

RAPPORT PROJET IMAGE

1) Introduction

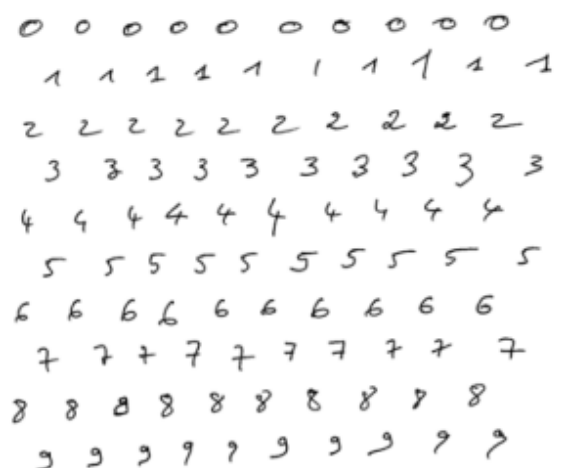
a) Présentation du projet

L'objectif de ce projet est de réaliser un moteur d'OCR de chiffres manuscrits. Le mot OCR signifie Reconnaissance Optique de Caractères ou reconnaissance de texte. C'est une technologie qui permet de convertir différents types de documents tels que des documents papiers scannés, des fichiers PDF ou des photos numériques en fichiers modifiables par exemple. Durant ce projet, nous avons donc créé un programme qui compare une image avec une bibliothèque d'image sur différentes caractéristiques. Après une série de calculs que nous détaillerons plus bas, l'image de la bibliothèque dont les caractéristiques sont les plus proches de notre image est élue. Étant donné que le caractère correspondant à l'image de la base est connu, on peut supposer que si les caractéristiques sont rigoureusement choisies, le caractère inconnu correspond au caractère de l'image sélectionnée.

b) Présentation de la base

Cette base contient 120 images. Chaque image correspond à 1 caractère. Il y a 10 fois chaque caractère (0,1,2,3,4,5,6,7,8,9,-,+).

Chaque chiffre est écrit d'une manière différente. Le nom de chaque fichier est construit selon le masque suivant : "<chiffre>_<numéroImage>". Toutes les images sont en PNG et sont d'une taille différente.





Exemple d'une base
Similaire


c) Enjeux Scientifiques

Les enjeux scientifiques sont nombreux et certains compliquent beaucoup la tâche.

Toutes les images ont une taille différente dans la base. C'est un enjeu parce qu'un pixel (ou groupe de pixel) peut exister sur une image et pas sur une autre.

Ensuite, il y a 12 possibilités de résultat. Chaque image est certes différente mais en même temps la différence est légère parfois. Certains caractères entre eux se ressemblent beaucoup, comme par exemple un «7» et un «1» : 

L'inverse est également possible, deux 8 par exemple peuvent ne pas du tout se ressembler : 

Pour finir, une difficulté que nous avons rencontrée, c'est avec l'image 1_6.png :  Cette image est très basique étant donné qu'il s'agit d'un seul trait. Nos algorithmes ont beaucoup été perturbés par cette image. C'est régulièrement celle qui ressemblait le plus aux autres images selon certains de nos algos.

2) Proposition d'une solution

a) Chaîne de traitement

Afin de comparer les images, nous avons défini une chaîne de traitement qui servira de "repère" pour différencier les images.

Nous avons choisi comme chaîne de traitement les niveaux de gris, l'idée proposée au début. Cela implique que pour chaque pixel d'une image, on prend son niveau de gris et on s'en sert comme donnée. Cela s'avère très utile vu que les images de la base sont en noir et blanc.

On peut par exemple faire la moyenne de tous les niveaux de gris d'une image (cela ne suffit pas bien évidemment) ou encore détecter les contours, grâce à un changement brutal de niveau de gris d'un voisin à un autre.

