de return...).
----------------------------------------------------------------------
Traceback (most recent call last):
  File "E:\Numération-Tp\tp2-reconnaissance-chiffres-tp2_gagua-main\tests\test_Image.py", line 206, in test_similitude_is_not_none
    assert image.similitude(image) is not None
AssertionError
```

Question 4-4 :

La méthode similitude est écrit de la manière suivante. Les tests n'ont posé aucun souci majeur :

```python
def similitude(self, im):
    #Initialisation
    x=0
    #Parcours des colonnes en fixant les lignes
    #Comparaisons de l'intensité de chaque 2 pixels de ligne 'i'et de colonne 'j'
    for i in range(self.H):
        for j in range(self.W):
            if self.pixels[i][j]==im.pixels[i][j]:
                x+=1
            else:
                x+=0
    return(x/(self.H*self.W))
```

```
In [16]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests/test_Image.py', wdir='E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests')
.........Reloaded modules: image
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test5.JPG (54x61)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
.............lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test4.JPG (58x40)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/test3.JPG (43x38)
lecture image : ../assets/test2.JPG (50x40)

----------------------------------------------------------------------
Ran 22 tests in 0.326s

OK

In [17]:
```

Question 4-5 :

La méthode reconnaissance_chiffre est écrit de la manière suivante. Les tests n'ont posé aucun souci majeur :

```python
def reconnaissance_chiffre(image, liste_modeles, S):
    #Binarisation + Localisation + Sauvegardes
    im_2 = image.binarisation(S)
    im_2_located = im_2.localisation()
    x = 0
    index = 0
    #Parcours d'une liste d'images modèles et sauvegarde de la similitude maximale ainsi l'indice de l'image avec la similitude.
    for i in range(len(liste_modeles)):
        result = im_2_located.resize(liste_modeles[i].H,liste_modeles[i].W)
        if result.similitude(liste_modeles[i])>x:
            x=result.similitude(liste_modeles[i])
            index=i
    return index
```

```
In [24]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests/test_reconnaissance.py', wdir='E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/tests')
.Reloaded modules: image
lecture image : ../assets/_0.png (32x22)
lecture image : ../assets/_1.png (32x18)
lecture image : ../assets/_2.png (32x20)
lecture image : ../assets/_3.png (32x20)
lecture image : ../assets/_4.png (32x24)
lecture image : ../assets/_5.png (32x20)
lecture image : ../assets/_6.png (32x21)
lecture image : ../assets/_7.png (32x21)
lecture image : ../assets/_8.png (32x22)
lecture image : ../assets/_9.png (32x22)
lecture image : ../assets/test2.JPG (50x40)
lecture image : ../assets/_0.png (32x22)
lecture image : ../assets/_1.png (32x18)
lecture image : ../assets/_2.png (32x20)
..lecture image : ../assets/_3.png (32x20)
lecture image : ../assets/_4.png (32x24)
lecture image : ../assets/_5.png (32x20)
lecture image : ../assets/_6.png (32x21)
lecture image : ../assets/_7.png (32x21)
lecture image : ../assets/_8.png (32x22)
lecture image : ../assets/_9.png (32x22)
lecture image : ../assets/test1.JPG (54x43)
lecture image : ../assets/_0.png (32x22)
lecture image : ../assets/_1.png (32x18)
lecture image : ../assets/_2.png (32x20)
lecture image : ../assets/_3.png (32x20)
lecture image : ../assets/_4.png (32x24)
lecture image : ../assets/_5.png (32x20)
lecture image : ../assets/_6.png (32x21)
lecture image : ../assets/_7.png (32x21)
lecture image : ../assets/_8.png (32x22)
lecture image : ../assets/_9.png (32x22)
lecture image : ../assets/test3.JPG (43x38)
```

Question 4- 6 :

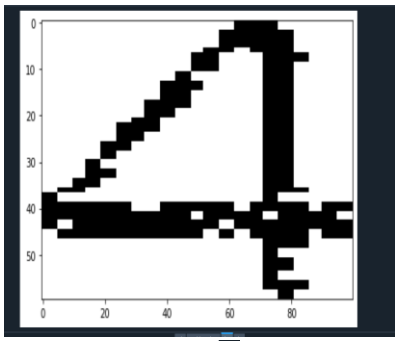
- **Test 1 :** Le test montre le chiffre 4. La reconnaissance de ce chiffre s'est faite sans problème

```
image = Image()
image.load(path_to_assets + 'test1.JPG')
image.display("Exemple d'image")

#==========================================================================
# Binarisation de l'image et affichage
#==========================================================================
S =10
image_binarisee = image.binarisation(S)
image_binarisee.display("Image binarisee")

#==========================================================================
# Localisation de l'image et affichage
#==========================================================================
image_localisee = image_binarisee.localisation()
image_localisee.display("Image localisee")

