Deepdefacer

Deep Learning for MRI Anonymization

Anish Khazane
Stanford Center for Reproducible Neuroscience

Motivation

- Growing storage and dissemination of MRI images for research, publication, etc raises privacy concerns as neuroimaging data can reveal facial information.
- HIPAA defines any full photographic images as protected health information, and mandates de-identification prior to sharing of the data

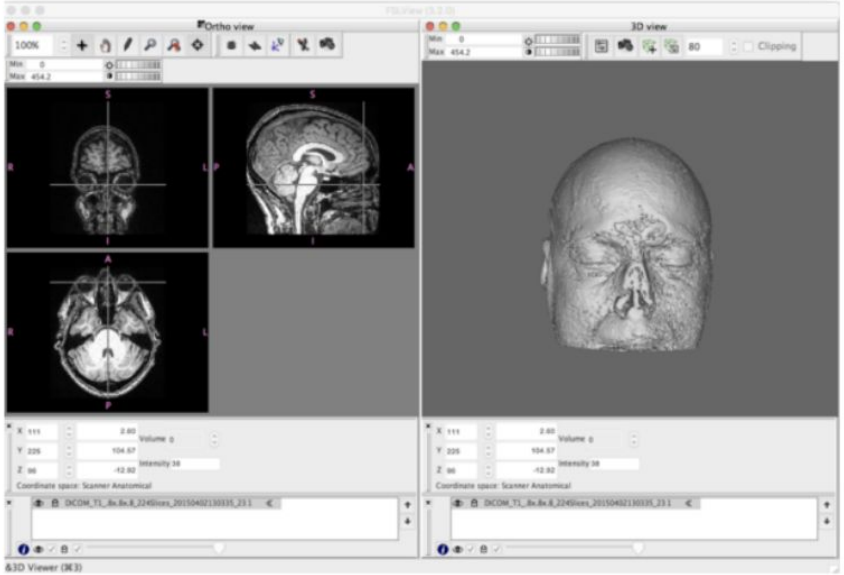
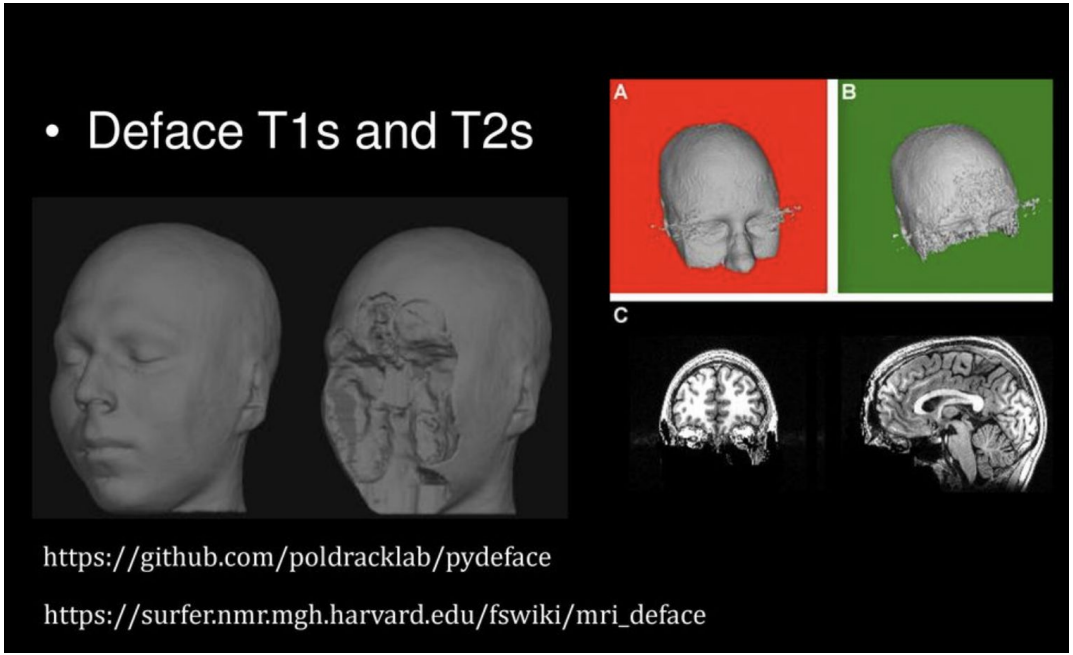


Fig. 1 An example of a 3D reconstructed image from a MRI scan.

