Visual decoding project
Work environment description

Current environment: /home/ralmeida/miniconda3/envs/machinelearnin

In case it is deleted from home, there is a file in
/media/RCPNAS/MIP/Michael/students_work/rodrigo
called machinelearnin_backup.yml , which can be used to restore it, running
conda env create --name machinelearnin -f machinelearnin_backup.yml

In any case, all the packages needed in the environment and their versions are in a file called packages.txt

Modules:
dataset_new.py - functions to load and preprocess the dataset
face_detection2.py - functions to detect faces
imports.py - all the necessary imports
models_new_2.py - defining model classes, functions to train and test models
visualisation_new_2.py - functions to see the image reconstructions and metrics
perturbation.py - functions to perturbate models
run_overnight.py and others with similar names - scripts to leave running overnight to train different types of models

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
new_loss.ipynb - testing a loss related to face detection
new_mask.ipynb - creating custom masks
new_run.ipynb - current main notebook
old_run.ipynb - old main notebook

perturbation_new.ipynb - testing perturbation method
run.ipynb - basic pipeline for loading data, training a model and testing it
Schaefer 1000 mask.ipynb - original notebook for masking
Setup_newfmri.ipynb - get fmri data (to numpy) with all voxels
Splitting.ipynb - explorations related to the splitting method
Statistical analysis.ipynb - explorations related to the statistical tests
.~Forrest Gump preprocessing.ipynb - explorations related to trying to preprocess Forrest Gump to be able to use it

Trained models:
temporal_decoder_4609_x_y_z - decoder accounting for temporal features using 4609 voxels, trained for x epochs, with a TR window size of y, and l1 regularization of z
decoder_4936_x - decoder for data with 4936 voxels, trained for x epochs
decoder_all_4609_x - decoder trained on all subjects (other models trained on just the average), for data with 4609 voxels, trained for x epochs
encoder_15364_2 - best encoder model for the 15364 voxels mask (dataset_ID = 6660)
encoder_decoder_4609_11 - best encoder-decoder model for the 4609 voxels mask (dataset_ID = 6661)

Best decoder models: decoder_4609_350 ,
temporal_decoder_4609_352_TRwindow3_l1_1e-08

Subfolders:
aal_SPM12 - Contains the anatomical atlas
auto_extracted - to store relevant extracted parts of images
batch_results - to store faces
decoder_dataset_all_subjects_sub-S02 - train, test and validation set for subject 2
encoder_dataset_6660 - Contains the splitted dataset (train/test/val) for fMRI mask of 15364 voxels (before SNR)

encoder_dataset_6661 - Contains the splitted dataset (train/test/val) for fMRI mask of 4609 voxels (after SNR)
extracted - for testing the grabcut algorithm for getting relevant parts of images
faces_output - similar to batch_results
fmris_4609+ffa - all TRs of each video, with the 4609 voxels + voxels from the ffa mask
fmris_all_voxels - all TRs of each video, with all voxels
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
middle_frames_original_size - contains the original frames at the middle of each TR for every video, in their original resolution
my_faces - similar to batch_results, used while trying algorithms
outputs - to store image reconstructions and metrics
processed_data - train and test sets for every subject, with 4609 voxels
processed_data_new - train and test set with 4936 voxels (just for subject 1)
processed_videos - all videos processed to 112x112 resolution and only 1 frame per TR (not split between sets)
Trash - Contains old files

Masks:
enhanced_LFFA.npy - mask for the left FFA (fusiform face area) plus the reflection of the mask of the right FFA
enhanced_RFFA.npy - analogous to the previous one, for the other side
enhanced_union_FFA.npy - union of the two previous masks
LFFA_roi.nii.gz - mask for the left FFA, not resampled to match our fMRI data
mask_4609+ffa.npy - mask for the top 30% voxels with highest SNR among the 15364 voxels covering visual regions plus the FFA
mask_4609+ffa+ppa.npy - previous mask plus the PPA (parahippocampal place area)
mask_schaefer1000_15364.npy - contains voxels from the visual regions based on the schaefer 1000 parcellation

mask_schaefer1000_4609.npy - contains the top 30% voxels with highest SNR among the 15364 voxels covering visual regions
ppa_prob.nii - mask for the PPA, not resampled to match our fMRI data. Each entry has a value between 0 and 1 related to its importance to the PPA (the area is still being researched)
region_ids_15364.npy - 1D array with 15364 elements containing the number/id of the region from the atlas that each voxel in the 15364 mask belongs to
region_ids_4609.npy - analogous to previous array, but for the 4609 mask
region_ids_4609+.npy - 1D array with 902629 (=91*109*91, number of voxels in each TR) elements, all with 0s except in the locations corresponding to the voxels of the 4609 mask
region_ids_900k.npy - analogous to previous one, but for the 15364 mask
region_ids_all2.npy - 1D array with 902629 elements containing the number/id of the region from the atlas that each voxel belongs to
resampled_lffa.npy - mask for the left FFA resampled to match our fMRI data
resampled_ppa0.5.npy - mask for the PPA, only with voxels with value 0.5 or above in the ppa_prob mask
resampled_ppa0.9.npy - analogous to previous one, with 0.9 instead of 0.5
resampled_ppa.npy - ppa_prob resampled to match our fMRI data
resampled_rffa.npy - mask for the right FFA resampled to match our fMRI data
RFFA_roi.nii.gz - mask for the right FFA, not resampled to match our fMRI data
Schaefer2018_1000Parcels_Kong2022_17Networks_order_FSLMNI152_2mm.nii.gz: parcellation used to define the mask with 15364 voxels

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

Other files no longer used were moved to the Trash folder

Last updated 03/07/2025 Rodrigo Anjos rodrigo.anjos@epfl.ch