DUCLAUT Nathan
GARCIN Anthony
(03/12/2021)

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

I. Introduction

Lors de ce TP, nous tenterons de d'associer des images de nombres aux nombres qu'ils sont réellement. Les images seront en réalités des tableaux et nous suivrons le protocole proposé par l'énnoncé.

II. Travail préparatoire

1. Question (1)

Nous observons que l'attribut H correspond à la hauteur de l'image. De plus, cela correspond au nombre de lignes du tableau. Nous utilisons H car hauteur = Height en anglais. De la même manière, W correspond à la largeur de l'image ou le nombre de colonnes du tableau et largeur se dit Width en anglais.

2. Question (2)

La méthode binarisation crée une nouvelle image et permet d'éliminer les pixels gris. Cela permet d'obtenir une image plus nette. Nous passons par un seuil en dessous duquel un pixel gris devient blanc et au-dessus duquel un pixel gris devient noir. Nous utilisons une double boucle pour afin de passer à travers toutes les lignes et les colonnes et une condition if sur le seuil.

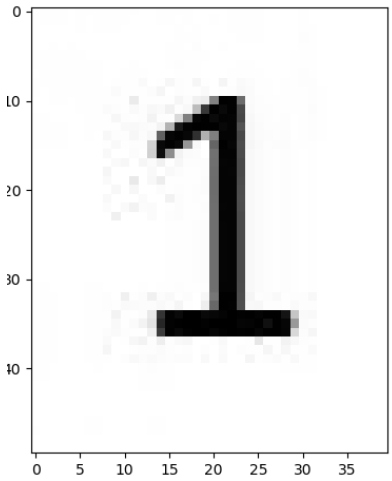

Image non binarisée

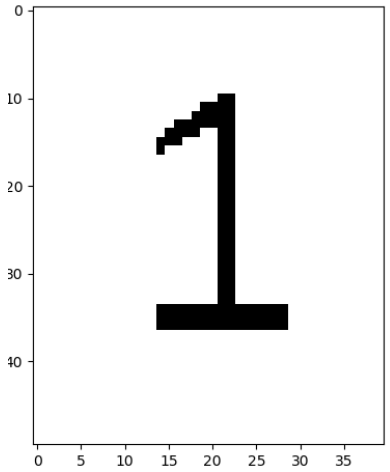

Image binarisée

3. Question (3)

A présent nous cherchons à localiser l'image. Nous souhaitons que le motif occupe tout le cadre de l'image, les pixels blancs autour du motif seront donc supprimés.
On définit d'abord Cmin et Lmin à 0 et Cmax et Lmax respectivement la largeur max et la hauteur max. Ces quatre coordonnées forment un rectangle qui occupe toute l'image. A l'aide d'une double boucle pour, on parcourt toute l'image et à chaque pixel noir rencontré on ajuste la taille du rectangle en comparant la position actuelle aux valeurs existantes de Cmin,Lmin,Cmax et Lmax.

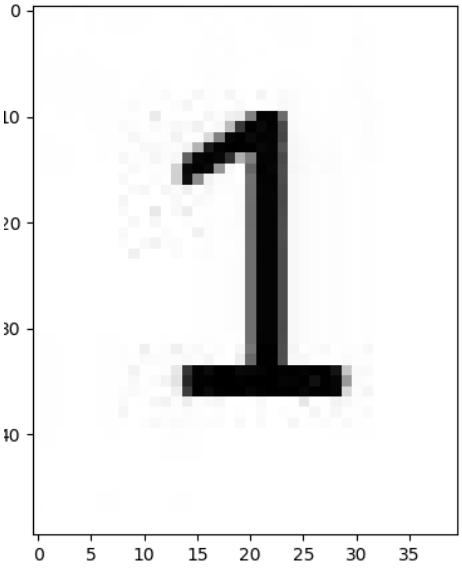

Image non recadrée

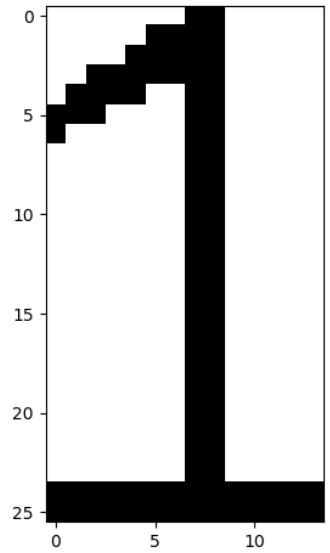

Image recadrée

III. Reconnaissance automatique de chiffre

1. Question (1/2)

Dans cette question, nous observons les résultats de nos deux méthodes précédentes pour des valeurs de seuil différentes. Nous ajustons cette valeur dans le programme main.

