

## HANDS ON AI – Certificat IA de l'UMONS

### Défi 1 : Partie Localisation

#### Manuel d'utilisation de la plateforme Floydhub

## 1. Installation de floydHub :

Afin d'utiliser proprement la plateforme Floydhub, il faudra l'installer en local en utilisant un simple paquet avec « pip ». Pour cela, nous vous conseillons d'utiliser un environnement virtuel tel que « Anaconda ». Vous pouvez télécharger Anaconda depuis le site : <https://www.anaconda.com/distribution/#download-section>

Pour l'installation de Floydhub, il faudra lancer les commandes selon votre environnement choisi :

- Si vous utilisez « conda »

```
conda install -y -c conda-forge -c floydhub floyd-cli
```

- Si vous voulez faire une installation directe avec la commande « pip »

```
pip install -U floyd-cli # Python 2
```

```
pip3 install -U floyd-cli # Python 3
```

- Si vous utilisez virtualenv

```
sudo pip install virtualenv
```

```
virtualenv path/to/floyd
```

```
source path/to/floyd/bin/activate
```

```
pip install -U floyd-cli
```

Afin de tester si l'installation s'est bien déroulée, vous pouvez taper la commande de connexion : **floyd login**

## 2. Chargement d'une base de données sur la plateforme floydHub :

La plateforme floydHub permet de créer, charger et partager des bases de données entre les utilisateurs. Pour charger une base d'images, vous devez suivre les instructions suivantes :

- Aller sur le site <https://www.floydhub.com>, connectez-vous avec votre user et password
- Cliquez sur **Datasets-> New Datasets**
- Donnez un nom sans espace (utiliser les caractères \_ et -)

- Revenez à votre environnement local : Anaconda par exemple
- Activez votre environnement
- Connectez-vous à vos comptes avec « floyd login » et pointer (avec cd) vers le répertoire de votre base de données (sur PC)
- Tapez les commandes suivantes pour charger les données :

**floyd data init user/nom-base**

**floyd data upload**

- Cette opération peut prendre un temps qui dépend de la taille de la base d'images. Vous pouvez également voir la vidéo partagée via Moodle : “**upload\_data.mp4**”

### 3. Création/ouverture d'un projet et lancement de Jupyter notebook :

Pour créer vos projets, il suffit d'aller sur l'adresse : <https://www.floydhub.com> et cliquer sur **Projects->New projects**.

Ensuite, dans l'onglet **Overview**, cliquer sur **Create Worskpace-> Create Worspace**. Vous aurez deux choix :

- from scratch (démarrer à zéro)
- depuis un projet qui est sur github

Dans votre cas, il faudra choisir la deuxième option en fournissant l'adresse du dossier git : [https://github.com/belarbi2733/keras\\_yolov3](https://github.com/belarbi2733/keras_yolov3)

Il faudra ensuite choisir l'environnement. Dans votre cas, ce sera **Tensorflow 1.10** avec la machine **GPU K80**. Notons que si vous voulez faire des tests rapides, vous pouvez utiliser le **CPU**. Par contre, pour l'entraînement, il vaut mieux choisir un **GPU**. Il suffira ensuite de cliquer sur le nom du projet et un nouvel onglet s'ouvrira

A gauche, vous verrez le fichier **Key\_localisations.ipynb**. Un onglet **Jupyter Notebook** s'ouvre pour vous donner la main à le manipuler de la même façon que Google Colab. Vous pourrez donc installer des libraires que vous voulez avec la commande **!pip install ....**

Pour cette partie aussi, vous pouvez voir la vidéo partagée via sur Moodle “**connexion\_création.mp4**”

## 4. Attacher une base de données à votre projet :

Pour attacher une base de données existante sur la plateforme floydhub, nous vous invitons à suivre les instructions partagées via la vidéo “**attach\_data\_sets.mp4**” (Moodle)

