WOLF Manon
GIGANTE Elodie
Date du TP : 03/12/2021

Rapport de TP2 – Lecture automatique de chiffres par analyse d'image

I. Introduction

Dans ce TP, on va chercher à manipuler des images sous forme de tableaux de pixels, puis utiliser les fonctions trouvées pour simuler un système d'OCR.

III. Travail préparatoire

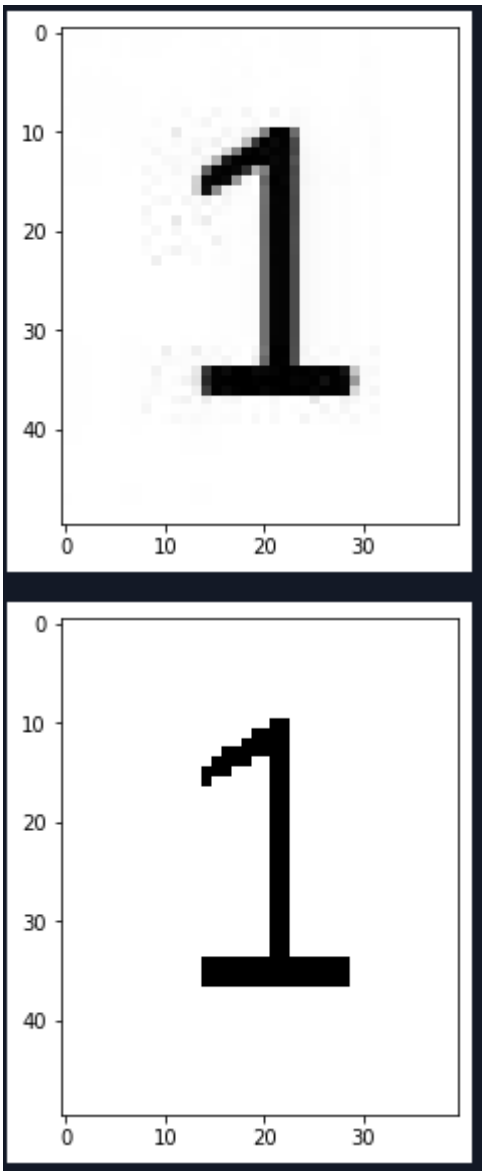
Question 2

```
def binarisation(self, S):
    # creation d'une image vide
    im_bin = Image()

    # affectation a l'image im_bin d'un tableau de pixels de meme taille
    # que self dont les intensites, de type uint8 (8bits non signes),
    # sont mises a 0
    im_bin.set_pixels(np.zeros((self.H, self.W), dtype=np.uint8))

    # TODO: boucle imbriquees pour parcourir tous les pixels de l'image im_bin
    # et calculer l'image binaire
    for i in range(self.H):
        for j in range(self.W):
            if self.pixels[i][j] >= S:
                im_bin.pixels[i][j] = 255
            else:
                im_bin.pixels[i][j] = 0
    return im_bin
```

On parcourt tous les pixels du tableau et on compare leur valeur à S. On attribue la nouvelle valeur en fonction

