

Reconnaissance de plaque d'immatriculation

TP 1 : Pré-traitements - localisation et extraction

La première étape d'un système de reconnaissance de formes est généralement la localisation et l'extraction des formes que l'on cherche à identifier (Figure 1).

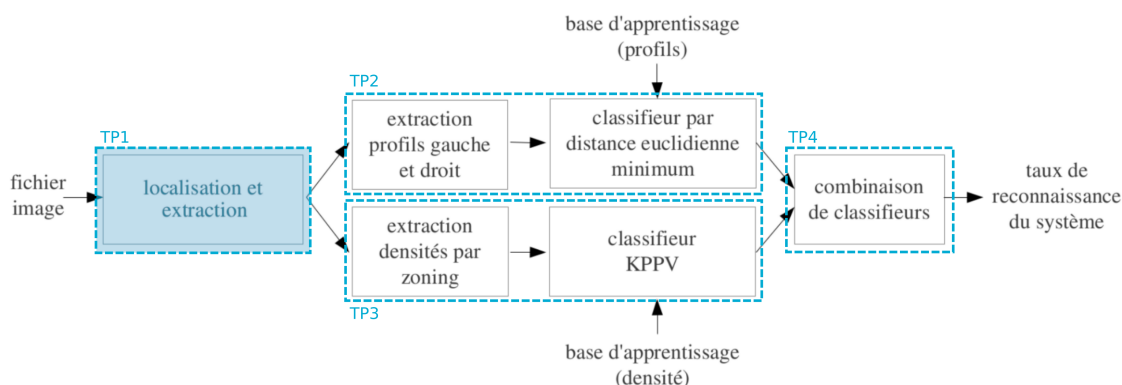


Figure 1: Schéma fonctionnel et planification des séances de TP

L'objectif de ce premier TP est d'élaborer et de développer un programme permettant de localiser et d'isoler chacun des caractères des plaques d'immatriculation provenant des images des bases de test et d'évaluation du dataset "CCPD_SubDataset".

I. Préambule

De nombreuses informations concernant la plaque d'immatriculation visible dans une image du dataset sont directement indiquées dans le nom du fichiers lui-même. Les noms des fichiers peuvent être décomposés en 7 éléments (délimités par le caractère "-") :

1. zone : Rapport entre la surface de la plaque d'immatriculation et la surface totale de l'image,
2. degré d'inclinaison : degré d'inclinaison horizontale et degré d'inclinaison verticale,
3. coordonnées de la boîte englobante : les coordonnées des sommets gauche-haut et droite-bas,
4. emplacement des quatre coins : les coordonnées exactes $\mathbf{u}_i = [u_i, v_i]$ $i = (1, \dots, 4)$ des quatre coins de la plaque dans l'ensemble de l'image,
5. caractères de la plaque : chaque plaque est composé d'un caractère chinois, d'une lettre puis de cinq lettres ou chiffres. Le nom du fichier contient l'indice de chaque caractère tel que : ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'O'].

II. Principes

La détection dans une image de la plaque d'immatriculation elle-même nécessite des notions qui n'ont pas encore été étudiées, c'est pourquoi cette partie n'est pas à traiter dans ce sujet. De plus, nous nous intéresserons ici qu'à l'apprentissage de la reconnaissance des cinq derniers caractères des plaques.

Les différentes étapes de pré-traitements à automatiser au cours de cette séance sont :

1. extraction des coordonnées des 4 coins des plaques d'immatriculation depuis le nom des fichiers (Figure 2a);
2. extraction de la région d'intérêt, redressement et binarisation (Figure 2b);
3. localisation des boîtes englobantes de chaque caractère (Figure 2c);
4. extraction de chaque caractère individuellement (Figure 2d).



Figure 2: Étapes de pré-traitements afin d'extraire les cinq derniers caractères d'une plaque d'immatriculation

III. Travail à réaliser

III.I Extraction d'information depuis le nom des fichiers

Écrire un programme permettant de :

1. Parcourir l'intégralité des fichiers d'un dossier donné (ex. "test" ou "train").

2. Extraire du nom de chaque fichier les coordonnées des 4 coins $\mathbf{u}_i = [u_i, v_i]$ $i = (1, \dots, 4)$ de la plaque d'immatriculation.

