

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

### I. Introduction

Ce TP est à propos de la lecture automatique de chiffres par analyse d'image, on doit, à partir d'une image en entrée, faire un programme permettant d'analyser une image et de reconnaître le chiffre contenu sur celle-ci.

### II. Travail préparatoire

#### 1. Question (2).

On compare chaque valeur de pixel de `self.pixels` à la valeur seuil `S`, si la valeur du pixel est supérieur il sera blanc, sinon il sera noir. On rentre ensuite les nouvelles valeurs dans une nouvelle image via la méthode `set_pixels`

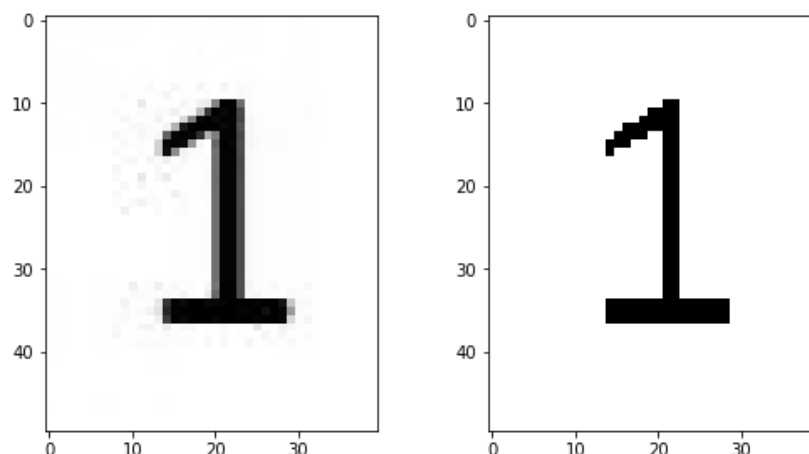
#### 2. Question (3).

On commence par trouver les valeurs de `i` en parcourant le tableau lignes par lignes pour `imin` la première ligne où on trouve un pixel noir, ensuite pour chaque nouveau pixel noir on compte sa ligne pour `imax` jusqu'à l'épuisement de tous les pixels.

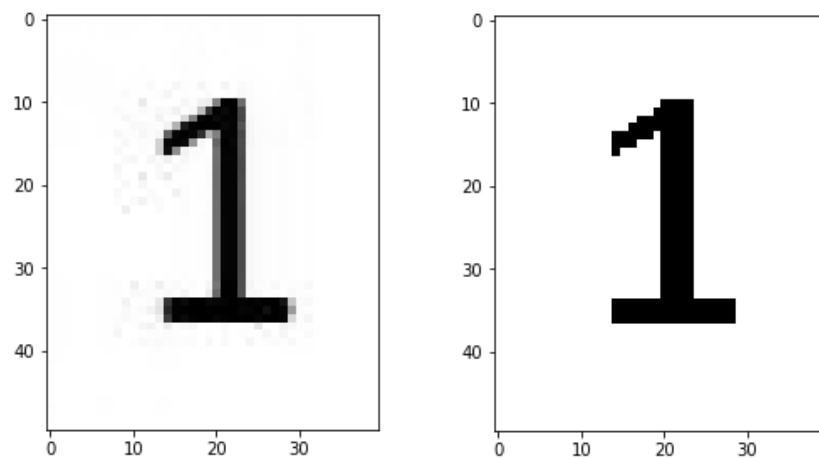
Ensuite on effectue la même chose pour `cmin` et `cmax` en parcourant le tableau par colonne cette fois.

