# Sources

## Scripts

Github : <https://github.com/pat92fr/Carre92/tree/master/02%20-%20Logiciel%20PC/03%20-%203DCNN%20Pipeline>

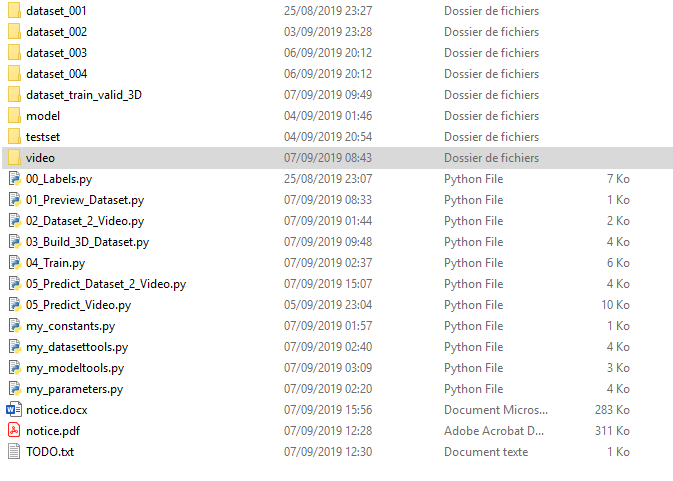
## DATASET SOURCE

Onedrive : <https://1drv.ms/u/s!AgPL1p8-kSaRg9JQD1uURRMgiD2Low?e=h9OGPs>

Télécharger et décompresser les fichiers dataset\_00n.rar

## INSTALLATION

Le répertoire de travail doit contenir les éléments suivants :



Créer les dossiers manquants à la main, notamment « dataset\_train\_valid\_3D », « model », et « video ».

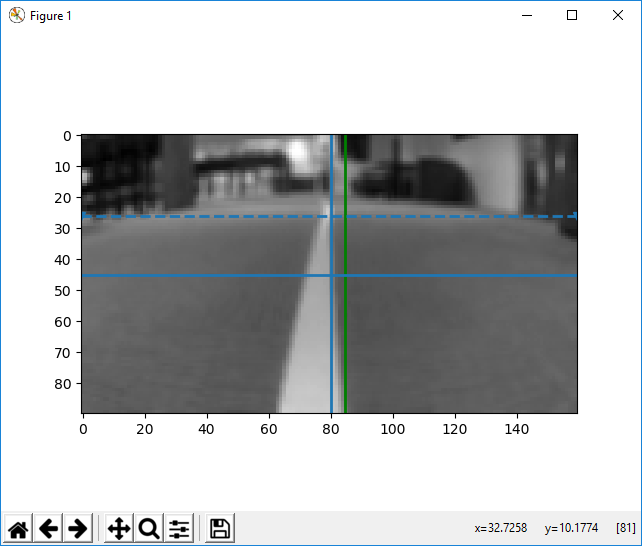
## Fichier de configuration

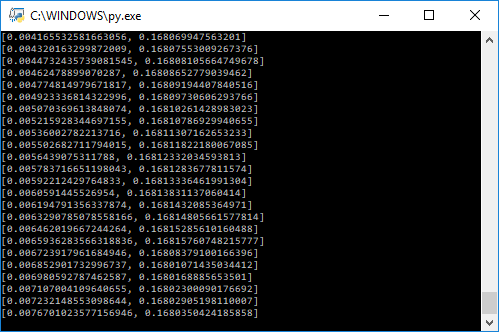
Le fichier « **my\_parameters.py** » contient toutes les données de configuration de toutes les étapes.

# Procédure

## Outil de visualisation des dataset SOURCE (optionnel)

Lancer « 01\_Preview\_Dataset.py » pour visualiser l’ensemble des DATASET. La sortie standard affiche la valeur des commandes de DIRECTION et de GAZ pour chaque image. La fenêtre graphique affiche l’image prétraitées (filtrage flou + N&B) avec un repère (bleu) et la commande de direction (vert). La ligne pointillée bleue délimite la zone supérieure de l’image, éliminée avant traitement par le CNN.

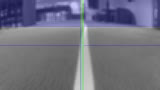




## Conversion des DATASET SOURCE en une vidéo (optionnel)

Lancer « 02\_Dataset\_2\_Video.py » pour convertir l’ensemble des DATASET en un seul média. Le fichier est disponible dans le dossier « vidéo » sous le nom « dataset\_preview.avi ». Le bandeau supérieur bleu correspond à la zone de l’image, éliminée avant traitement par le CNN.

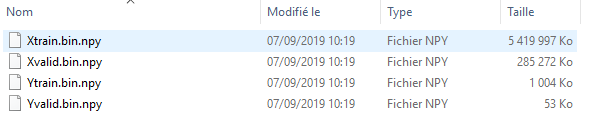




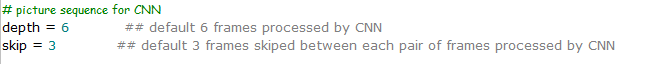


## Création dES DATASET 3D TRAIN-VALID (UNE SEULE FOIS)

Lancer « 03\_Build\_3D\_Dataset.py » et patienter ! Les fichiers binaires contenant les séquences d’image (DATASET 3D) sont générés et stockés dans le dossier « dataset\_train\_valid\_3D ».

