PROJET IG.2405

Reconnaissance automatique de la signalisation des lignes de Métro parisien

1. Sujet

La société IMAGIK souhaite développer un produit d'assistance à la mobilité dans le métro parisien pour les personnes malvoyantes. Pour cela, elle propose une paire de lunettes spéciales équipée d'une caméra. Cette caméra envoie les données images à un centre de traitement qui se charge de réaliser la détection de la signalisation et d'indiquer la direction à suivre à l'utilisateur, celui-ci ayant préalablement indiqué sa destination finale.

La société IMAGIK charge des élèves de l'ISEP du module de reconnaissance des lignes de métro dans des images extraites du flux vidéo. Les lignes qui devront être reconnues sont indiquées dans la figure ci-dessous :

FIG. 1. Lignes du métro parisien

Ainsi, le programme de démonstration demandé devra fournir pour chaque image la localisation des signes de lignes de métro détectés et le numéro de la ligne pour chaque signe.

(a) Image source

(b) Détection et localisation

(c) Reconnaissance : lignes 1, 14, 7, 11

FIG. 2. Exemple de résultat d'analyse

2. Bases de données et programmes fournis

Ce sujet est une application typique de vision par ordinateur, comprenant les étapes de prétraitements, segmentation, reconnaissance. Les étapes de segmentation et de reconnaissance nécessiteront de définir un certain nombre de paramètres, comme par exemple les couleurs recherchées, les dimensions des objets, des seuils de similarité, etc. Il faut impérativement définir ces paramètres sur une **base d'apprentissage**. La capacité de généralisation des algorithmes sur des images non apprises sera évaluée sur une **base de test**. C'est une méthodologie générale qu'il faut impérativement respecter dans tout problème de reconnaissance supervisé (i.e. mis au point à partir de données connues).

L'ensemble des images est dans le répertoire **BD**. Il y en a à ce jour 261…

On a retenu 1/3 des images pour l'apprentissage (images 3, 6, 9, ….) et 2/3 des images pour le test (images 1, 2, 4, 4, 7, 8,…). Le fichier **Apprentissage.xls** résume les signes de métro qui sont dans les images d'apprentissage : pour chaque signe, numéro de l'image source, coordonnées de la boite englobante, numéro de la ligne de métro reconnue. Le fichier **Test.xls** a la même structure pour les images de la base de test. Les fichiers **Apprentissage.mat** et **Test.mat** stockent les mêmes informations suivant la même structure. On peut les appeler sous Matlab par :

```
load 'Apprentissage.mat'
load 'Test.mat'
```

Dans les deux cas, on récupère une matrice **BD** de *nb_symboles* lignes x *6* colonnes correspondant exactement aux chiffres des fichiers Excel. Il s'agit de la **vérité terrain** qui sera utilisée pour l'apprentissage et pour l'évaluation des performances.

Vous pouvez afficher sous Matlab les images et la reconnaissance pour les deux bases de données, avec le programme **viewRecognition.m**. Pour les données de la vérité terrain :

```
viewRecognition('Apprentissage.mat','Learn',1)
viewRecognition(Test.mat',Test',1)
```

Vous pourrez faire de même avec les résultats de reconnaissance que vous aurez générés avec votre programme, par les commandes :

```
viewRecognition('myLearnResults.mat','Learn',f)
viewRecognition('myTestResults.mat',Test',f)
```

où 'myLearnResults.mat' et 'myTestResults.mat' sont les fichiers .mat qui stockent vos résultats pour les bases d'apprentissage et de test respectivement, et *f* est l'inverse du facteur de redimensionnement que vous avez appliqué aux images sources pour l'analyse (par exemple, si les images ont été réduites d'un facteur 0.5, *f* devra prendre la valeur 2 pour redimensionner les boites englobantes à afficher dans les images source).