#==========================================================================
# Redimensionnement de l'image et affichage
#==========================================================================
image_resizee = image_localisee.resize(60, 100)
image_resizee.display("Image redimensionee")

#==========================================================================
# Lecture modeles et reconnaissance
#==========================================================================
liste_modeles = lecture_modeles(path_to_assets)
chiffre = reconnaissance_chiffre(image, liste_modeles, 70)
print("Le chiffre reconnu est : ", chiffre)
```

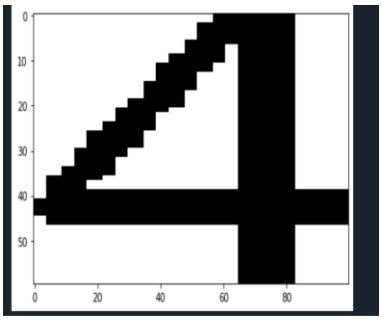
-->On peut dire que en augmentant le seuil l'image est plus en plus contraste et on aura plus de retouches noires sur l'image. Par contre les seuils faibles donnent des images qui peut être parfois non lisible car si le seuil est très faible donc le code va éliminer la plupart des pixels.

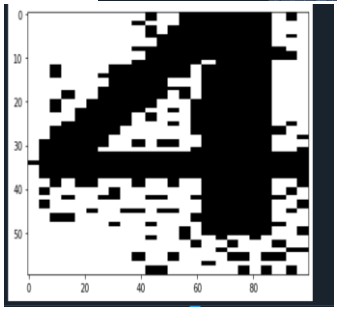
S=230

S=10



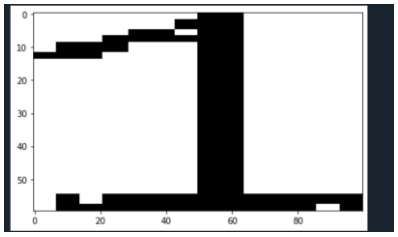
S=130.



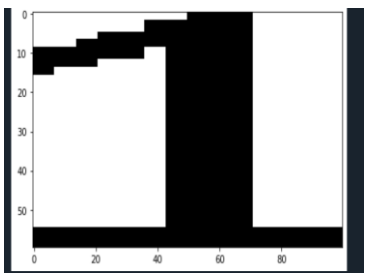


