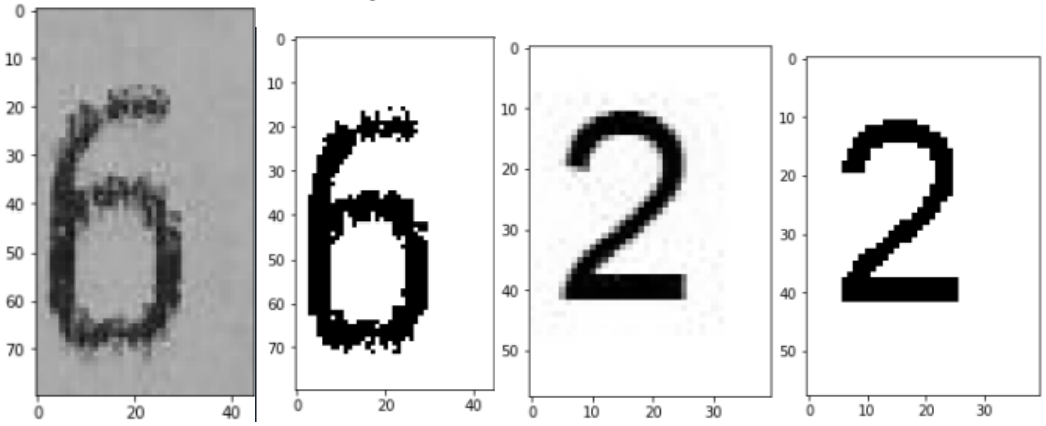
BLATRIX Alexis
BUSSEUIL Lucas
24/11/2021

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

III. Travail préparatoire

2. La méthode binarisation a pour attribut une image avec des pixels ayant des valeurs comprises entre 0 et 255, ainsi qu'un seuil de binarisation. La méthode renvoie cette image modifiée avec des pixels ayant désormais pour valeur 0 ou 255. Le seuil de binarisation impose la limite où chaque pixel ayant une valeur inférieure aura comme nouvelle valeur 0, et 255 pour les pixels ayant des valeurs supérieures au seuil. Cela nous permet d'avoir une image en noir et blanc.



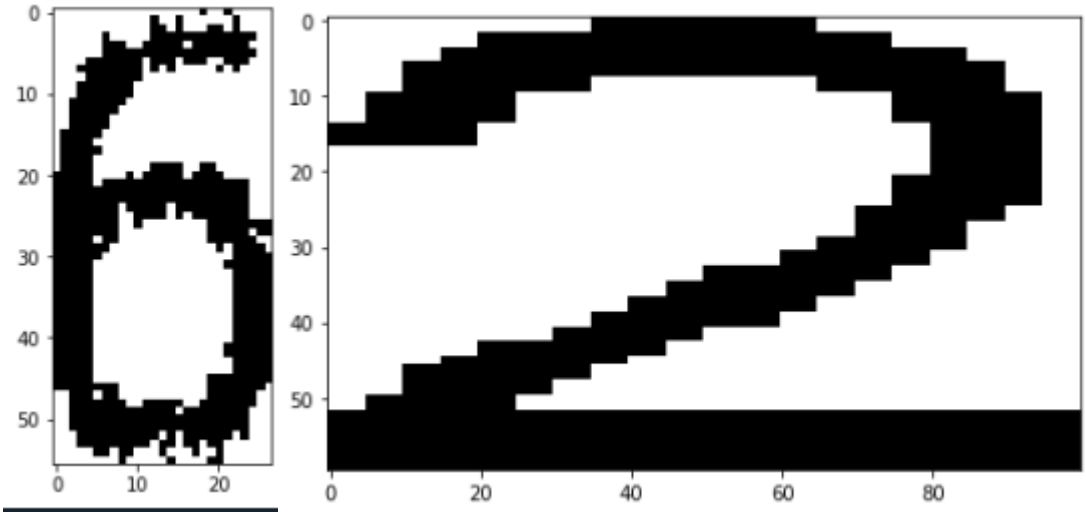
3. La fonction localisation se sert de l'emplacement des derniers pixels de couleur noir (<127 pour nous) sur la largeur et la hauteur pour délimiter l'emplacement du chiffre à l'intérieur de l'image. On utilise donc 2 boucles pour naviguer sur les pixels de la largeur et de la longueur afin de déterminer le point maximal auquel on a un pixel noir.

