PRUVOST Jordan
PAULIN Maxime
Date du TP (19/11/2021)

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

I. Introduction

L'objectif du TP est d'analyser une image (ici celle d'un chiffre) à partir de la création d'une classe **Image** et d'en analyser ses aspects afin de déterminer à quel chiffre correspond l'image.

II. Travail préparatoire

1. Question (1).

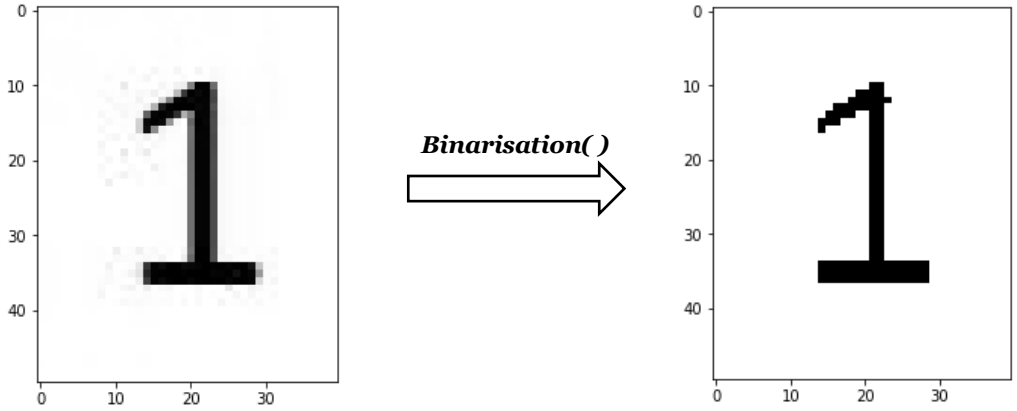
Les différents attributs de la classe **Image** sont sa hauteur (**H**), sa largeur (**W**) et elle est représenté par un tableau qui à chaque valeur de celui-ci, représente un pixel (ainsi que ses informations) en 2D.

2. Question (2).

L'objectif est d'implémenter à la classe **Image** une méthode de binarisation de l'image afin que cette dernière soit plus simple à traiter.

Cette méthode consiste à analyser chaque pixel de l'image et de lui assigner une nouvelle valeur en fonction d'un seuil **S** fixé. Si la valeur du pixel est supérieure au seuil fixé, le pixel prendra la valeur 255 (et sera donc blanc), sinon, il prendra la valeur noire (et sera donc noir). On procède par une boucle for qui parcourt chaque pixel de la ligne (avec **W** pour valeur maximale) et chaque colonne (avec **H** pour valeur maximale).

Ainsi on obtient la nouvelle image suivante :



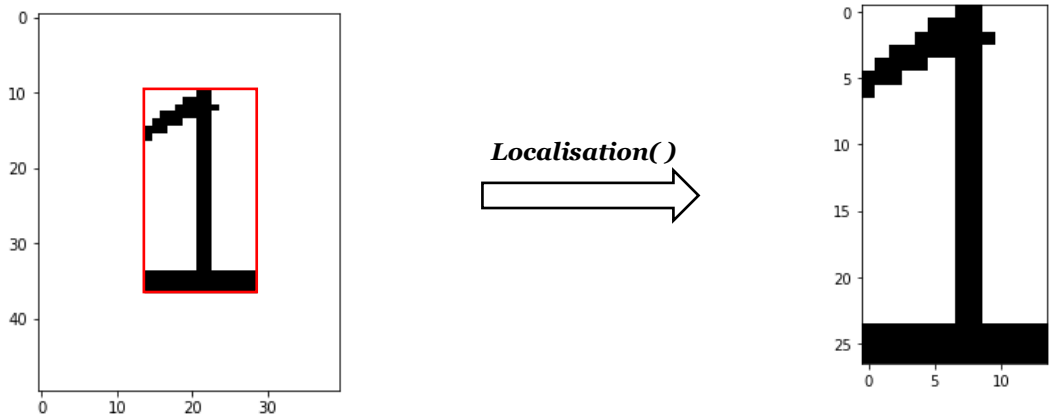
On remarque ainsi ici que l'image est désormais basée que sur deux valeurs de pixels

3. Question (3).

L'objectif de cette question est de créer une méthode permettant de recadrer l'image sur le chiffre.

Cette méthode consiste à analyser chaque ligne de l'image et de vérifier si oui ou non il y a présence d'un pixel noir sur cette dernière. Dans le cas contraire, on passe à la ligne suivante et on effectue le même processus jusqu'à ce que la ligne arrive à la valeur **H**. Afin d'évité les problèmes de recadrage de l'image lorsqu'une ligne est totalement blanche, nous avons privilégié une seconde analyse par le bas de l'image et de vérifier si chaque ligne possède un pixel noir ou non, et ainsi de remonter les lignes jusqu'au haut de l'image. La méthode a également effectué avec la gauche et la droite de l'image, mais cette fois ci en analysant les colonnes plutôt que les lignes.

