

SYS-809 Vision par ordinateur

Laboratoire 2 : Introduction à OpenCV-Python

1 Objectif du laboratoire

Ce laboratoire offre une introduction à la bibliothèque OpenCV avec le langage de programmation python.

Nous nous intéressons aux points suivants :

- Comprendre la syntaxe et les mécanismes de base du langage de programmation Python.
- Se familiariser avec la bibliothèque OpenCV.
- Apprendre à installer un environnement python.

Tout le laboratoire se fait dans le Jupyter Notebook openCV.ipynb. Un guide pour l'installation de l'environnement de travail sur votre ordinateur est disponible à la section 3, 4 et 5.

2 Python par rapport à Matlab

- Il existe des milliers de modules python accessible gratuitement fait par des tierce-partie disponible avec PyPI, nous allons principalement utiliser OpenCV. Les modules disponibles sur Matlab sont payant et principalement fais par Matlab.
- En python, une faible proportion des fonctions est disponible de base. Il faut importer des librairies pour y avoir accès.
- L'indexation commence à 0 plutôt qu'à 1 dans MATLAB
- Python utilise « # » pour commenter une ligne et MATLAB utilise « % »
- Les deux sont des langages interprétés

3 A-3450

Nous utilisons ce local du département de LOGTI pour avoir accès aux Graphical Processing Unit (GPU) dont sont équipés les ordinateurs de ce local. Les GPU permettent d'effectuer une grande quantité de calcul en parallèle, contrairement aux CPU qui font leurs calculs en série (principalement). Les GPU vont être utilisé pour entraîner un CNN au laboratoire 3 et probablement dans votre projet.

Plusieurs parties de l'installation sont déjà fait sur les ordinateurs du A-3450

- Anaconda est installé
- Visual Studio Code est installé
- Une partie de la procédure à suivre pour utiliser les GPU est faites.

Il vous reste à installer l'environnement python en passant par le « anaconda prompt » (étape 4.2) et installer l'extension « Python » sur Visual Studio Code.

4 Installer python sur un ordinateur windows

1. Downloader anaconda (<https://docs.anaconda.com/anaconda/install/windows/>)
2. Ouvrir anaconda Prompt
3. Créer un nouvel environnement de travail avec python3.9 (utiliser la commande suivante)

```
conda create -n sys809 python=3.9
```

4. Activer le nouvel environnement (utiliser la commande suivante)

```
conda activate sys809
```

5. Utiliser les commandes suivantes (une par une) pour installer les environnements de développement (toujours répondre avec « yes » ou « y »)

```
conda install -c conda-forge jupyterlab
```

```
python -m pip install --upgrade pip  
pip install pillow matplotlib sklearn  
pip install opencv-python
```

Au A-3450

```
conda install tensorflow-gpu
```

Sur un ordinateur personnel (sans GPU)

```
pip install tensorflow
```

5 Installer Visual Studio Code

Télécharger VS code à partir de <https://code.visualstudio.com/download>.

Installer l'« extension » Python (de Microsoft).

6 Jupyter Notebook

Le laboratoire se fait avec un Jupyter notebook. Commencer par lire la page d'aide :

<https://code.visualstudio.com/docs/datascience/jupyter-notebooks>. Ensuite ouvrez le Jupyter Notebook

(extension « .ipynb ») du laboratoire à partir de VS code. Vous devez sélectionner le kernel python sys809 pour exécuter le fichier. Tout le laboratoire se fait dans le notebook. Le code doit être écrit et exécuté dans les cellules de code et les réponses aux questions doivent être écrites dans les cellules Markdown. Les sections à remplir sont intitulées respectivement « Code à écrire » et « Réponses ».

7 Évaluation

Remettre le fichier original renommé avec vos nom (openCV_vos_noms.ipynb) contenant les exercices de programmation et les réponses aux questions. Lors de l'évaluation, le code ne va pas être exécuté. Il faut donc que les « output » des cellules soient enregistré dans le Jupyter notebook. Le barème est disponible à la page suivante. La qualité du code et de l'algorithme est évaluée seulement pour la dernière question. Pour les autres questions écrivez un code lisible et fonctionnel.

8 Grille d'évaluation

Équipe :

Exercices		Commentaires
/10	Graphique	
/15	Lecture, affichage, ...	
/10	Transformations géométriques	
/10	Détection des contours	
/5	Détection et suivi des visages	
/30	Extraction des caractéristiques locales	
/80		