IV. Reconnaissance automatique de chiffre

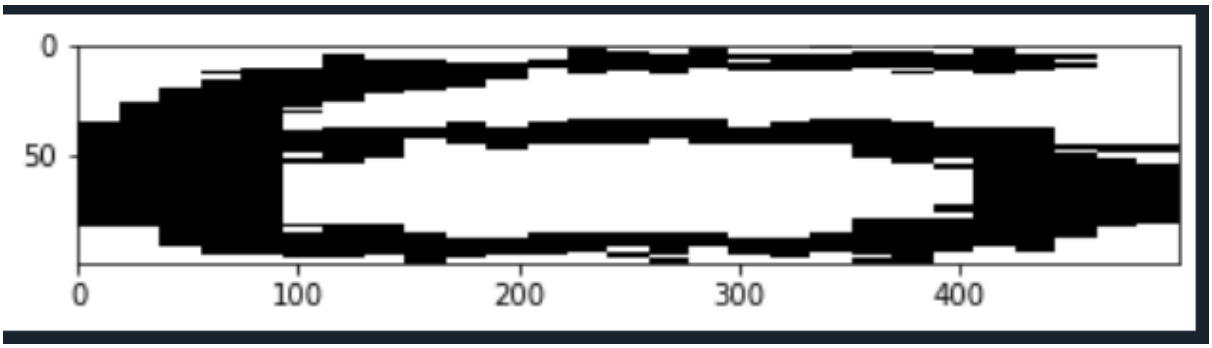
1. Ci-dessous le résultat du fichier test_image.

```
----------------------------
Ran 22 tests in 0.587s

OK

In [179]:
```

2. Ci-dessous, 2 exemples de la fonction localisation :



3. Pour changer la taille d'une image on utilise simplement la fonction resize existante afin de modifier la taille de l'image (hauteur, largeur) avec les données entrées par l'utilisateur.



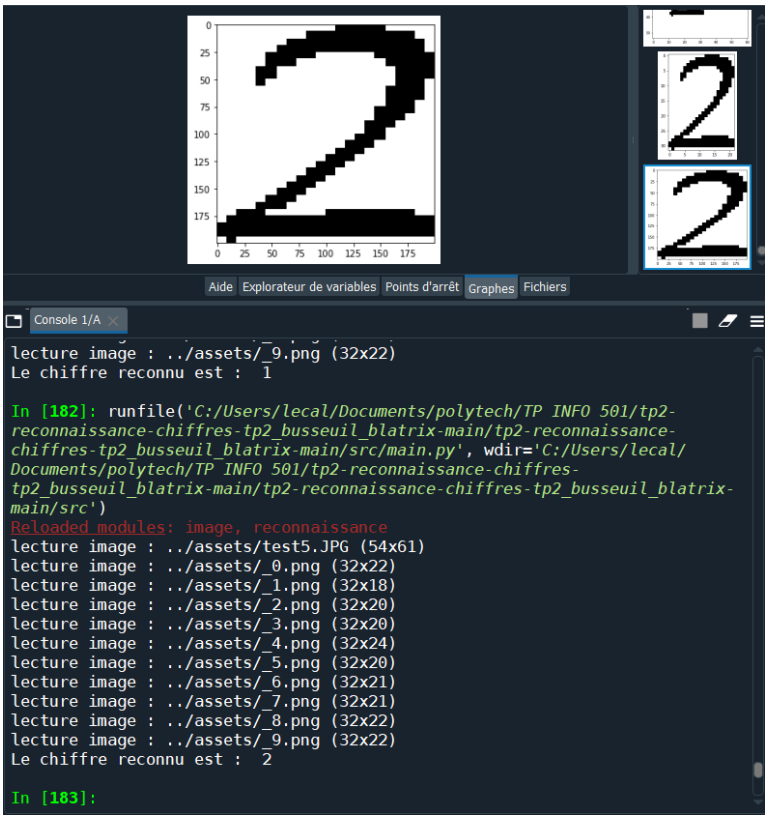
4. Similitude compare 2 images : elle les remet à la même taille en utilisant resize, les convertit en binaire avec binarisation puis avec l'aide de 2 boucles for compare chaque pixel des 2 images entre eux. La fonction renvoie un chiffre compris entre 0 et 1, plus le chiffre est élevé plus la similitude est élevée.