Enfin, le programme **evaluation.m** vous permettra d'estimer les performances de votre algorithme par le calcul de la matrice de confusion, et de taux de reconnaissance :

```
evaluation ('Apprentissage.mat', 'myLearnResults.mat',f)
evaluation ('Test.mat', 'myTestResults.mat',f)
```

3. Travail demandé

Concevoir un programme « metro.m » de reconnaissance des signes de lignes de métro :

Ce programme devra sauvegarder un fichier .mat des résultats de reconnaissance, contenant, pour chaque signe reconnu, le numéro de l'image source, la boite englobante du signe et la classification obtenue, ces données étant structurées suivant le format indiqué dans la section précédente (format des fichiers identique à celui de 'Test.mat' et 'Apprentissage.mat').

Le fichier **metro.m** fourni spécifie le format **imposé**.

Respecter la méthodologie :

Ajuster les paramètres de votre programme sur la base d'apprentissage, sans utiliser les images de la base de test. Evaluer les performances séparément sur les deux bases pour bien évaluer la capacité de généralisation.

Fournir un rapport :

Ce rapport doit être un document bien écrit et rigoureux, rédigé à la manière d'un article scientifique. Il doit contenir les items suivants :

- Position du problème
- Etat de l'art (succinct). Citer les référence par des numéros [1],[2,], …
- Description des méthodes : attention, il ne s'agit pas de recopier du code, mais de formaliser par des équations les traitements appliqués. **Aucun code ou pseudo-code ne doit être présent dans le rapport.** Il faut commencer par présenter les principales étapes du programme (prétraitements, segmentation, …) par un schéma fonctionnel. Pour chaque étape, introduire les méthodes et les variables avec des phrases et formaliser mathématiquement les méthodes par des équations. Bien mettre en évidence les paramètres du programme.
- Résultats expérimentaux et discussion.
- Conclusion et perspectives.
- Références bibliographiques numérotées (articles, liens internet …) qui sont citées dans le corps du rapport.

NB : il n'est pas interdit de reprendre des codes existants mais les sources doivent être citées avec exactitude, même s'il ne s'agit que d'une partie du programme final. **Sinon, le travail s'apparentera à un plagiat et cela sera sanctionné par un 0**. Les codes repris doivent être maîtrisés, expliqués et discutés, voire améliorés. La note attribuée prendra en compte l'apport personnel.

Soutenir :

Préparer une présentation PowerPoint et une démonstration. La présentation PowerPoint devra être réalisée dans le même esprit que le rapport.

Rendre :

- Codes Matlab : respecter les noms des répertoires et des fichiers image, ainsi que la répartition base de données d'apprentissage et de test. Votre code doit tourner sans intervention quelconque du correcteur ! Le programme principal doit correspondre à la syntaxe suivante :

Fichier : **metro.m** (voir fichier fourni)

```
Function [matFileOut,resizeFactor] = metro(type) % type = 'Learn' ou 'Test'
```

Ainsi les lignes de codes suivantes doivent impérativement fonctionner sans plantage :

```
[myLearnResults,resizeFactor] = metro('Learn');
evaluation ('Apprentissage.mat', myLearnResults.mat,1/ resizeFactor);
[myTestResults,resizeFactor] = metro('Test');
```

```
evaluation (Test.mat', myTestResults.mat,1/ resizeFactor);
```

- Bases de données .mat des résultats de reconnaissance, en apprentissage et en test.
- Rapport au format PDF.

Le tout doit être envoyé par mail à florence.rossant@isep.fr dans un fichier .zip. Intégrer toutes les données à l'exclusion des répertoires **BD** et **PICTO.**

Compétences évaluées :

- Capacité à concevoir un système de vision artificielle répondant à un problème posé. Pertinence dans le choix des méthodes. Apport personnel.
- Capacité à expliquer et à formaliser les méthodes proposées (rapport, présentation).
- Qualité de la rédaction et de la présentation orale : structure, rigueur, orthographe, clarté, etc…
- Respect de la méthodologie et du cahier des charges : apprentissage, test, spécifications des formats de données et des programmes.
- Qualité de la programmation : programmation modulaire, structurée, paramètres bien définis (et non codés en dur), code bien commenté, respect des alinéas, etc.
- Présentation des résultats expérimentaux, qualité des résultats, esprit critique.

Le piège à éviter : un programme constitué de méthodes ad-hoc, avec des paramètres codés en dur gérant un nombre important de cas particuliers.