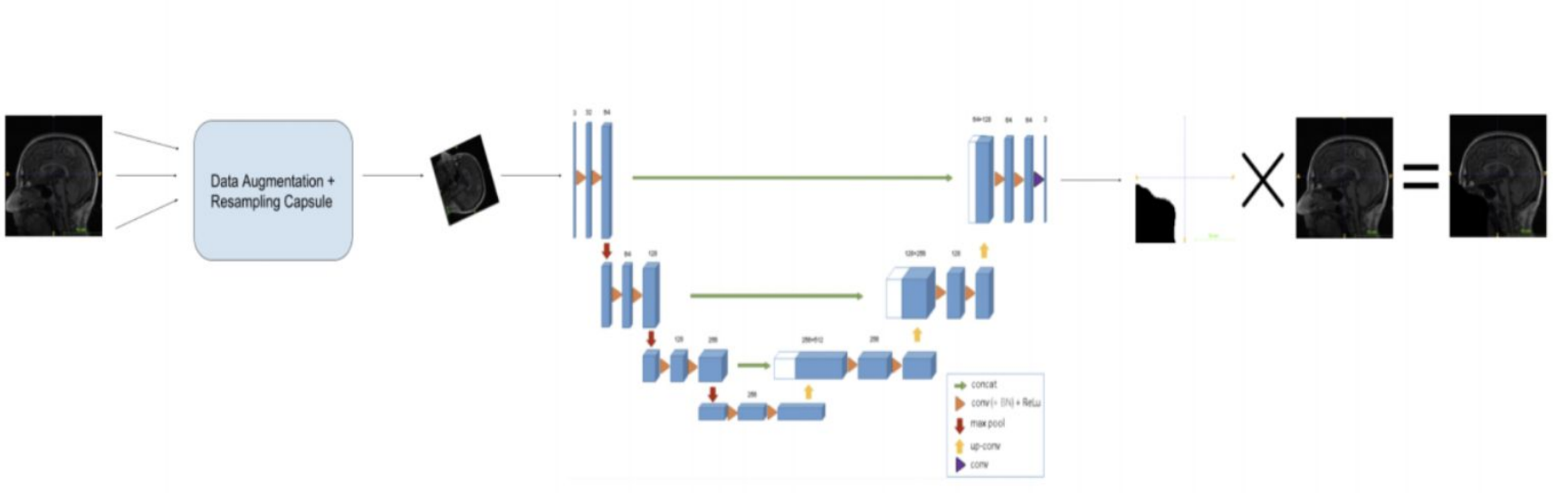
Traditional Approaches for MRI Anonymization

- Pydeface, MRI_Image, etc are convex hull based approaches that are very computationally expensive.
- On average ~ 4 minutes to deface an 3D MRI Image, with nearly 500-1000 images in a dataset, that's roughly 50 hours!



Deep Learning for MRI Anonymization

- We used a deep-learning based approach to recognize a ROI that covers the facial structure of these MRI scans, and then generated a binary mask that could be multiplied against the MRI image to deface it.

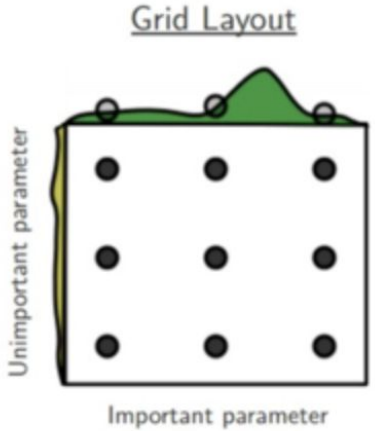


Summary of Approach

- Used a 3D “U-NET” image segmentation network for this task; this is a popular network for biomedical image segmentation, partly due to its capability to take in any image size.
- Data augmentation capsule used to turn roughly 1500 MRI images into 20,000. Used the following techniques
    - Spatial rotations of 3D Images
    - Spatial scaling of 3D Images
    - Change resolution / quality of images
- Resampling module an optimization trick for applying DL on large images
    - E.g 256 x 256 x 256 image resampled down to 64 x 64 x 64, passed into network, then resampled back up to 256 x 256 x 256.