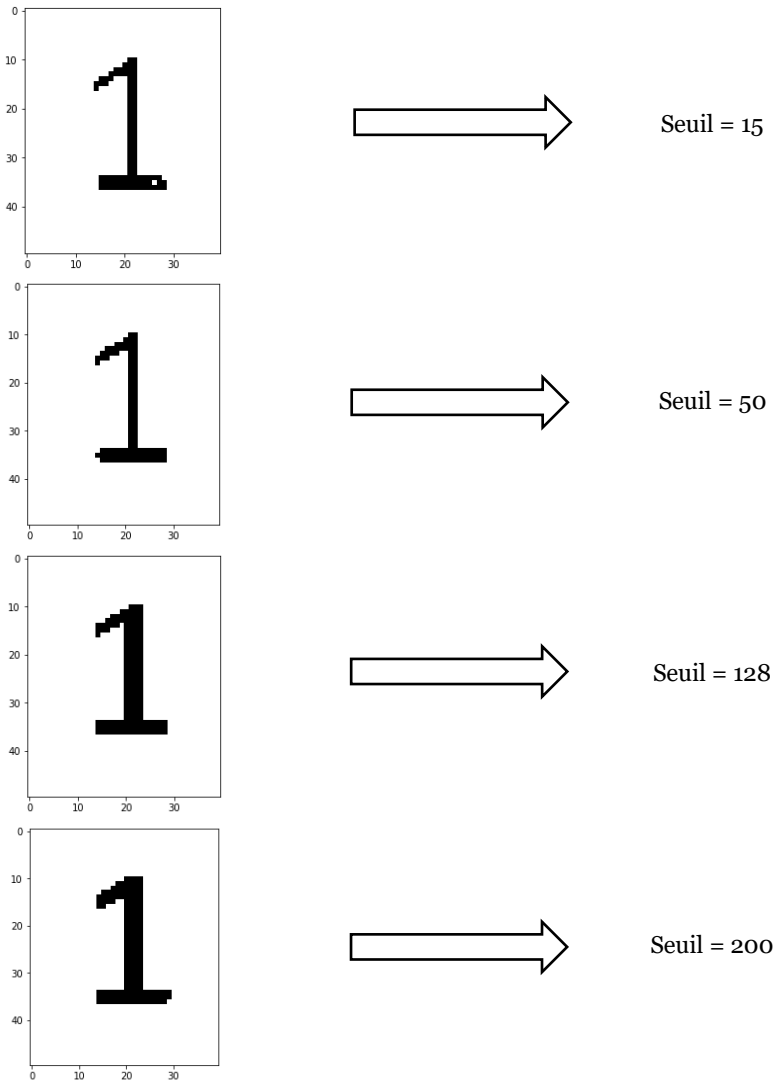
Ainsi, après recadrage, l'image ressemble à ceci :



On observe ici que l'image du chiffre 1 a été recadrée à partir des positions $l_{min} = 10$, $l_{max} = 37$, $c_{min} = 14$ et $c_{max} = 28$

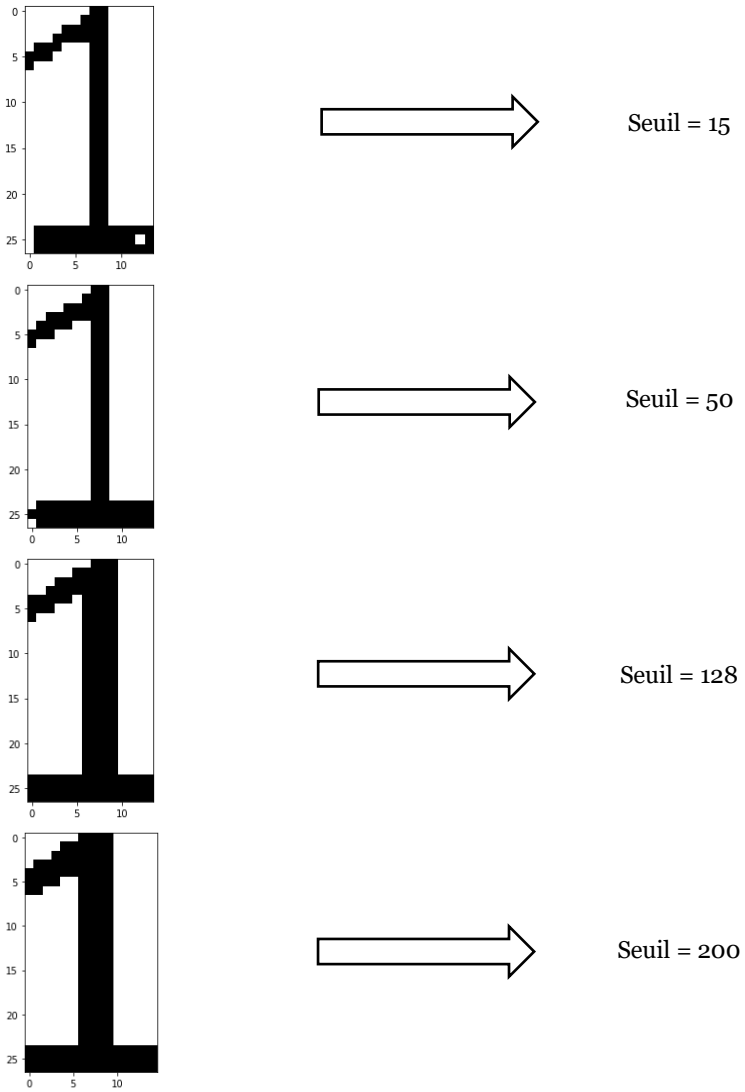
III. Reconnaissance automatique de chiffre

