ADAPTED *VOXELMORPH* DOCUMENTATION

Sarah Liu

DEPENDENCIES:
Code was built using the following package versions:

- Numpy (1.24.3)
- OpenCV (opencv-python=4.10.0.84)
- Pathlib (1.0.1)
- Neurite (0.2)
- SciPy.io (scipy=1.13.1)
- Tqdm (4.66.4)
- PIL (pillow=10.3.0)
- Pytorch (torch=2.3.1+cu118)
- Sklearn (scikit-learn=1.5.0)
- Seaborn (0.13.2)
- Tifffile (2024.5.22)
- Matplotlib (3.9.0)
- VoxelMorph (0.2)

FILE DESCRIPTIONS:
There are two types of executable files: ***train*_{suffix}*.py*** and ***test*_{suffix}*.py***. Use 'debug' parameter to enable faster execution of the whole script (shortened data loading and training).

train*_{suffix}*.py files are run to create and train a new instance of VoxelMorph:

- Import statements
- Model hyperparameters: all file paths and training parameters should be edited here
- Data preprocessing: the data is organized into 'moving' (source) and 'fixed' (target) numpy array datasets and then fed into custom generators
    - Dataset processing is currently customized to different file setups using specific folder names/structures; may need to edit to fit specific data
- Model creation: the model is defined here, model layers, optimizer and loss functions can be edited here
- Training: Loops through epochs; each epoch has a training and validation stage
    - Calculates simulation loss ($\mathcal{L}_{sim}$: penalizes differences between predicted and real) and smoothing loss ($\mathcal{L}_{smooth}$: penalizes spatial differences in deformation field ϕ) for each batch, adds losses together to represent loss for entire batch, which is backpropagated through model
        - Losses can be changed/defined in the ***losses.py*** file
    - Average loss across all batches represents loss for the epoch
    - Following training stage, validation stage calculates loss for each validation batch and averages to find validation loss for the epoch
        - Validation loss for each epoch is recorded and saved to 'val_losses.txt'
    - If epoch validation loss is less than previous, save new model weights
        - New weights are saved to 'model_weights.pth'

test*_{suffix}*.py files are used to benchmark the model and follow a similar structure to the training files:

- Import statements
- Model hyperparameters

- Data preprocessing
- Model loading: the model is loaded here from a previously-specified weights file
- Prediction: Loops through testing generator, predicts deformation field ϕ and generates "moved" image (source or "moving" image convolved with ϕ, targeted to be similar to consecutive or "fixed" image)
    - Predicts flow for each image pair, then sums all flow values to find cumulative flow for entire sequence
- Visualization: includes several plots to visualize deformation field and displacement estimation

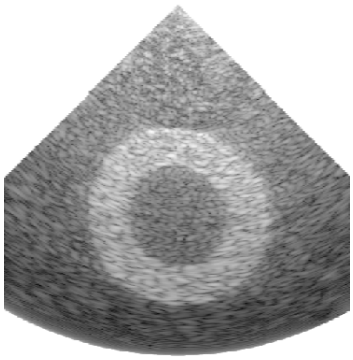
generators.py and ***losses.py*** are Voxelmorph scripts that have been changed by implementing custom losses/generator objects. All the executable files use generators.custom_generator to feed image pairs into the model.

main.py is for visualizing a series of true/predicted images next to each other using a custom class SliceViewer. Right now they are set up to show one whole circular systole simulation, but can be altered to show individual before/after predictions.

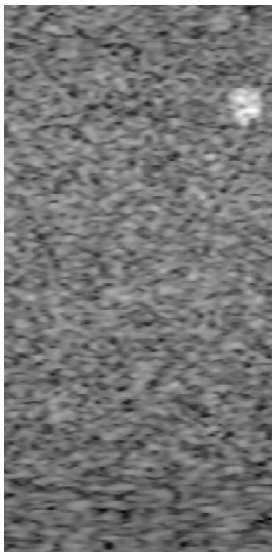
I don't remember what has to happen with the ***__init__.py*** and ***setup.py*** files, but I think they have to be in the directory for the code to work.

{suffix} KEY AND DATA FORMATTING:

- ***_jad*** files are VoxelMorph training/testing files for the circular systole heart simulation
    - One training file is built to process B-modes, other for the RF data
    - Data format: Data is organized into five overarching folders. Within these general folders, individual folders each hold the .mat data for single sequences, including B-mode data, RF data and the simulation masks. One sequence pictures the simulation from the beginning to end of 'systole.'

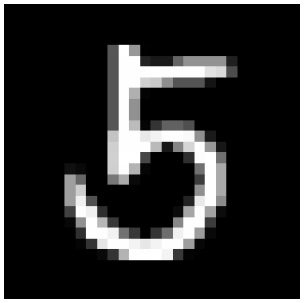


- ***_sim*** files are training/testing files for the moving spot simulation B-modes
    - Image frames are cropped so that only the simulation appears; excludes matlab axes and border
    - Data format: Vertical and horizontal sequences are organized into folders labeled by the movement direction and coordinates. Within folders, ordered .png images are labeled with coordinates of the pictured inclusion.



- ***_CAMUS*** files are for the CAMUS dataset
    - ‘img_test_2CH_contrast’: Images are contrast-enhanced using ImageJ and cropped to 512 x 512 pixels
    - Data format: Ultrasound scans are organized into a training/validation folder and a testing folder. Each scan is a 3-dimensional .tif file, where the heart is pictured from the beginning to end of systole.

- ***_mnist*** files are tester files built to verify that VoxelMorph is working correctly
    - Downloads MNIST dataset directly from sklearn databases
    - Data format: Data is downloaded from keras online databases.



- ***_greg_rf*** file is for applying VoxelMorph to Greg’s mystery data