

Test Recrutement Stage

Introduction

Précisions pour répondre convenablement à ce test :

- Aucune *deadline* n'est imposée pour ce test. Prenez le temps que vous jugerez nécessaire pour le réaliser. Les correcteurs privilégieront la qualité des réponses à la rapidité d'exécution. Attention toutefois, pensez que vous répondez à ce test en même temps que d'autres candidats. Pensez à fournir vos résultats dans un délai raisonnable.
- Chaque exercice de ce test peut être réalisé de manière indépendante.
- Si vous avez des doutes ou incompréhensions durant la réalisation de ce test, n'hésitez pas à transmettre vos questions à l'adresse contact@virobotic.com.
- **Concernant la production de codes, scripts, programmes, etc... pour ce test :**
 - Toutes les productions de code (sauf si exception) devront être réalisées en langage python.
 - Pour toutes productions portant sur le traitement d'images, vous devrez **obligatoirement** utiliser la bibliothèque opencv : <https://opencv.org/>
 - Pour toutes productions traitant des matrices ou tableaux multidimensionnels, vous devrez **obligatoirement** utiliser la bibliothèque numpy : <https://numpy.org/>
 - Pensez à commenter vos codes et à apporter par écrit les précisions que vous jugerez nécessaires concernant vos réalisations.
 - Chaque programme doit être le plus généraliste possible. Il doit traiter toutes les images d'un même dossier source. Ainsi, aucun programme ne doit contenir de données brutes permettant de traiter spécifiquement une image et pas les autres.
 - Faire un Readme.txt qui détaille quels sont les prérequis pour lancer vos scripts (version de python, version des bibliothèques installées, etc...), les arguments à renseigner à l'appel de chaque programme, et les données de sorties attendues en résultat de vos programmes (noms et format de fichiers, nom et arborescence de dossiers, etc...).
 - Rédiger un paragraphe explicatif sur la ou les méthodes que vous aurez utilisées sous format pdf (N'hésitez pas à illustrer avec des images de résultats intermédiaires)
- Merci de transmettre vos réponses, vos productions, vos codes commentés à contact@virobotic.com dans un seul dossier archive .zip.
- Dans le cas où vous resteriez bloqué durant un exercice. Vous avez la possibilité de demander une aide/une indication à Virobotic afin d'être en mesure d'avancer. Bien entendu, cette aide sera prise en compte lors de la correction.

Exercice 1 : Clustering

L'objectif de cet exercice est d'entraîner et de tester un modèle de clustering à partir de données d'entraînement et de données tests.

- Pour entraîner votre modèle vous exploiterez le fichier <data_training_ex1.json> fourni avec ce sujet.
- Pour tester votre modèle vous exploiterez le fichier <data_test_ex1.json> fourni avec ce sujet.

La structure des données d'entraînement est la suivante :

- 100 objets différents
- 30 images par objet
- 40 dimensions par image
- Chaque Objet est identifié par un entier : de 0 à 99
- Chaque Image est identifiée par un entier : de 0 à 29
- Les 40 dimensions sont les mêmes pour toutes les images pour tous les objets.

La structure des données de test est la même que pour les données d'entraînement. Cependant, seuls 5 objets sont représentés, et une seule image par objet est représentée. Les identifiants de chaque objet sont indiqués afin de vous permettre de qualifier la qualité de votre modèle entraîné.

Pour cet exercice, vous devez :

- Développer un script permettant de :
 - Extraire les données du fichier d'entraînement.
 - Entraîner un modèle de clustering à partir des données d'entraînement. L'objectif étant que chaque objet soit défini par un cluster.
 - Extraire les données du fichier de test.
 - Tester et qualifier la qualité de votre modèle de clustering à partir des données de test et des indicateurs que vous choisirez.
- Vous avez le choix du ou des modèles de clustering à exploiter pour cet exercice.
- Vous avez le choix du ou des méthodes de qualification de votre modèle (choix des indicateurs, choix du mode de *reporting* sur tableur et/ou via représentation graphique, etc...)
- Toutes les données se trouvent dans le fichier <data_exercice1.zip> fourni avec ce Test.

Merci de nous transmettre dans un dossier .zip :

- Le(s) fichier(s) sources de votre script python commenté(s).
- Le Readme.txt tel que décrit en Introduction.
- Le paragraphe explicatif tel que décrit en Introduction (en y intégrant vos résultats de qualification de la qualité de votre modèle de clustering).