Un autre choix de traitement était le binaire mais nous avons préféré rester sur la chaîne de traitement utilisée initialement.

b) Détail d'un algorithme d'extraction

Nous nous allons ici détailler un des algorithmes implémenté dans notre code : **le zoning**

Dans notre code, c'est la méthode `divideImage()` qui s'occupe d'appliquer cet algorithme.

```
private static Double divideImage(ImageProcessor ip) {  
    float width = ip.getWidth();  
    float height = ip.getHeight();  
    double sum = 0;  
    int occurrence = 0;  
    for (float x = 0; x < width; x = width / 3 + x) {  
        for (float y = 0; y < height; y = height / 3 + y) {  
            ip.setRoi((int) x, (int) y, (int) ip.getWidth() / 3, (int) ip.getHeight() / 3);  
            double meanGray = meanImage(ip.crop());  
            sum += meanGray;  
            occurrence++;  
        }  
    }  
    return sum / occurrence;  
}
```

Le principe est simple.

Tout d'abord on récupère la taille de l'image, c'est-à-dire hauteur + largeur (height + width en anglais). Cet information est cruciale pour découper l'image en plusieurs parties.

Ensuite on va parcourir à l'aide d'une double boucle *for* chaque partie découper de l'image. Ici on a découpé en 9 : le **x** ou le **y** à chaque fois avance de **width/3** ou **height/3** respectivement. A chaque passage de boucle on définit une nouvelle "région d'intérêt" (Region of Interest ou ROI). Cette dernière servira à sélectionner une partie de l'image sur laquelle on obtiendra la moyenne des niveaux de gris grâce à la méthode `meanImage()`.

Une fois tous les niveaux des gris de chaque ROI obtenus, on fait leur moyenne et on peut retourner le coefficient du vecteur (un *double*).

Ce coefficient sera utile par la suite pour calculer la distance entre les vecteurs (et estimer les différences entre les images).

3) Analyse des résultats

a) Présentation d'une matrice de confusion

La génération d'une matrice de confusion est un bon moyen de synthétiser plusieurs dizaines de tests clairement. En effet, lorsque l'on compare une image à 119 autres, il est facile de savoir à quelle image elle ressemble le plus à l'aide de cette matrice.

Elle prend cette forme là :

	0	1	2	3	4	5	6	7	8	9	+	-
0	5	0	0	1	0	0	3	0	1	0	0	0
1	0	4	1	0	1	1	1	1	0	1	0	0
2	0	0	1	2	2	4	1	0	0	0	0	0
3	2	1	2	2	0	0	3	0	0	0	0	0
4	0	1	2	1	0	3	0	3	0	0	0	0
5	0	2	3	0	2	3	0	0	0	0	0	0
6	3	1	0	2	0	0	2	1	0	1	0	0
7	0	1	1	2	2	1	1	1	0	1	0	0
8	0	0	1	0	0	0	1	0	8	0	0	0
9	0	1	0	0	2	0	1	2	0	4	0	0
+	0	0	0	0	0	0	0	0	0	0	9	1
-	0	0	0	0	0	0	0	0	0	0	0	10

Le taux de reconnaissance est de 58 %

Lorsque l'on regarde la première ligne, on peut voir qu'un 0 est reconnu 5 fois en temps que 0, 0 fois en temps que 2, 3 fois en temps que 6, etc.

	0	1	2	3	4	5	6	7	8	9	+	-
0	5	0	0	1	0	0	3	0	1	0	0	0

Sur notre matrice, on obtient un résultat final de 58% de reconnaissance.

On peut apercevoir que les images avec de “+” et des “-” sont nettement mieux reconnues par notre OCR que d'autres images comme 2 ou 4 qui n'ont pas du tout été reconnues.

4) Conclusion

Pour rappel, notre matrice de confusion indique un résultat final de 58% de reconnaissance.

Ce chiffre peut être considéré comme passable mais compréhensible. En effet 4 algorithmes ont été implémentés pour la reconnaissance d'image : le zoning, le rapport isométrique, et le nombre de contours (verticale et horizontale). Ces derniers ne sont pas suffisants, bien que nécessaire de notre avis.

De plus l'analyse des niveaux de gris n'est peut être pas la meilleure chaîne de traitement possible. Tous ces éléments ont la possibilité d'être modifiés et améliorés, ce qui permettrait d'augmenter significativement les résultats.