- **Test 2 :** La reconnaissance du chiffre 1 également n'a pas posé problème dans le main:

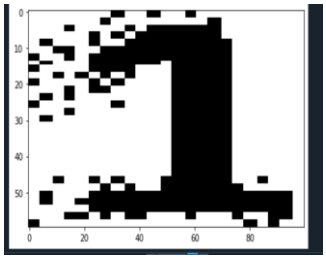
S=30



S=130

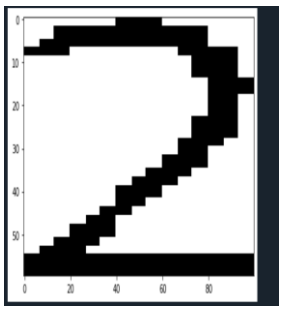
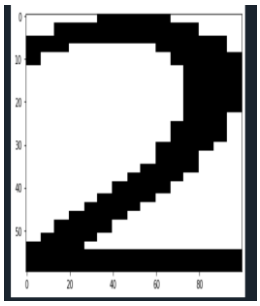
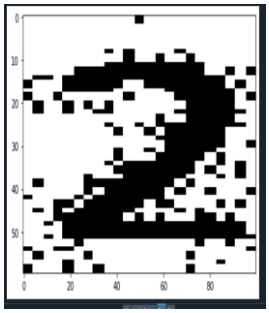


S=250



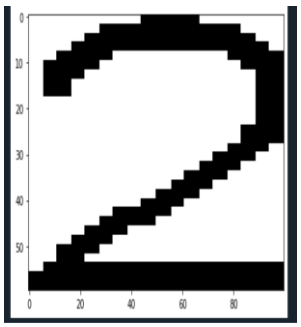
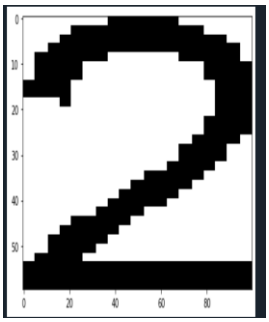
- **Test 3 :** La reconnaissance du chiffre 2 n'a posé aucun problème dans le main

S=30 S=130 S=250

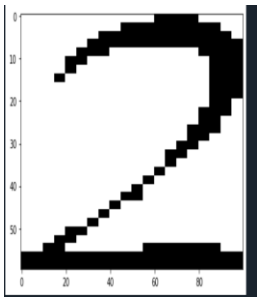
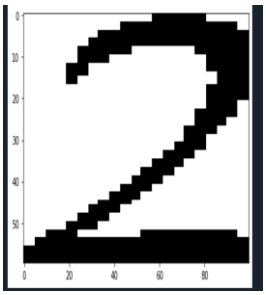
- **Test 4 :** Même chiffre que le test 3. La reconnaissance n'a pas posé de problème

S=30 S=130 S=250

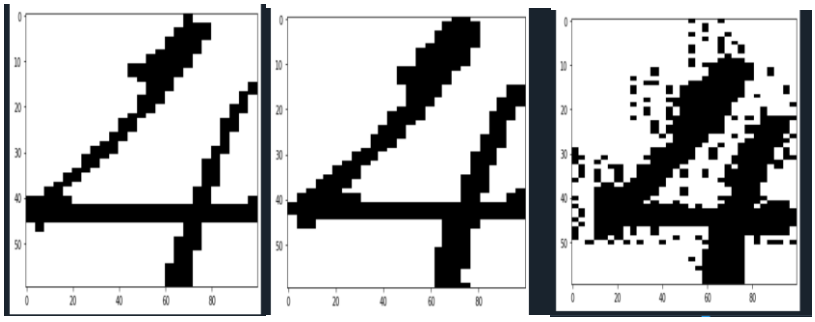

- **Test 5 :** Même chose que les Test 4 et 3

S=30 S=130 S=250

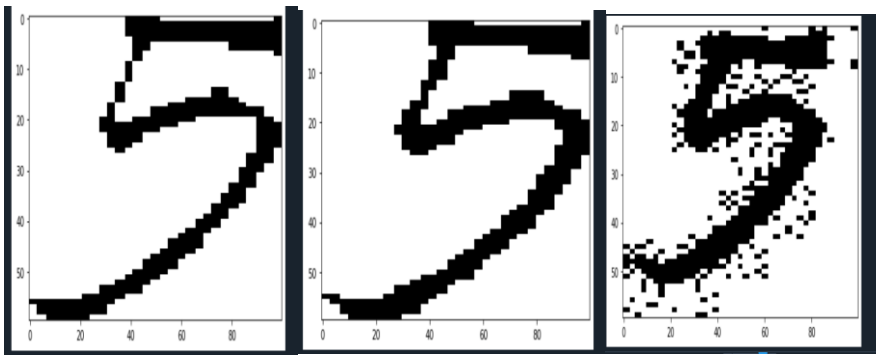

- **Test 6 :** La reconnaissance du chiffre 4 n'a posé aucun problème

S=30 S=130 S=250



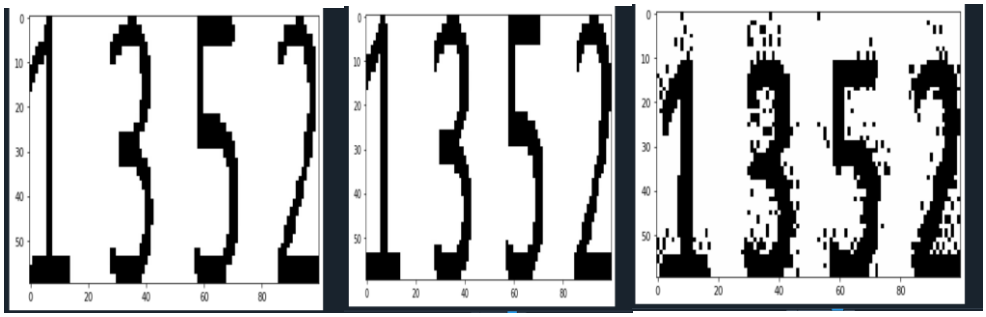
- **Test 7 :** La reconnaissance du chiffre 5 n'a posé aucun problème

S=30 S=130 S=250



- **Test 8 :** La reconnaissance pour ce test n'a pas fonctionné par contre

S=30 S=130 S=250

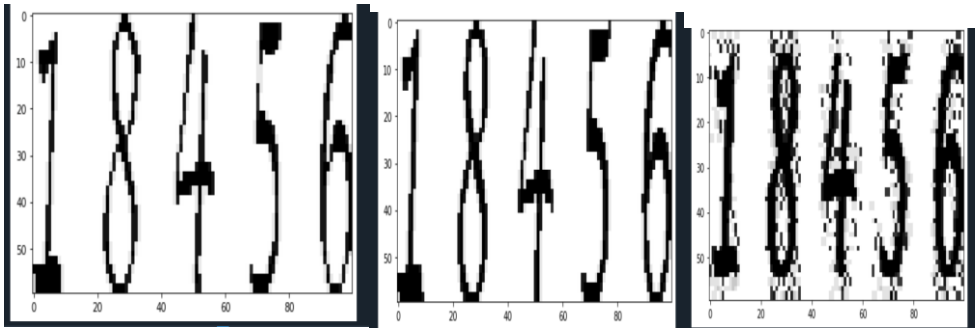