## 5. Lancer des jobs avec floydhub :

L’avantage de floydHub (par rapport à Google Colab par exemple) est de pouvoir lancer des jobs (un entraînement par exemple) qui peuvent durer longtemps sans devoir attendre la fin de calcul devant sa machine. En effet, avec floydhub, vous pouvez lancer des jobs depuis vos PC vers la plateforme floydHub. Une fois la tâche terminée, vous recevrez un email indiquant la fin du job en question. Avant de lancer des jobs, il faut savoir que :

- Les jobs doivent être envoyés depuis vos PCs et non pas depuis la plateforme floydhub
- Le projet doit donc être cloné sur vos PCs afin de lancer le **job** et l’envoyer à la plateforme **FolydHub**.
- La taille du projet ne doit pas dépasser les 100 mo
- Si vous avez un fichier ou un dossier qui dépasse les 100 mo, vous ne pouvez pas lancer le job, il faudra donc le charger sur la plateforme comme indiqué dans la vidéo “**upload\_data.mp4**”
- A la fin de chaque **job**, il est possible de récupérer les résultats sur vos machines, ou les utiliser pour un autre job
- Avec le compte que vous avez, vous pouvez lancer un seul job à la fois.

Revenons à notre problème de localisation de clés ou trousseaux de clés. Pour lancer un job, nous vous conseillons de suivre les étapes suivantes :

1. Télécharger le projet depuis Github (git clone) ou cliquer sur Download (fichier zip) :  
[https://github.com/belarbi2733/keras\\_yolov3](https://github.com/belarbi2733/keras_yolov3)
2. Lancer le terminal (sur **Linux**) ou Anaconda (sur **Windows**) et activer l’environnement.
3. Pointer via le terminal vers le projet avant d’initialiser l’environnement :

```
cd keras_yolov3
```

```
floyd init keras_yolov3 #une fenêtre (navigateur) s’affiche pour créer un job
```

#### 4. Lancer l'entraînement via la commande suivante :

```
floyd run --gpu --env tensorflow=1.10 --data
amine2733/datasets/keys_and_background/1:keys_and_background --data
amine2733/datasets/yolo_weights/1:yolo_weights "python train1_job.py --
initial_epoch1 0 --epoch1 100 --batch_size1 32 --annotation 'annotations.csv'
--classes 'model_data/key_classes.txt' --anchors
'model_data/yolo_anchors.txt' "
```

Vous l'avez remarqué, la commande présente plusieurs arguments :

**--gpu** : pour utiliser la carte graphique k80. Pour utiliser une autre carte (v100) il faudra donner l'argument **--gpu2** mais il faut avoir le crédit pour.

**--env** pour spécifier le framework à utiliser (tensorflow)

**--data** pour utiliser une base d'images ou un fichier de poids par exemple

Pour notre problème de localisation, vous pouvez utiliser une base de données déjà annotée et partagée (via floydhub) avant de lancer les jobs. Il faudra seulement adapter les chemins dans le script. Le chemin pour aller à cette base est : **/floyd/input/keys\_and\_background/**

Le même principe doit être appliqué pour le fichier **yolo\_weights.h5** qui est disponible dans **amine2733/datasets/yolo\_weights/1**. Ce dernier est partagé avec **/floyd/input/yolo\_weights**

Une fois le job terminé, vous recevrez un email notifiant la fin du job. Notons que pour le problème de localisation, il faudra lancer un deuxième job avec le fichier **train2\_job.py**, mais il faut adapter le code afin que le code « train2 » puisse utiliser le fichier de poids généré à partir du premier **job**.

Pour utiliser le résultat d'un job, il suffit d'utiliser **--data nom\_complet\_job/dossier:dossier**. Pour utiliser ce dossier dans le code, on peut le récupérer via le chemin : **/floyd/input/dossier**

Pour plus de documentation, vous pouvez voir les deux sites :

<https://docs.floydhub.com/>

<https://medium.com/@egao/getting-started-with-floydhub-c4659316ad4b>