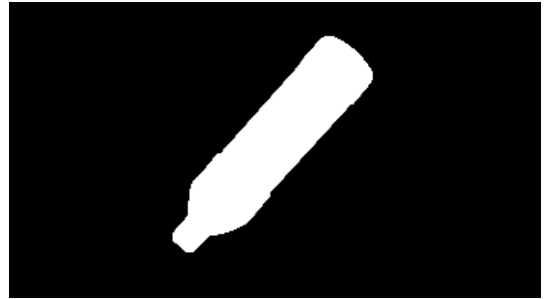
Exercice 2 : Segmentation d'objets sur fond vert

L'objectif de cet exercice est de développer un programme permettant de générer des masques binaires à partir d'images RGB d'articles sur fond verts :

- Le programme doit générer automatiquement (sans intervention de l'utilisateur) des masques binaires à partir d'images RGB d'objets sur fond verts. 0 pour les pixels du fond vert ; 255 pour les pixels de l'objet (voir exemple ci-dessous).



*Entrée :
Image RGB*



*Résultat :
Masque binaire (0 ; 255)*

- Les masques devront être de mêmes dimensions que les images RGB.
- Les masques générés en sorties de programme doivent être impérativement enregistrés sous format jpg.
- Le programme doit traiter toutes les images présentes dans un même dossier. Ce qui signifie que si n images sont présentes dans le dossier, le programme devra traiter les n images du dossier, et générer n masques et donc n fichiers jpg.

Pour tester votre programme, vous exploiterez les images fournies dans le fichier <images_exercice2.zip> fourni avec ce Test.

Merci de nous transmettre dans un dossier .zip :

- Le(s) fichier(s) sources de votre programme python commenté(s).
- Les images résultats de vos tests de votre programme au format jpg.
- Le Readme.txt tel que décrit en Introduction.
- Le paragraphe explicatif tel que décrit en Introduction.

Exercice 3 : Génération d'images artificielles de pinces

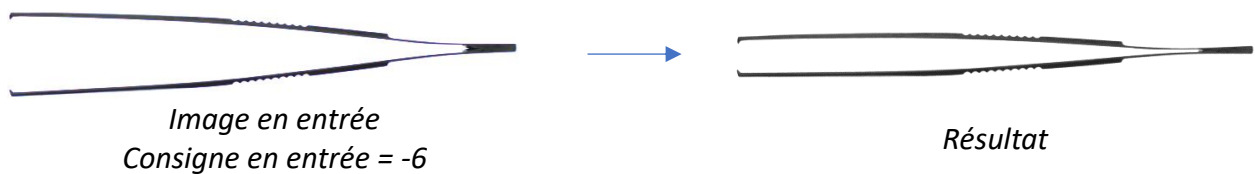
L'objectif de cet exercice est de générer, à partir d'images réelles, des images artificielles de pinces dont les branches sont plus ou moins écartés l'une de l'autre :

- Le programme doit générer automatiquement (sans intervention de l'utilisateur) des images RGB artificielles de pinces à partir d'images réelles RGB. Pour chaque image générée les branches de la pince doivent être plus ou moins écartés l'une de l'autre. Cet écart est défini par la consigne en entrée du programme qui indique la différence de la valeur de l'angle (exprimée en degrés) entre les deux branches entre l'image source et l'image désirée.

Exemple 1 : L'écart entre les deux branches de la pince est de 10° sur l'image source. La consigne est de 6. L'écart entre les deux branches est de 16° sur l'image résultat.



Exemple 2 : L'écart entre les deux branches de la pince est de 10° sur l'image source. La consigne est de -6. L'écart entre les deux branches est de 4° sur l'image résultat.



- Le programme doit gérer convenablement le cas où la consigne d'angle donnerait un résultat non réaliste (c'est-à-dire dans le cas où les deux branches se superposeraient dans l'image résultat).
- Le programme doit gérer convenablement le cas où la consigne d'angle est de zéro. (Le programme ne doit dans ce cas pas générer de nouvelles images.)
- Les images résultats doivent conserver les proportions des images d'entrée.
- Le programme doit gérer convenablement le cas où l'image résultat est plus grande que l'image d'entrée. C'est-à-dire que la hauteur de l'image doit être ajustée.
- Les images générées en sorties de programme doivent être impérativement enregistrées sous format jpg.
- Le programme doit traiter toutes les images présentes dans un même dossier. Ce qui signifie que si n images sont présentes dans le dossier, le programme devra traiter les n images du dossier, et générer n nouvelles images au format jpg.

Pour tester votre programme, vous exploiterez les images fournies dans le fichier <images_exercice3.zip> fourni avec ce Test.

Merci de nous transmettre dans un dossier .zip :

- Le(s) fichier(s) sources de votre programme python commenté(s).
- Les images résultats de vos tests de votre programme au format jpg.
- Le Readme.txt tel que décrit en Introduction.
- Le paragraphe explicatif tel que décrit en Introduction.