

Le Debruiteur - Guide utilisateur

- Jonas Freiburghaus
- Romain Capocasale
- He-Arc, INF3dlm-a
- Image Processing course
- 2019-2020

Liste des packages utilisé

- python = 3.7.4
- cv2 = 4.2.0
- numpy = 1.17.4
- tensorflow = 2.1.0
- keras = 2.3.1
- matplotlib = 3.1.1
- pandas = 0.25.1
- skimage = 0.16.2
- PIL = 6.2.0
- tqdm = 4.36.1
- jupyter-notebook = 6.0.1

Installation

Les différents packages nécaissaire seront installés dans un environnement virtuel. Tout d'abord, il faut avoir `Python` installé sur sa machine.

PIP

Premièrement, il faut s'assurer d'avoir `PIP` installé sur sa machine avec la commande : `$ pip --version`.

Si `PIP` n'est pas installé, il peut être installé via le script `get-pip.py` présent dans le répertoire `user_guide` avec la commande `$ python get-pip.py` ou via le [site web officiel de PIP](#).

Virtualenv

Une fois `PIP` installé, veuillez exectuer la commande `$ pip install virtualenv` pour installer `virtualenv`.

Par la suite, un environnement virtuel peut être créé avec la commande : `$ virtualenv [env_name]`.

Une fois l'environnement créé, l'environnement virtuel peut être lancé via la commande : `$. [env_name]/Script/activate` ou si cela ne fonctionne pas avec : `$ source [env_name]/Scripts/activate`.

Requierments

Une fois l'environnement virtuel lancé, veuillez exectuer la commande : `$ pip install -r requierments.txt` pour installer les packages nécaissaire au projet. Ce fichier est présent dans le dossier `user_guide`.

Arborescence

- `debruiteur` : le dossier contient la librairie créé pour le projet
 - `generator` :

- **datagenerator.py** : contient une classe fournissant les fonctionnalités équivalente à un generateur python mais pour les images. Cette classe est utile pour fournir des images lors de l'entrainements des réseaux de neurones.
- `metrics` :
- **metrics.py** : contient des méthodes permettant de comparer des images entre elles selon les différentes métriques défini pour le projet.
- `models` :
- **autoencoder.py** : contient les méthodes permettant de construire l'architecture des réseaux de neurones de type autoencoder (Dense et Convolution).
- **blocks.py** : contient les méthodes permettant de construire les différents blocs du réseau de neurones.
- **gan.py** : contient les méthodes nécessaires à la construction du réseau de neurones Generative Adversarial Network.
- `noise` :
- **noise.py** : contient les différentes classes permettant d'ajouter du bruit sur les images.
- **noisereduction.py** : contient les différents filtres permettant de réduire le bruit sur les images
- `plots` :
- **plots.py** : contient différentes méthodes utilitaires pour l'affichage des images.
- `preprocessing` :
- **preprocessor.py** : contient les différentes méthodes permettant de charger les images, de les redimensionner et d'y ajouter le bruit. Contient également les méthodes permettant de créer les différents dataframes avec le chemin des images.
- `statistics` :
- **statistics.py** : contient les méthodes permettant de calculer les différentes statistiques sur les méthodes de réduction de bruits.
- `utils` :
- **utils.py** : contient différentes méthodes utilitaires au projet.
- `images` : contient les images initiales.
- `resized_images` : contient les images redimensionnées (ce dossier se crée après avoir exécuté la cellule en question)
- `noised_images` : contient les images redimensionnées (ce dossier se crée après avoir exécuté la cellule en question)
- `saved_models` : contient les modèles entraînés
- `user_guide` : contient le guide utilisateur
- `LeDebruiteur.ipynb` : notebook jupyter pour l'entraînement des modèles. Il n'est pas nécessaire d'entraîner les modèles à chaque fois. Une fois qu'un modèle est entraîné, il est enregistré dans `saved_models` et peut être rechargé à tout moment.
- `EDA.ipynb` : notebook jupyter mettant en œuvre toutes les méthodes de la librairie créée pour ce projet.

Exécution des notebooks

Les notebooks peuvent être exécutés avec la commande : `$ jupyter notebook`. Une fois le notebook en question choisi, il faut se rendre sous l'onglet `Noyau` -> `Changer de noyau` et vérifier que l'environnement créé précédemment est bien activé pour le notebook.