1. Question (1).



On remarque que plus le seuil est élevé, plus le contraste entre les pixels noirs et blancs va être important.

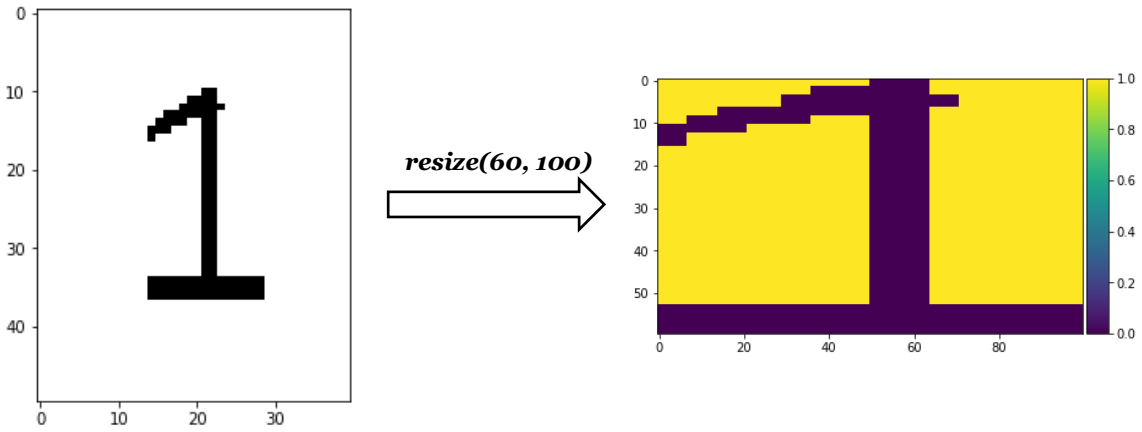
2. Question (2).



On remarque ici qu'en fonction du seuil, les position max et min changent légèrement en prenant respectivement des valeurs plus petites et plus grandes.

3. Question (3).

L'objectif est d'implémenter à la classe **Image** la méthode **resize** qui redimensionne simplement l'image en y affectant de nouvelle valeur pour **H** et **W**.



On remarque que l'image a simplement été redimensionnée et qu'une nouvelle a été créée.

4. Question (4).

L'objectif ici est d'applique la méthode **similitude** afin de comparer avec corrélation, deux images entre elle.

On effectue donc, pour chaque ligne et chaque colonne du tableau, représentant ici chaque pixel de l'image une comparaison et on décompte le nombre de pixels identiques entre les deux images pour une valeur de **H** et de **W** donnée.

5. Question (5).

L'objectif est de comparer chaque image de chiffre avec l'image donnée et de les comparer entre elle grâce à **similitude** en ayant au préalable appliquer les méthodes de **binarisation** et **localisation** pour l'image que l'on veut analyser. On calcule par la suite pour chaque image un taux de similitude avec l'image à analyser et on en déduit celle qu'elle représente le mieux. Ici, on détermine quel chiffre représente l'image en prenant la plus grande valeur présente dans la liste de **lecture_modeles**.

```
In [17]: runfile('C:/Users/jorda/OneDrive/Bureau/tp2-reconnaissance-chiffres-
tp2_pruvostjordan_paulinmaxime-main/src/main.py', wdir='C:/Users/jorda/OneDrive/
Bureau/tp2-reconnaissance-chiffres-tp2_pruvostjordan_paulinmaxime-main/src')
Reloaded modules: image, reconnaissance
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)
Le chiffre reconnu est :  1

In [18]:
```

On remarque que, par suite des analyses des images, le chiffre reconnu est le **1**.

IV. Conclusion

En conclusion, ce TP nous a permis de mieux appréhender le traitement et la comparaison d'image sous Python ; c'est une méthode qui peut être très intéressante, notamment dans le concept de captcha mais également dans de nombreux autres domaines de l'informatique.

Quelques difficultés nous ont également fait défaut nous empêchant d'avancer dans la suite, comme de la **localisation** de la partie préparatoire où le retour de notre fonction au format **array** était faussé par une mauvaise mise en forme de notre part, ne permettant pas un recadrage correct de l'image.

Dans l'ensemble, ce travail a été très instructif et a élargie nos notions en termes d'utilisation de l'outils Python.