```
In [27]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/src/main.py', wdir='E:/Numération-Tp/tp2-reconnaissance-chi
Reloaded modules: image, reconnaissance
lecture image : ../assets/test9.JPG (66x239)
lecture image : ../assets/_0.png (32x22)
lecture image : ../assets/_1.png (32x18)
lecture image : ../assets/_2.png (32x20)
lecture image : ../assets/_3.png (32x20)
lecture image : ../assets/_4.png (32x24)
lecture image : ../assets/_5.png (32x20)
lecture image : ../assets/_6.png (32x21)
lecture image : ../assets/_7.png (32x21)
lecture image : ../assets/_8.png (32x22)
lecture image : ../assets/_9.png (32x22)
Le chiffre reconnu est :  0
```

--> Le code qu'on a mis a la capacité de connaitre un seul nombre, et c'est justement pourquoi le code n'a pas reconnu aucun chiffre pour ce test.

- **<u>Test 9 :</u>** La reconnaissance n'a pas fonctionné ici non plus

S=30 S=130 S=250

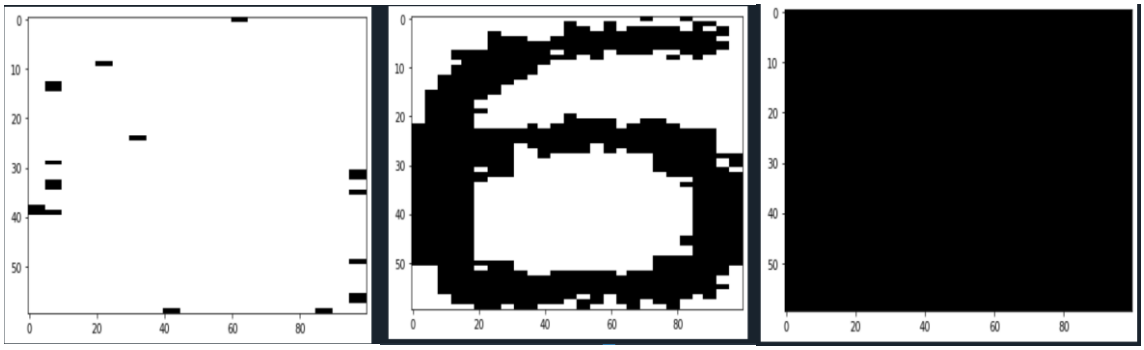


```
In [27]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/src/main.py', wdir='E:/Numération-Tp/t
Reloaded modules: image, reconnaissance
lecture image : ../assets/test9.JPG (66x239)
lecture image : ../assets/_0.png (32x22)
lecture image : ../assets/_1.png (32x18)
lecture image : ../assets/_2.png (32x20)
lecture image : ../assets/_3.png (32x20)
lecture image : ../assets/_4.png (32x24)
lecture image : ../assets/_5.png (32x20)
lecture image : ../assets/_6.png (32x21)
lecture image : ../assets/_7.png (32x21)
lecture image : ../assets/_8.png (32x22)
lecture image : ../assets/_9.png (32x22)
Le chiffre reconnu est :  0
```

--> Chiffre non reconnu, même cas précèdent.

- **Test 10 :**

S=30 S=130 S=250



```
In [32]: runfile('E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/src/main.py', wdir='E:/Numération-Tp/tp2-reconnaissance-chiffres-tp2_gagua-main/src')
Reloaded modules: image, reconnaissance
lecture image : ../assets/test10.JPG (80x45)
lecture image : ../assets/_0.png (32x22)
lecture image : ../assets/_1.png (32x18)
lecture image : ../assets/_2.png (32x20)
lecture image : ../assets/_3.png (32x20)
lecture image : ../assets/_4.png (32x24)
lecture image : ../assets/_5.png (32x20)
lecture image : ../assets/_6.png (32x21)
lecture image : ../assets/_7.png (32x21)
lecture image : ../assets/_8.png (32x22)
lecture image : ../assets/_9.png (32x22)
Le chiffre reconnu est :  6
```

--> l'image est un peu floue parceque les intensités des pixels de l'image sont trop élevés et c(est justement pourquoi on a une image toute noire dans les seuils les plus élevés.

IV - Conclusion :

La reconnaissance par analyse d'image sur python est très utile et efficace. Cependant, nous avons remarqué que pour les images constituées de plusieurs chiffres, il n'y a pas eu de reconnaissance. Cela représente une limite des méthodes que nous avons programmé. Il serait intéressant de proposer une méthode qui analysera l'image de manière fractionnée afin de reconnaitre chaque chiffre constituant ainsi un nombre. Il faudrait également trouver des solutions/méthodes quant aux images similaires à celui du test 10 où l'arrière-plan n'est pas forcément blanc mais gris.