JANOT Romain
BABOLAT Anatole
RIVAULT Mathieu
Date du TP : (1/12/2021)

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

I. Introduction

Les images numériques sont représentées par des tableaux a 2 entrées
Chaque case du tableau correspond à un pixel dont la couleur est codée entre 0 et 255 dans le cas d'une image en niveaux de gris.
Le but de ce TP est d'utiliser différentes fonctions afin de pouvoir "lire" un nombre sur une image en niveaux de gris.
Pour arrive à ce but la, nous allons coder 4 fonctions puis utiliser ces fonctions afin de comparer une image "test" avec une banque de modèles permettant de définir le chiffre affiché sur l'image test.

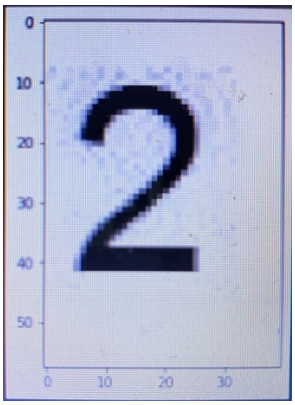
II. Section 1 du TP

1. Question (1).

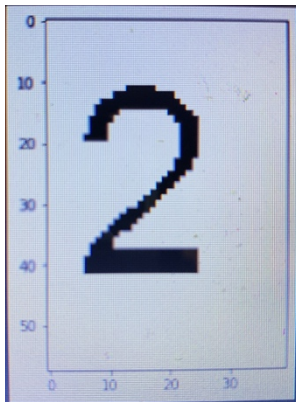
La Fonction Binarisation permet, a partir d'une image contenant des pixels entre 0 et 255, de transformer tous les pixels soit en 0 soit en 255 donc soit en Blanc soit en Noir, l'image finale comporte donc 2 couleurs.
Pour cela, on parcourt chaque pixel avec 2 boucles for, afin de comparer les pixels et vérifiés si leurs valeurs est inférieur au seuil (à l'aide d'un IF).

Voici les résultats obtenus pour le chiffre 2 :

Avant Binarisation :

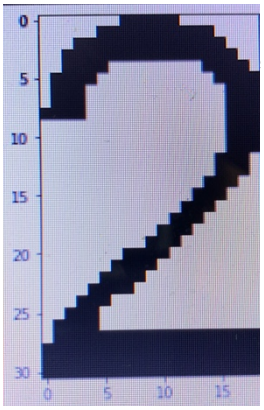


Après binarisation :



2. Question (2).

La fonction localisation permet de réduire la taille de l'image. On supprime les lignes et colonnes où il n'y a que des pixels blanc afin d'optimiser la taille de l'image.

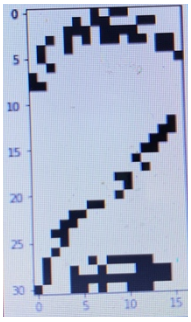


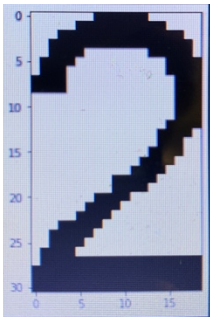
avec l'image du chiffre 2 binarisée.

III. Reconnaissance automatique de chiffre.

Q1) et Q2)

Plus le seuil est petit, moins le chiffre est visible et donc il y a moins de pixel noir. On voit donc de moins en moins le chiffre

Pour S =2 : 

Pour s = 100 : 

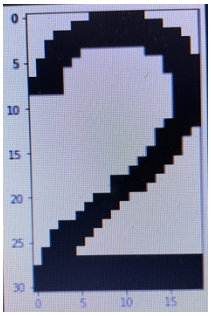
Pour S = 250 : 

La localisation s'adapte automatiquement.