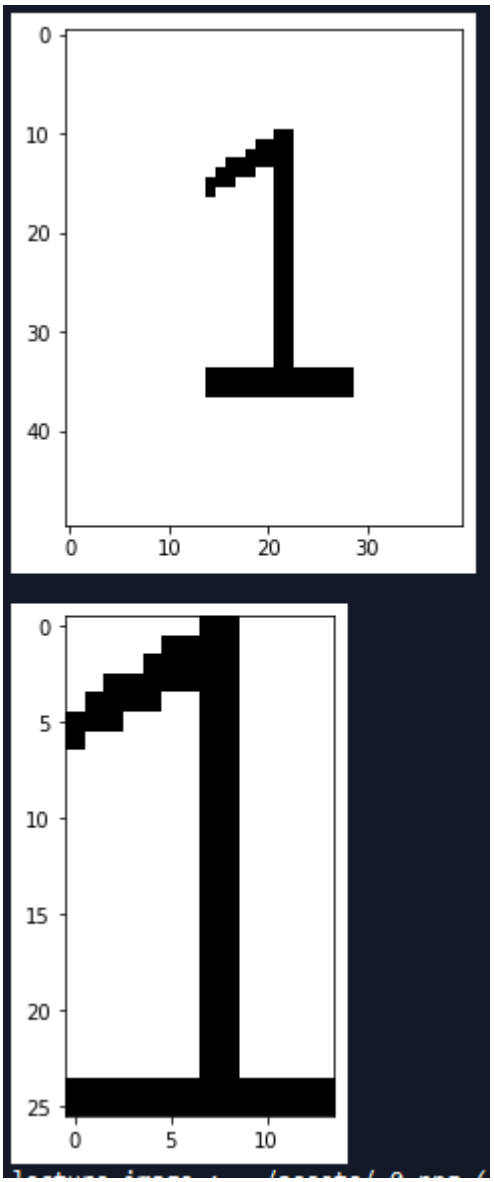


Question 3

```
def localisation(self):
        lim_gauche = self.W
        lim_haut = self.H
        lim_droite = 0
        lim_bas = 0
        for i in range(self.H):
            for j in range(self.W):
                if self.pixels[i][j] == 0 and i < lim_gauche:
                    lim_gauche = i
                if self.pixels[i][j] == 0 and i > lim_droite:
                    lim_droite = i
                if self.pixels[i][j] == 0 and j < lim_haut:
```

```
            lim_haut = j
        if self.pixels[i][j] == 0 and j > lim_bas:
            lim_bas = j
im = Image()
im.set_pixels(self.pixels[lim_gauche:lim_droite,lim_haut:lim_bas])

return im
```



IV. Reconnaissance automatique de chiffres

Question 3

```
def resizee(self, new_H, new_W):
    pixels_resized = resize(self.pixels, (new_H,new_W), 0)
    im_resized = np.uint8(pixels_resized*255)
    im = Image()
    im.set_pixels(im_resized)
    return im
```

On applique la fonction resize, en redimensionnant le résultat pour avoir le bon format.

Question 4

```
def similitude(self, im):
    similitude = 0
    if self.H != im.H or self.W != im.W:
        print('Les images n ont pas la meme taille')
    else:
        for i in range(self.H):
            for j in range(self.W):
                if self.pixels[i][j] == im.pixels[i][j]:
                    similitude += 1
                    taux = similitude/(self.H*self.W)
        return taux
```

On compare un à un tous les pixels des deux images, en incrémentant chaque fois qu'ils sont identiques. On divise le résultat par la taille de l'image pour avoir un taux de similitude. On a également ajouté une vérification de la taille des images au début.

Question 5

```
def reconnaissance_chiffre(image, liste_modeles, S):
    im_b = image.binarisation(S)
    im_loc = im_b.localisation()
    simMax=0.0
    for i in range(len(liste_modeles)):
        newH=liste_modeles[i].H
        newW=liste_modeles[i].W
        im_res = im_loc.resizee(newH, newW)
        taux_Sim = im_res.similitude(liste_modeles[i])
        if taux_Sim > simMax :
            simMax=im_res.similitude(liste_modeles[i])
```

```
            iMax = i
    return iMax
```

On applique d'abord la binarisation, puis la localisation. Ensuite, on fait une boucle pour adapter la taille de l'image à chaque modèle et calculer la similitude. Enfin, on garde et renvoie l'indice du modèle ayant donné la plus grande similitude.

Question 6

Tableau récapitulatif des essais :

Test	Seuil	Résultat
5	40	2
5	10	2
5	100	2
7	100	7
7	50	7
7	20	7
10	40	4
10	180	8
10	70	6

Nous remarquons que pour certains essais, le programme ne reconnait pas le bon chiffre lorsqu'il n'y a pas le bon seuil. En général, plus le chiffre est net sur l'image, plus le programme le reconnaît facilement.

IV. Conclusion

Nous avons réussi à terminer entièrement le TP sans trop de difficultés. Ce TP nous a permis de pratiquer l'utilisation des fonctions, méthodes et classes sans les confondre, ainsi que de se rendre compte de l'utilité de la programmation appliquée à un problème assez complet. Nous avons pu aussi observer les résultats de nos codes grâce aux images.