### III. Reconnaissance automatique de chiffre

#### 1. Question (1).

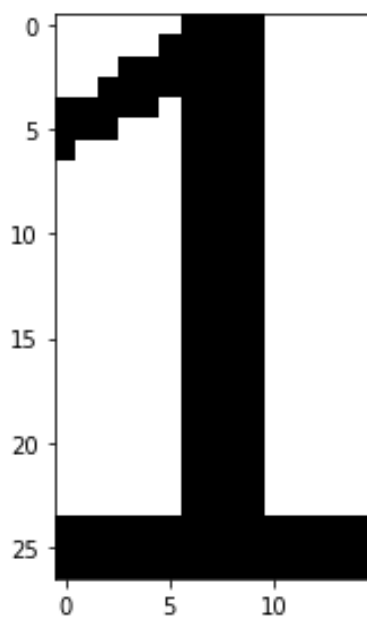


Test pour `S = 70`

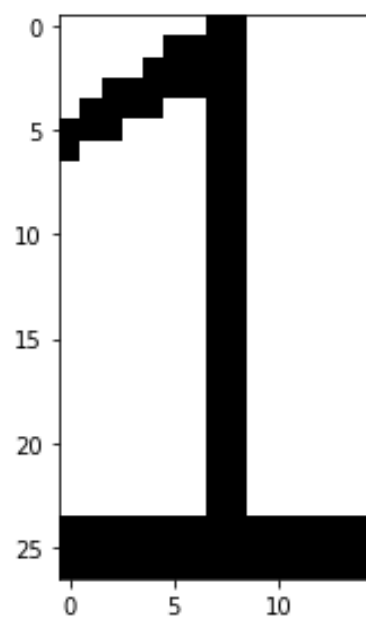


Test pour  $S = 137$

## 2. Question (2).



Localisation pour  $S = 137$



Localisation pour  $S = 70$

### 3. Question (3).

On crée une nouvelle image puis on utilise la fonction `resize` sur l'attribut `pixels` de l'image qu'on implante directement avec la méthode `set_pixels`. On renvoie finalement la nouvelle image obtenue

### 4. Question (4).

Pour calculer la similitude entre deux images, qu'on suppose de même taille dans la fonction, on utilise un compteur de chaque pixel en commun, et à la fin on renvoie ce nombre divisé par le nombre total de pixels, ce qui renvoie bien la proportion de pixels égaux.

### 5. Question (5).

On a une liste contenant les images des 10 chiffres. On calcule pour chaque image la similitude avec notre image en entrée préalablement binarisé et localisé puis on adapte les dimensions de l'image à l'image du chiffre que l'on va comparer.

Ensuite, pour chaque chiffre on étudie la similarité et on renvoie finalement l'indice de l'image dans la liste des chiffres (ce qui équivaut à donner le chiffre car à l'indice 0 de la liste on trouve l'image du chiffre 0).

### 6. Question (6).

	S = 120	S = 70	S = 241
Test1	Le chiffre reconnu est : 4	Le chiffre reconnu est : 4	Le chiffre reconnu est : 4
Test2	Le chiffre reconnu est : 1	Le chiffre reconnu est : 1	Le chiffre reconnu est : 1
Test3	Le chiffre reconnu est : 2	Le chiffre reconnu est : 2	Le chiffre reconnu est : 2
Test4	Le chiffre reconnu est : 2	Le chiffre reconnu est : 2	Le chiffre reconnu est : 2
Test5	Le chiffre reconnu est : 2	Le chiffre reconnu est : 2	Le chiffre reconnu est : 2
Test6	Le chiffre reconnu est : 4	Le chiffre reconnu est : 4	Le chiffre reconnu est : 4
Test7	Le chiffre reconnu est : 9	Le chiffre reconnu est : 9	Le chiffre reconnu est : 7
Test8	Le chiffre reconnu est : 2	Le chiffre reconnu est : 2	Le chiffre reconnu est : 2
Test9	Le chiffre reconnu est : 3	Le chiffre reconnu est : 3	Le chiffre reconnu est : 7
Test10	Le chiffre reconnu est : 6	Le chiffre reconnu est : 6	Le chiffre reconnu est : 8

Pour test8 et test9 il manque un système de separation des chiffres les 4 chiffres sur les image sont compressé en un.

Pour test10 avec un seuil de 241 on trouve 8 car l'image est toute noire et 8 est le chiffre avec le plus de pixel noir.

#### **IV. Conclusion**

TP plutôt trivial, seul vrai difficulté : le fait que tous les échantillons de chiffre qu'on utilise pour la reconnaissance n'aient pas une dimension standardisée ce qui est d'un malaise palpable.