

Projet d'initiation à la recherche en image

Romain Chagnaud
Alexandre Malagnac

Introduction	1
Présentation du projet	1
Présentation de la base	1
Enjeux scientifiques	1
Proposition d'une solution	2
Chaîne de traitement/analyse	2
Algorithme d'extraction de caractéristiques utilisé	2
Analyse des résultats	3
Présentation d'une matrice de confusion	3
Analyse de cas qui n'ont pas marché	4
Conclusion	5
Rappel des résultats obtenus	5
Perspective d'amélioration des résultats	5

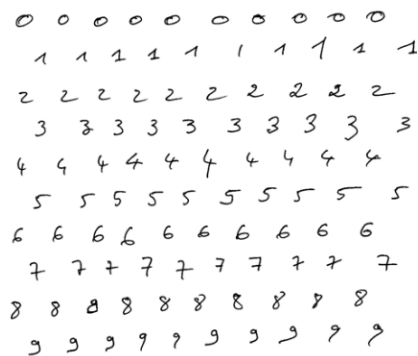
1. Introduction

- Présentation du projet

Le but du projet consiste en la réalisation d'un moteur OCR permettant la classification de chiffres manuscrits par plus proches voisins.

Le système OCR (Optical Character Recognition), ou bien ROC (Reconnaissance Optique de Caractères), est un système qui permet de scanner un texte ou un document qui pourra ensuite être édité sur un smartphone ou sur un ordinateur, souvent avec l'aide d'une intelligence artificielle.

- Présentation de la base



La base comporte 100 images représentant des chiffres manuscrits. Ces données nous permettront d'effectuer de nombreux tests pour une meilleure analyse globale.

- Enjeux scientifiques

Le système OCR est une technologie polyvalente qui est utilisée dans de nombreuses situations : documentation officielle, ou encore jeux récréatifs.

Par exemple, aujourd'hui les centres de tri de courrier utilisent souvent le système OCR pour gérer et trier le courrier. Cela permet de classer plus vite et d'améliorer les délais de livraison.

Ce système est aussi populaire sur les applications pour smartphone. Il est utile pour différentes fonctions : scanner un ticket de caisse, ou encore scanner et reconnaître un document écrit à la main.

2. Proposition d'une solution

- Chaîne de traitement/analyse

Le redimensionnement de l'image :

Les images de la base ont une faible résolution et ne sont pas toutes de la même taille. Le redimensionnement permet donc une meilleure visibilité des courbes du caractère manuscrit et d'effectuer l'analyse sur le même nombre de pixels pour chaque image.

Image non redimensionnée :



Image redimensionnée :



La conversion en noir et blanc de l'image :

Les images de la base sont en niveau de gris. Si on détermine un seuil correct de conversion en noir et blanc, alors on pourra déterminer avec plus de précision les contours du caractère manuscrit.

Image après conversion :



- Algorithme d'extraction de caractéristiques utilisé

On utilisera un algorithme kNN pour déterminer chaque caractère manuscrit de la base.

L'algorithme des K plus proches voisins ou K-nearest neighbors (kNN) :

(kNN) est un algorithme de Machine Learning qui appartient à la classe des algorithmes d'apprentissage supervisé simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression

Plus d'informations sur l'algorithme kNN:

<https://datascientest.com/knn>

Pour nous simplifier la tâche, le jeu de données d'entraînement et de test sera le même. On procédera de cette manière afin de mieux optimiser le code mais aussi car la base du projet ne contient pas suffisamment de données (100 images avec seulement 10 par chiffre).

3. Analyse des résultats

- Présentation d'une matrice de confusion

```
n_test : 100
k : 7
Veuillez patienter 100 tours :
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
4 65 66 67 68 69 70 71 72 73 74 75 76 77

Résultats du test OCR :
0| [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
1| [1, 1, 1, 1, 1, 7, 1, 1, 1]
2| [2, 2, 2, 2, 2, 2, 2, 6, 2]
3| [3, 7, 3, 3, 3, 3, 3, 3, 3]
4| [4, 4, 4, 6, 4, 4, 4, 4, 4]
5| [5, 5, 5, 5, 5, 5, 5, 5, 5]
6| [6, 6, 6, 6, 6, 6, 6, 6, 6]
7| [7, 7, 7, 7, 7, 7, 7, 7, 7]
8| [8, 8, 3, 8, 8, 8, 2, 8, 8]
9| [9, 9, 9, 9, 9, 3, 9, 9, 9]
Le taux de reconnaissance est de 93.0%
```

On effectue 100 tests (pour 100 images) et k=7 (on testera à chaque fois pour les 7 voisins les plus proches d'un élément).