Modeling Tricks

- Overfitting was a common problem. Used a grid search to search for optimal filter sizes, number of layers, etc for network.
- Data augmentation + resampling helped with model generalization
- *Changing optimization objective from directly de-identifying the MRI to creating binary masks sped up training.



Results

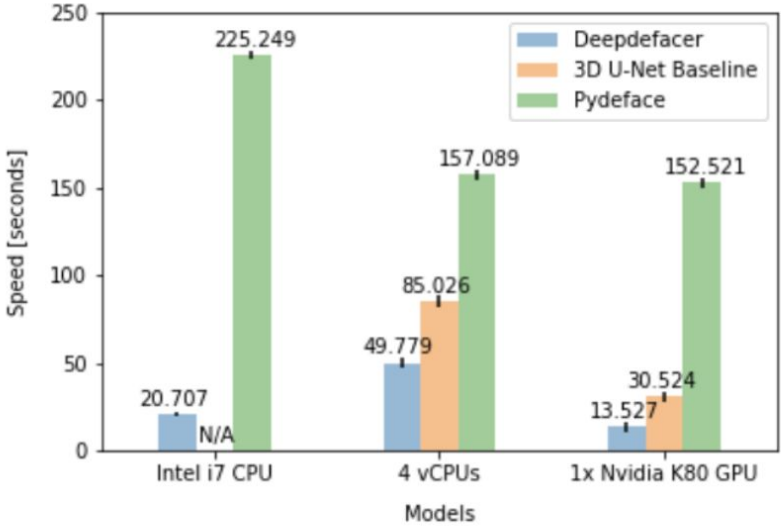


Fig. 6 Average speed test results across commercial CPU and GPU devices. The 3D U-Net baseline model crashes while running on an Intel i7 CPU, and Pydeface does not significantly benefit from running computations on GPU.

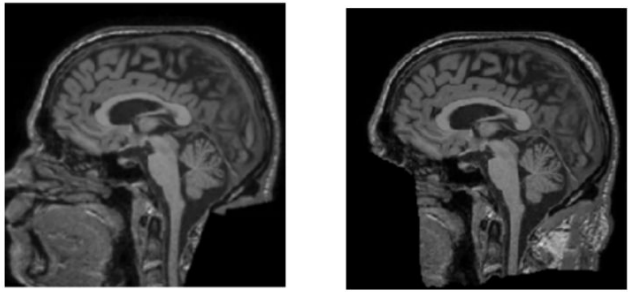


Fig. 8 (a) is a mislabeled output from the Pydeface program, while (b) is the same output from Deepdefacer. While Pydeface is unable to generalize well to all types of MRI images, Deepdefacer is still able to create accurate masks relying on facial patterns rather than intricate voxel details within an image.

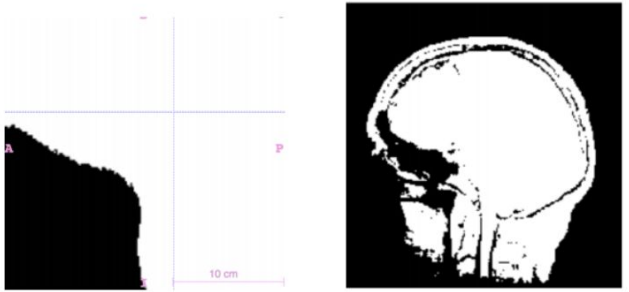


Fig. 9 (a) is a segmentation map produced by Deepdefacer (b) is a segmentation map produced by the 3D U-Net baseline. Axis in (a) follow the same definition listed under Figure 4.

Suggestions for Deep Learning Projects

- Use data augmentation if you have trouble with overfitting ~ scaling, changing resolution quality, rotating, etc can replicate images without gathering any additional data
- Grid search is a wonderful tool for hyperparameter search
- Having optimization trouble? Consider an alternative (but related) objective that's easier to train on.
- We ran all of our experiments on Google Cloud's free-tier GCP clusters ~ this gave us $300 or so in free credits to run training jobs.

https://cloud.google.com/free

Thank you!