Vous pouvez, par exemple, utiliser cette expression régulière :

```
'\d+-\d+_\d+-\d+&\d+_\d+&\d+-(?<x1>\d+)&(?(y1>\d+)_
(?(x2>\d+)&(?(y2>\d+)_?(x3>\d+)&(?(y3>\d+)_?(x4>\d+)&(?(y4>\d+)-.*'
```

3. Extraire du nom de chaque fichier les 5 derniers caractères de la plaque d'immatriculation.

Vous pouvez, par exemple, utiliser cette expression régulière :

```
'\d+-\d+_\d+-\d+&\d+_\d+&\d+-\d+&\d+_\d+&\d+_\d+&\d+_\d+&\d+-\d+_\d+_
(?(d1>\d+)_?(d2>\d+)_?(d3>\d+)_?(d4>\d+)_?(d5>\d+)-'
```

III.II Extraction de la région d'intérêt

Sur l'ensemble du dataset, la pose (position et orientation) de la caméra par rapport au véhicule est variable. De plus, étant donné la perspective, les plaques d'immatriculation ne sont pas projetées dans l'image sous la forme de rectangles mais sous la forme de quadrilatères convexes (Figure 2a). Pour rendre l'extraction des caractères de la plaque et l'apprentissage de leur reconnaissance plus simples, il serait préférable d'unifier les données en simulant une caméra fronto-parallèle, et à une distance fixe des plaques d'immatriculation.

Admettons qu'avec cette caméra "virtuelle", les 4 coins d'une plaque soient toujours projetés aux coordonnées : $\hat{\mathbf{u}}_1 = [0, 0]$, $\hat{\mathbf{u}}_2 = [110, 35]$, $\hat{\mathbf{u}}_3 = [110, 0]$ et $\hat{\mathbf{u}}_4 = [0, 35]$. Puisque les 4 coins d'une plaque appartiennent à un même plan 3D, les 4 coins "réelles" \mathbf{u}_i et les 4 coins "virtuelles" $\hat{\mathbf{u}}_i$ sont reliés par une projection homographique $\mathbf{H}_{3 \times 3}$, tel que : $\hat{\mathbf{u}}_i \propto \mathbf{H}\mathbf{u}_i$ (en coordonnées homogènes). À partir d'une estimation de cette homographie il est alors possible de projeter (*warper*) les points de l'image réelle vers une image simulée supposément face à la plaque d'immatriculation (Figure 2b).

Écrire un programme permettant :

1. d'estimer l'homographie reliant $\hat{\mathbf{u}}_i$ et \mathbf{u}_i (sous Matlab/Octave voir la fonction "computeHomography" donnée, sous Python/C++ voir la fonction "findHomography"),
2. de warper l'image d'entrée pour générer la région d'intérêt, autrement dit, l'image de la plaque redressée de taille 110×35 (sous Matlab/Octave voir la fonction "warpImage" donnée, sous Python/C++ voir la fonction "warpPerspective").

III.III Localisation des boîtes englobantes de chaque caractère

Écrire un programme permettant de :

1. calculer l'histogramme de projections verticales de la région d'intérêt, et y identifier les passages $0 \rightarrow x$ et $x \rightarrow 0$. Ces passages correspondent à la position verticale de début et de fin de chaque caractère (Figure 3).
2. calculer l'histogramme de projections horizontales pour chaque caractère, et y identifier les passages $0 \rightarrow x$ et $x \rightarrow 0$. Ces passages correspondent à la position horizontale de début et de fin de chaque caractère (Figure 4).
3. sauvegarder les coordonnées des boîtes englobantes de chaque caractère (ex. les coordonnées du coin haut gauche et du coin bas droit),

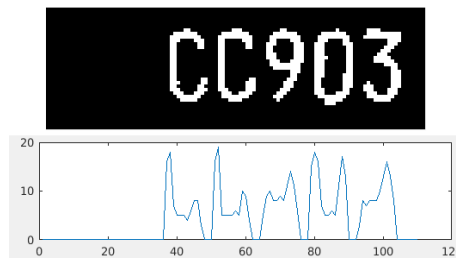


Figure 3: Projection verticale pour détecter le début et la fin de chaque caractère

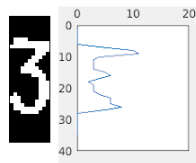


Figure 4: Projection horizontale pour détecter le bas et le haut d'un caractère

Finalement, appliquer le programme ainsi développé à l'ensemble de la base de données ("test" et "train") afin de créer un nouveau dataset plus léger et pré-traité.