Description of subfolders and files in folder Video_dataset

Last update: 23/01/25

By: Florian David (florian.david@epfl.ch)

Subfolders:

- aal_SPM12: Contains the anatomical atlas (see visualisation.py)
- encoder_dataset_6660: Contains the splited dataset (train/test/val) for fMRI mask of 15364 voxels (before SNR)
- encoder_dataset_6661: Contains the splited dataset (train/test/val) for fMRI mask of 4609 voxels (after SNR)
- fMRIs_schaefer1000_15364: Contains fMRIs from the 30 subjects (and average) under the 15364 voxels mask
- fMRIs_schaefer1000_4609: Contains fMRIs from the 30 subjects (and average) under the 4609 voxels mask
- processed_videos: Contains the preprocessed videos
- Trash: Contains old notebooks, folders, and model weights that were used throughout the project

Notebooks:

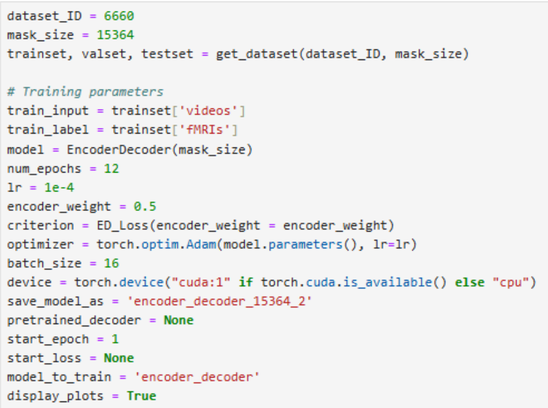
- Find_best_model.ipynb: cross validation method (might not be well defined, see again how to define it properly)
- GPU_memory.ipynb: in case there are issues with CUDA memory
- run.ipynb: main notebook containing the useful pipelines
- Schaefer 1000 mask.ipynb: explorations related to the masking method
- Splitting.ipynb: explorations related to the splitting method
- Statistical analysis.ipynb: explorations related to the statistical tests

Python files:

- dataset.py: functions related to the preprocessing of the dataset
- imports.py: all necessary imports
- models.py: model classes definitions, training and testing loops
- visualisation.py: plotting functions

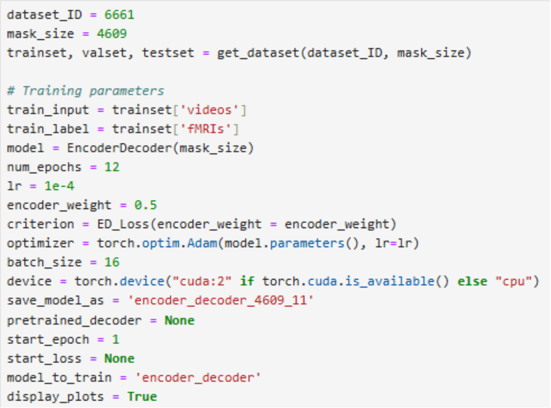
Trained models:

- encoder_15364_2: best encoder model for the 15364 voxels mask (dataset_ID = 6660).
- encoder_4609_2: best encoder model for the 4609 voxels mask (dataset_ID = 6661).
- encoder_decoder_15364_2: best encoder-decoder model for the 15364 voxels mask (dataset_ID = 6660). Hyperparameters: see below left
- encoder_decoder_4609_11: best encoder-decoder model for the 4609 voxels mask (dataset_ID = 6661). Hyperparameters: see below right



```
dataset_ID = 6660
mask_size = 15364
trainset, valset, testset = get_dataset(dataset_ID, mask_size)

# Training parameters
train_input = trainset['videos']
train_label = trainset['fMRIs']
model = EncoderDecoder(mask_size)
num_epochs = 12
lr = 1e-4
encoder_weight = 0.5
criterion = ED_Loss(encoder_weight = encoder_weight)
optimizer = torch.optim.Adam(model.parameters(), lr=lr)
batch_size = 16
device = torch.device("cuda:1" if torch.cuda.is_available() else "cpu")
save_model_as = 'encoder_decoder_15364_2'
pretrained_decoder = None
start_epoch = 1
start_loss = None
model_to_train = 'encoder_decoder'
display_plots = True
```



```
dataset_ID = 6661
mask_size = 4609
trainset, valset, testset = get_dataset(dataset_ID, mask_size)

# Training parameters
train_input = trainset['videos']
train_label = trainset['fMRIs']
model = EncoderDecoder(mask_size)
num_epochs = 12
lr = 1e-4
encoder_weight = 0.5
criterion = ED_Loss(encoder_weight = encoder_weight)
optimizer = torch.optim.Adam(model.parameters(), lr=lr)
batch_size = 16
device = torch.device("cuda:2" if torch.cuda.is_available() else "cpu")
save_model_as = 'encoder_decoder_4609_11'
pretrained_decoder = None
start_epoch = 1
start_loss = None
model_to_train = 'encoder_decoder'
display_plots = True
```

Masks (see Schaefer 1000 mask.ipynb and dataset.py for more information):

- mask_schaefer_15364.npy: contains voxels from the visual regions based on the scheafer parcellation
- mask_schaefer_4609.npy: contains the top 30% voxels with highest SNR among the 15364 voxels covering visual regions

Other files:

- checkpoint.pth: involved in the training loop in case the kernel crashes
- mask.nii.gz: contains the old mask with 4330 voxels (not well defined)
- saliency_map.npy: saliency values for each voxel of the mask, generated within the testing loop
- Schaefer2018_1000Parcels_Kong2022_17Networks_order_FSLMNI152_2mm.nii.gz: parcellation used to define the mask with 15364 voxels