

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 développe une application sur smartphone couplée à une paire de lunettes spéciales équipée d'une caméra. L'utilisateur indique dans l'application la station de métro destination. L'application récupère les images acquises par la caméra, les traite pour extraire les informations relatives à la signalisation du métro, et indique à la personne la direction à prendre.

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

Etant donné les capacités de mémoire et de calcul limitées sur un smartphone, il est exigé de concevoir un système qui fonctionne en deux étapes :

- Une sélection des zones d'intérêt dans lesquelles se trouvent des pictogrammes candidats
- Une reconnaissance des pictogrammes candidats sur ces zones d'intérêt.

2. Bases de données

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. ou d'entraîner un réseau de neurones. Il faut impérativement mettre au point le système sur une **base d'apprentissage**. La capacité de généralisation des algorithmes sur des images non

appprises 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 boîte englobante, numéro de la ligne de métro reconnue. Le fichier **Test.xls** a la même structure pour les images de la base de test.

Il peut être nécessaire de définir une base de validation pour optimiser des hyperparamètres. Dans ce cas vous prendrez sur la base d'apprentissage et vous ferez vous-même la séparation des images répertoriées dans **Apprentissage.xls**.

3. Travail demandé, livrables

Vous développerez deux programmes :

- Une fonction « **metro** » qui prend en entrée le nom d'une image, et qui retourne la liste des signes reconnus (boîte englobante et classe). Cette fonction devra afficher l'image avec les cadres englobants et les classes reconnus.
- Un programme **metro_test.py** qui lance la fonction metro sur toutes les images d'un répertoire donné et qui stocke les résultats dans un fichier excel, au même format que les fichiers fournis.

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 références 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.

Soutenance :

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

Livrables :

- Codes python : 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 !

4. Compétences évaluées

- Capacité à concevoir un système de vision artificielle répondant à un problème posé. Pertinence dans le choix des méthodes. Apport personnel.
- Vue systémique (schéma fonctionnel).
- Vue systémique (schéma fonctionnel).
- 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...
- Etude expérimentale : évaluation quantitative, analyse critique.
- 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.
- Qualité des résultats, esprit critique.