5. La méthode reconnaissance_chiffre va comparer une image préalablement binarisée ainsi que localisé avec une liste d'images contenant les chiffres de 0 à 9. Pour chaque comparaison, le niveau de similitude est évalué et comparé avec le niveau de similitude pour l'image précédente. A la fin la fonction nous renvoie l'indice de l'image avec le niveau de similitude le plus élevé. Ci-dessous, le résultat du fichier test_reconnaissance. :

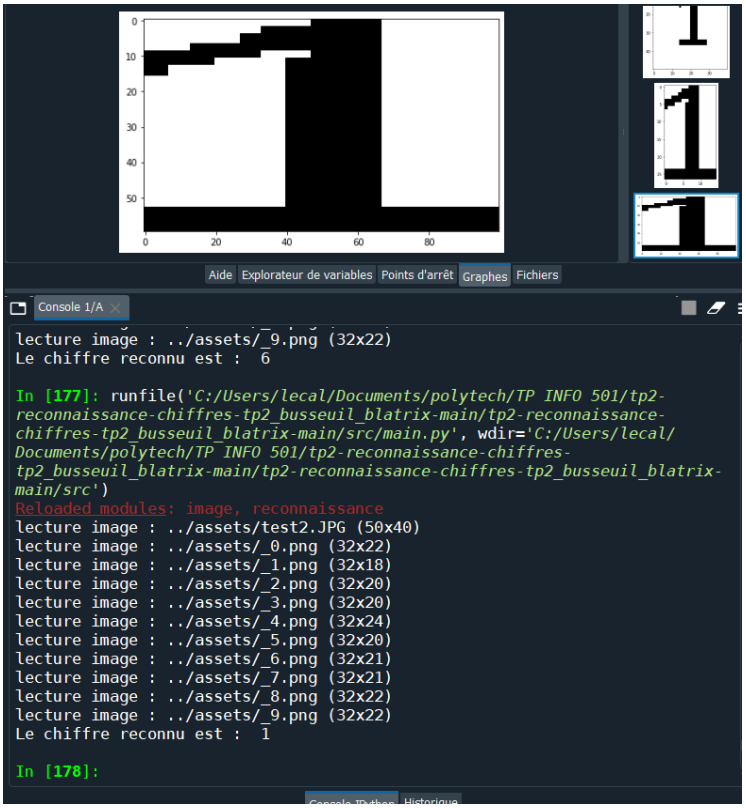
```
----------------------------------------------------------------------
Ran 3 tests in 0.268s

OK

In [180]:
```

6. Voici différents exemples de la méthode reconnaissance

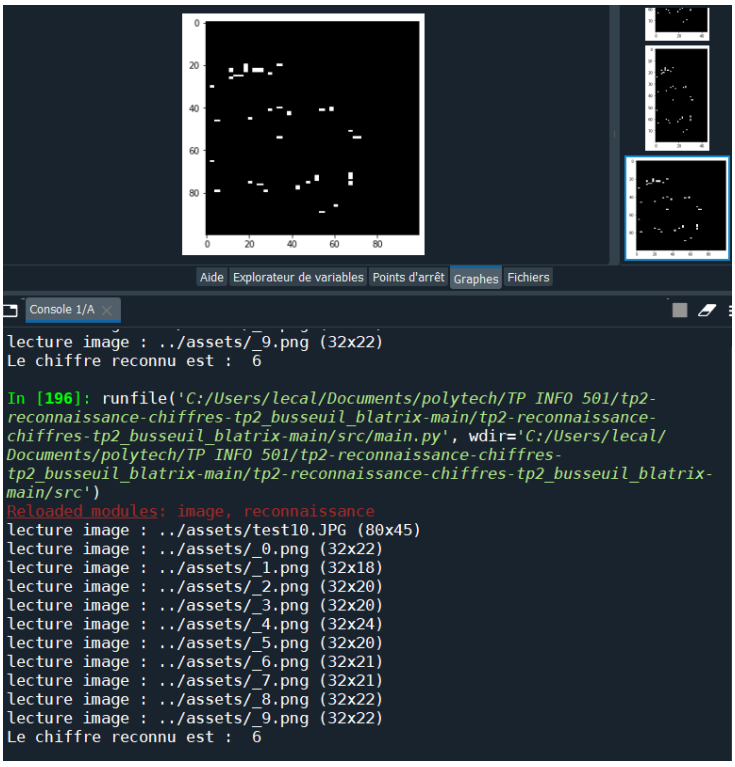
Size 100/100, Seuil 127



Size 100/100, Seuil 127



Size 100/100, Seuil 127



Size 100/100, Seuil 180, Chiffre 6

Size 100/100, Seuil 50

Pour nous le seuil parfait fut 127 (255/2). Nous n'avons pas eu de soucis en utilisant ce seuil et il parait plus logique d'utiliser une moyenne pour définir des niveaux entre 2 couleurs comme le noir et le blanc.

Conclusion

Ce TP nous a permis de découvrir les notions de modification et de comparaison d'images sur Python. Il était intéressant de jouer sur les données de modification afin de voir les différents seuils avant qu'une image change entièrement de couleur. A première vue le TP aurait pu paraitre compliqué mais il s'est finalement avéré à la fois simple et d'actualité (reconnaissance d'image).