Après avoir redimensionné les images et effectué l'algorithme d'extraction de caractéristiques, on peut voir que le résultat est assez satisfaisant : le taux de reconnaissance est de 93,0%.

Sur la matrice de confusion, on peut voir que le programme a deviné correctement tous les 0, 5, 6 et 7 de la base. On peut aussi voir que le chiffre 8 représente le plus grand taux d'erreur car un chiffre 8 manuscrit mal fait peut facilement ressembler à un autre nombre comme le 3 par exemple (en ne fermant pas correctement la boucle du 8).

- Analyse de cas qui n'ont pas marché

Fonctionnement de l'algorithme d'extraction sans chaîne de traitement :

```
n_test : 100
k : 7
Veuillez patienter 100 tours :
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
4 65 66 67 68 69 70 71 72 73 74 75 76 77

Résultats du test OCR :
0| [0, 1, 1, 0, 0, 0, 3, 1, 0]
1| [1, 3, 1, 1, 1, 1, 1, 1, 1]
2| [1, 2, 1, 2, 2, 1, 1, 1, 1]
3| [3, 1, 3, 3, 1, 3, 1, 1, 1]
4| [1, 4, 1, 1, 1, 1, 4, 4, 1]
5| [5, 1, 1, 1, 1, 5, 1, 0, 1]
6| [6, 6, 6, 6, 6, 6, 1, 1, 6]
7| [1, 1, 4, 7, 7, 4, 7, 7, 1]
8| [8, 9, 1, 1, 8, 1, 8, 1, 8]
9| [1, 9, 9, 3, 2, 9, 5, 1, 1]
Le taux de reconnaissance est de 46.0%
```

On peut voir ici que le taux de reconnaissance est de seulement 46%. La raison de ce résultat est que l'algorithme d'extraction n'est pas suffisant à lui seul : les images de la base ne sont pas toutes de la même dimension, ce qui rend le programme beaucoup plus imprécis (on ne calcule pas avec le même nombre de pixel pour chaque image).

Conversion en noir et blanc avec un mauvais seuil :

```
n_test : 100
k : 7
Veuillez patienter 100 tours :
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
4 65 66 67 68 69 70 71 72 73 74 75 76 77

Résultats du test OCR :
0| [0, 0, 0, 0, 0, 0, 0, 0, 0, 9]
1| [1, 1, 5, 1, 1, 5, 1, 1, 1]
2| [2, 2, 2, 2, 2, 2, 2, 2, 2]
3| [3, 7, 3, 3, 3, 3, 3, 3, 3]
4| [4, 4, 4, 7, 4, 4, 4, 4, 4]
5| [5, 5, 5, 5, 5, 5, 5, 5, 5]
6| [6, 6, 6, 6, 6, 6, 6, 6, 6]
7| [7, 4, 7, 7, 7, 7, 7, 7, 7]
8| [8, 8, 3, 3, 8, 5, 2, 5, 9]
9| [9, 9, 9, 9, 9, 3, 9, 5, 9]
Le taux de reconnaissance est de 84.0%
```

Ici, on applique le redimensionnement et la conversion en noir et blanc des images. Le résultat est cependant plus imprécis que si on avait appliqué seulement le redimensionnement ($84,0\% < 93,0\%$). La raison de cet imprécision est que la simple conversion de l'image en noir et blanc avec un seuil de base (pixel noir si $p < 128$ et pixel blanc si $p \geq 128$, p étant la valeur de gris du pixel à convertir) donne une image avec beaucoup de bruits (points blancs sur les lignes noires du caractère) et rend la lecture du caractère plus difficile.

4. Conclusion

- Rappel des résultats obtenus

Avec de simples algorithmes, nous arrivons à un taux de reconnaissance plus que convenable : 93% de taux de reconnaissance seulement grâce à un algorithme de redimensionnement et un algorithme kNN.

- Perspective d'amélioration des résultats

On pourrait améliorer ce résultat en :

- déterminant un seuil de conversion de l'image en noir et blanc optimal;
- utilisant un algorithme d'extraction de caractéristique de Zoning.