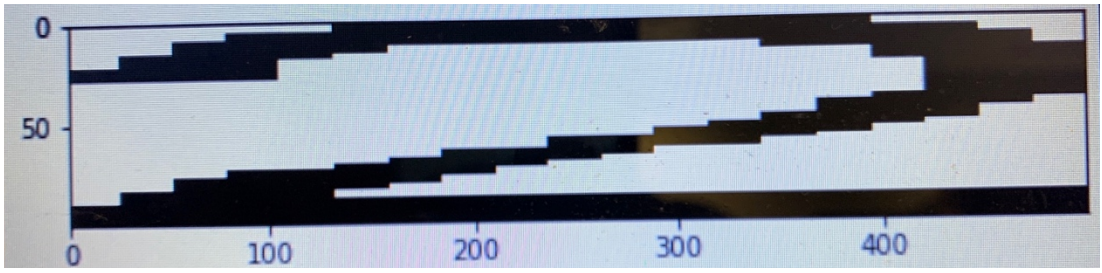
Q3)

La fonction Resize permet de redimensionner une image avec un nombre de pixel donné.

AVANT



APRÈS



L'image est modifiée pour être affiché selon 100 pixels par 500 pixels

Q4)

Dans la fonction similitude, on parcourt chaque pixels, grâce à 2 boucles FOR, ensuite avec un IF on compare les 2 pixels et s'ils sont égaux le compteur augmente.

Enfin, pour avoir un ratio, il suffit de diviser le nombre obtenu de pixels identiques avec le nombre totale de pixels.

Nous avons vérifié que la fonction marche bien.

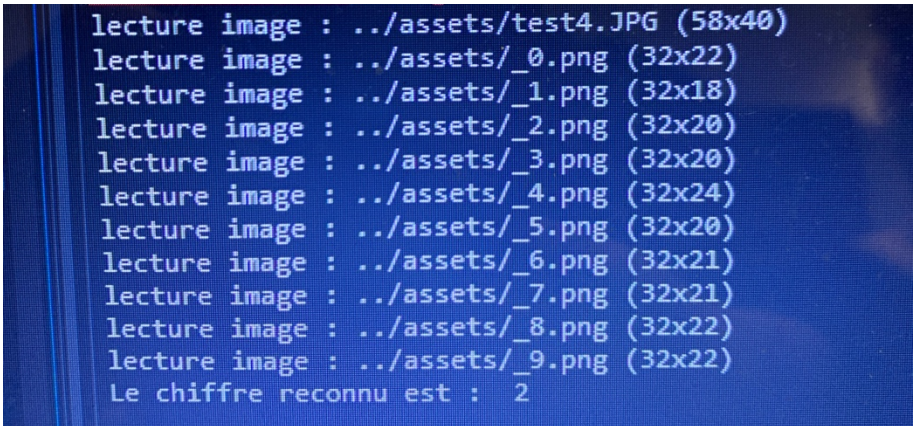
```
In [115]: liste_modeles[1].similitude(liste_modeles[2])
Out[115]: 0.6371527777777778

In [116]: liste_modeles[1].similitude(liste_modeles[1])
Out[116]: 1.0

In [117]:
```

Q5)

Comme indiqué dans les consignes, nous avons commencé par redimensionner l'image passer en paramètre avec les fonctions localisée et binarisée (CF question 1 et 2).

Ensuite nous avons mis un compteur pour l'indice max de la position dans la liste des modèles. Il suffit de parcourir la liste des modèles et d'appliquer la fonction similitude entre l'image redimensionnée et l'image de la liste des modèles de l'itération de la boucle. Si la similitude est plus grande que celle du compteur, l'indice est alors modifié.
Nous obtenons donc l'indice de l'image de la liste et donc le chiffre le plus ressemblant.

Voici le résultat obtenu :



IV. Conclusion

Ce TP nous a permis de mieux manipuler les fonctions, ainsi que la manipulation de plusieurs fichiers.
Nous avons eu du mal avec la syntaxe notamment au début du TP, mais nous avons compris nos erreurs et avons rectifié le tir.
Pour finir, les objectifs du TP nous paraissait difficiles au début mais le fait de décomposer en pleins de fonctions, nous à amener à comprendre et à simplifier les choses.