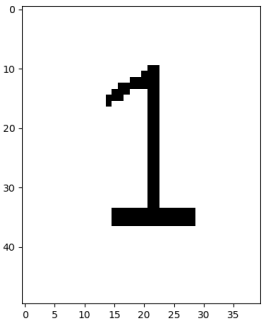

Image binarisée S=30

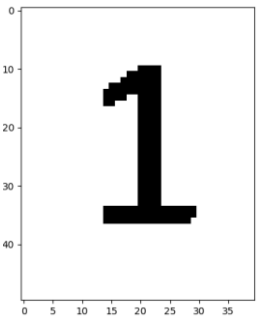

Image binarisée S=200

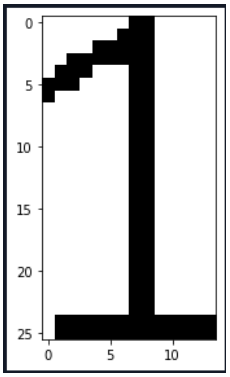

Image localisée S=30

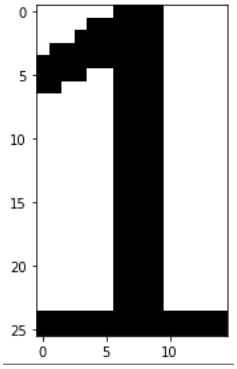

Image localisée S=200

Nous observons que nos méthodes fonctionnent pour plusieurs seuils et que celui-ci a pour effet « d'affiner » le chiffre lorsqu'il est faible et de « l'élargir » lorsqu'il est élevé. Cela était prévisible car plus le seuil diminue plus des gris intenses deviennent blancs et inversement.

2. Question (3)

Ici, nous allons réajuster la taille de notre image recadrée. Nous allons utiliser la fonction resize de la librairie skyimage qui permet cette fonctionnalité.

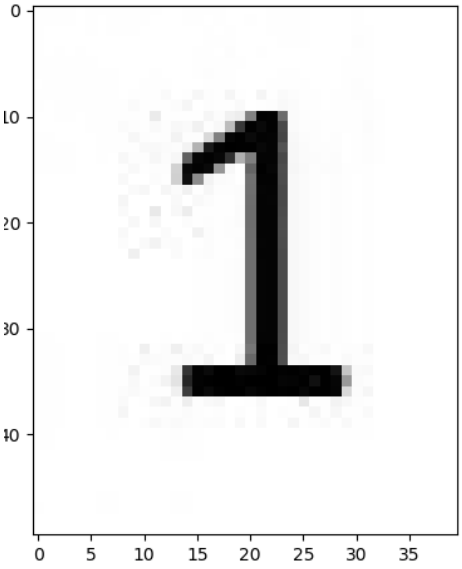

Image non réajustée

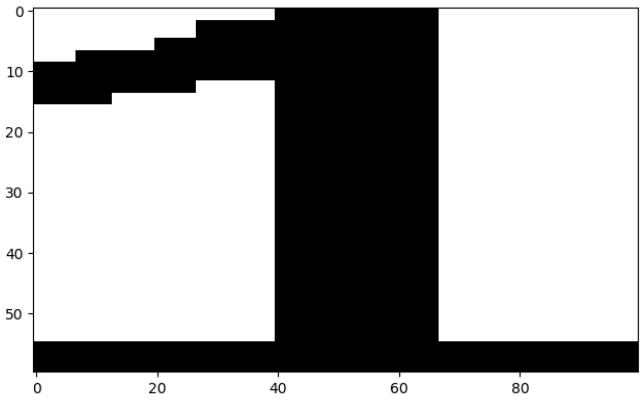

Image réajustée

3. Question (4)

Dans cette question, nous cherchons à déterminer le degré de similitudes entre deux images. On construit la fonction similitude qui donne la valeur de similitude (comprise entre 0 et 1) entre self et une image entrée en paramètre. Pour deux images de même taille, nous comparons chaque pixel deux à deux et observons l'intensité de ceux-ci. Ainsi, la similitude dépendra des pixels localisés au même endroit et de même intensité. La fonction nous retourne donc le nombre de pixel similaire sur le nombre total de pixel.

4. Question (5)

L'objectif est à présent d'associer le bon chiffre à l'image proposé. Nous devons donc réaliser la fonction reconnaissance_chiffre. Dans un premier temps, on binarise et localise l'image passée en paramètre de la fonction puis on initialise la similitude et le chiffre reconnu. On utilise une boucle pour afin de tester tous les modèles fournis et une condition if pour déterminer la similitude max et donc le chiffre reconnu.

IV. Conclusion

Ce TP nous a permis de comprendre les étapes du la reconnaissance d'images avec Python. Il faut considérer les images comme des tableaux de pixels et suivre le protocole suivant : binariser, localiser, réajuster la taille, estimer à similitude et enfin choisir, attribuer une valeur ou ici un chiffre à notre image.