

SY32 – Analyse et synthèse d’images

Projet : Détection de visages

1 Données

La partie « Projet » du Moodle contient une archive `projet_train.zip`. Cette dernière est composée d’un dossier `train` contenant les images d’apprentissage et un fichier `label.txt` contenant les annotations des visages.

```
project_train.zip
/project_train/
  train/
    0001.jpg
    0002.jpg
    ...
    1000.jpg
  label.txt
```

Le fichier `label.txt` indique pour l’ensemble des images d’apprentissage la position des visages avec sur chaque ligne du fichier : l’indice de l’image associée puis suivie de quatre nombres i, j, h, l représentant la boîte englobante :

- (i, j) : coordonnées (ligne, colonne) du coin supérieur gauche de la boîte
- (h, l) : taille (hauteur, largeur) de la boîte

L’indice d’une image peut apparaître plusieurs fois s’il y a plusieurs visages sur l’image.

```
/project_train/label.txt
1 40 179 241 179
2 19 60 135 89
2 87 156 83 55
...
1000 10 121 153 95
```

L’archive `projet_test.zip` contient 500 images de test.

```
project_test.zip
/project_test/
  test/
    0001.jpg
    0002.jpg
    ...
    0500.jpg
```

2 Travail à rendre

Rendu attendu pour le vendredi 19 avril 2019 à 23h59 (UTC+02:00) à télécharger sur le Moodle (« Remise des livrables ») sous la forme d'une archive `*.zip` contenant :

- Rapport au format `*.pdf` détaillant les différentes étapes de la construction du détecteur de visages avec une analyse des choix retenus. Toutes instructions particulières nécessaires à l'exécution du code doivent être détaillées dans le rapport.
- Code Python contenant au minimum deux scripts :
 - `train.py` qui doit générer un modèle de classifieur à partir des images d'apprentissage;
 - `test.py` qui doit générer les détections sur les images de test à partir du modèle généré par `train.py`
- Résultats de détection sur les images de test sous la forme d'un fichier texte `detection.txt` :
 - Chaque ligne du fichier correspond à une boîte de détection décrite par six valeurs k, i, j, h, l, s par ligne :
 - k : numéro de l'image
 - (i, j) : coordonnées (ligne, colonne) du coin supérieur gauche de la boîte
 - (h, l) : taille (hauteur, largeur) de la boîte
 - s : score de détection
 - Exemple :

```
001 20 32 64 128 0.23
001 20 32 64 128 0.23
001 34 60 80 160 2.3
002 10 32 32 64 0.73
003 24 120 80 160 1.3
...
```

- **Ne pas inclure** les images d'apprentissage et de test dans l'archive.

Les performances de détection peuvent être obtenues en téléchargeant le fichier de détection sur le « Tableau des scores » accessible depuis le Moodle.

Important : l'exécution du script `test.py` doit pouvoir générer le fichier `detection.txt` fourni dans l'archive. Le script `test.py` doit pouvoir être exécuté sans avoir à exécuter le script `train.py`, cela implique que tous les résultats issus de la phase d'apprentissage **nécessaires** à l'exécution de `test.py` doivent être enregistrés et fournis dans l'archive.

3 Barème

- Rapport : 12 points
- Code : 5 points
- Performance : 3 points
 - top 20% : 3 points
 - top 50% : 2 points
 - top 80% : 1 point
 - en dessous : 0 point
 - trois tentatives « gratuites » puis $-0,5$ point par tentative

4 Consignes

- Le projet est à faire individuellement ou en binôme.
- Le code doit être écrit exclusivement en Python.
- L'utilisation de bibliothèques autres que `scikit-learn` et `scikit-image` est autorisée, mais doit être clairement indiquée dans le rapport.
- Il est **interdit** (sauf à titre de contributions annexes au projet) :
 - d'utiliser un modèle pré-entraîné non issu des données d'apprentissage ;
 - d'utiliser des images d'apprentissage autres que celles fournies pour le projet ;
 - d'utiliser des codes de tierces personnes.