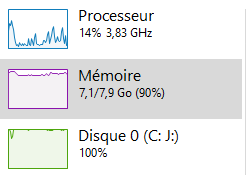


La génération et l’augmentation des DATASET se basent sur les paramètres du fichier « my\_parameters.py », notamment :



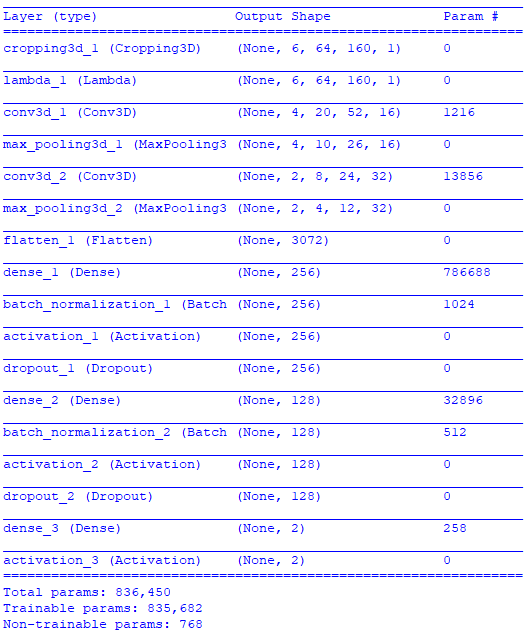


*Prérequis : optimiser/vérifier la configuration du swap. La taille des* DATASET *en mémoire dépasse 16Go.*

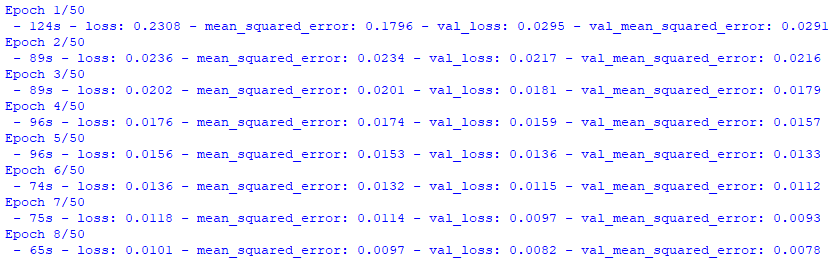


## Apprentissagedu 3D CNN (REPETER à chaque changement de parametres)

Lancer « 04\_Train.py » et patienter ! Au lancement, l’architecture du CNN est affichée.

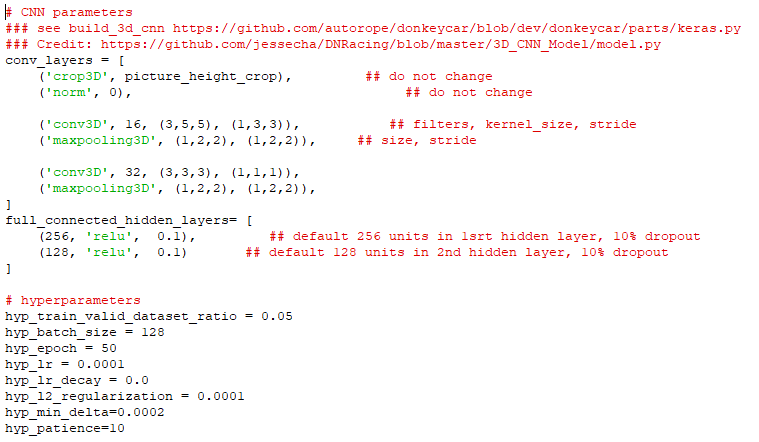


Pendant l’apprentissage, le cout et l’erreur est affichée pour les DATASET d’entrainement et de validation, à chaque fin d’EPOCH.



A la fin de l’apprentissage, le modèle et l’historique sont générés et stockés dans le dossier « model ».

L’apprentissage se base sur les paramètres du fichier « my\_parameters.py », notamment :



*Note : Les paramètres “conv\_layers » et « full\_connected\_hidden\_layers” sont à adapter en utilisant les mots clés : « conv3D », « maxpooling3D », « avgpooling3D », « dropout », « batchnorm » tirés de l’API Keras.*

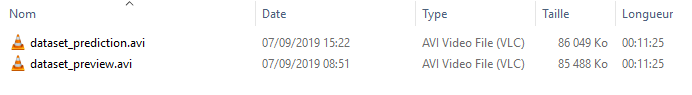
*Note : Le paramètre « epoch » permet de limiter le nombre d’itération d’apprentissage.*

*Note : Le paramètre “lr” correspond au “Learning Rate”*

*Note : Les paramètres « min\_delta » et « patience » sont exploités par la fonction de « Early Stopping ».*

## Prédiction appliquée aux DATASET SOURCE (REPETER à chaque APPRENTISSAGE)

Lancer « 05\_Predict\_Dataset\_2\_Video.py » pour convertir l’ensemble des DATASET en un seul média, incorporant à la fois la commande manuelle et la commande générée par le 3D CNN.



Le fichier est disponible dans le dossier « vidéo » sous le nom « dataset\_prediction.avi ». Le bandeau supérieur bleu correspond à la zone de l’image, éliminée avant traitement par le CNN. La ligne verte correspond à la commande manuelle de direction (DATASET source). La ligne rouge correspond à la commande automatique de direction (Prédiction donnée par le 3D CNN entrainné).

# Outils de vérification

## Visualisation de la Distribution de la commande de direction

## Visualisation du Volume en entrée du 3D CNN