BENITO Enzo

DALENC Nathan

Date du TP (30/11/2021)

# Rapport de TP2 – Lecture automatique de chiffres par analyse

# D’image

## Introduction

Les images numériques sont représentées par des tableaux à 2 dimensions, où chaque élément du tableau correspond à un pixel. Dans le cas d'une image en niveaux de gris, la valeur de chaque élément du tableau correspond à l'intensité du pixel.

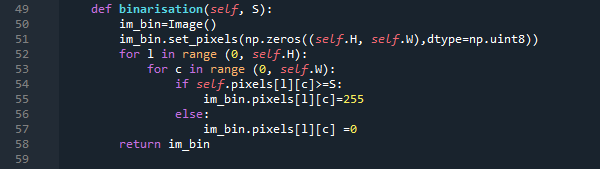
L'objectif de ce TP est d'illustrer, parmi les différentes techniques de lecture automatique de chiffres, une des solutions  
les plus simples : la reconnaissance par corrélation avec des modèles.

## Travail Préparatoire

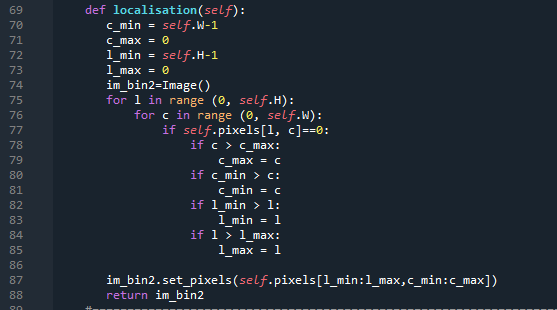
### Question (1).

H correspond à la hauteur du pixel et w correspond à la largeur du pixel.

### Question (2).

Dans la fonction binarisation nous utilisons deux boucle pour parcourir tous les pixels de l’image. Puis nouds comparons si le nombre qui caractérise la couleur est supérieur ou égal. Si c’est le cas le nombre prend la valeur de 255 correspondant à la couleur blanche sinon il prend la valeur 0 correspondant à la couleur noire.

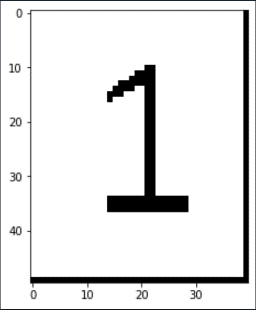
### Question (3).



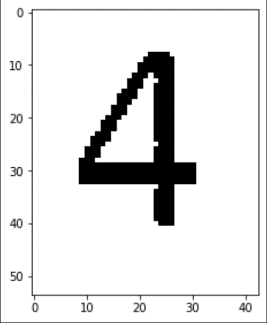
Dans la fonction localisation nous paramétrons le c\_min à la valeur la plus forte et le c\_max a la valeur la plus faible. De sorte à ce que les comparaisons puis le remplacement soit facilement exécutable. Nous faisons la même chose pour l\_min et l\_max. Puis nous formons 2 boucles pour parcourir toute les cases du tableau afin de pouvoir effectuer des comparaisons et si besoin remplacer les valeurs de c\_min, c\_max, l\_min et l\_max. Puis nous retournons l’image obtenue.

## Reconnaissance automatique de chiffre

### Question (1).

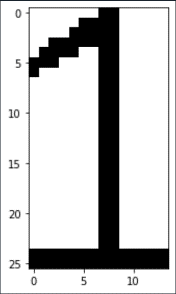
Pour cette fonction nous n’avons pas eu de problèmes.

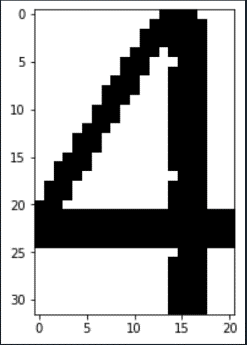
On peut voir qu’il ne reste que des pixels blancs ou noirs.



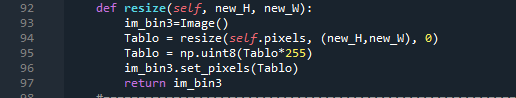
Un autre exemple :

### Question (2).

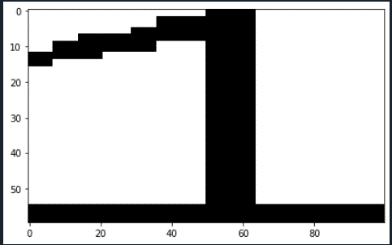
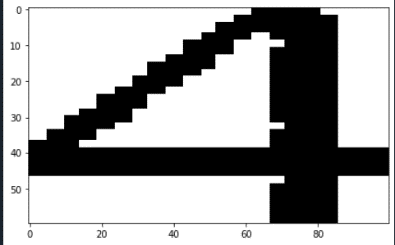
Lors de l’exécution de cette fonction nous avons eu un problème au niveau des comparaisons car nous nous sommes trompé pour les sens des inégalités de c\_min et l\_min.

Un autre exemple :

### Question (3).



Dans cette fonction nous créons une nouvelle image, une variable « tablo » où nous stockons les valeurs des pixels puis nous les convertissons en image avec la fonction set.pixels.



Nous obtenons ceci :

### Question (4)

### 

Pour cette fonction nous initialisons une variable à 0 puis en parcourant le tableau et en effectuant des test d’égalité de couleur et de position si cela est le cas nous incrémentons la variable de 1. A la fin nous effectuons le rapport entre la variable et la totalité des pixels.

### Question (5)

Pour cette fonction nous commençons par binariser l’image que nous voulons comparer puis nous la localisons. Ensuite nous créons une boucle pour parcourir toutes les images de la liste dans cette boucle nous créons une nouvelle image en redimensionnant notre image avec les caractéristiques de l’image de la liste. Puis nous effectuons une comparaison pour obtenir un indice de similitude. Donc si cet indice de similitude est supérieur à l’indice précédent il sera remplacé par celui-ci. Puis nous gardons le rang de l’image avec le plus fort indice de similitude. Et nous retournons ce rang.



## Conclusion

Ce TP nous a permis de mieux maîtriser le langage python et ses subtilités. En autres de nous familiarisé avec l’utilisation des fonctions précédemment créer. De plus de mieux comprendre nos erreurs d’indice dans les boucles du TP précèdent. Enfin nous avons compris le fonctionnement des consoles affichant les erreurs.