Lab Visualization and Medical Image Analysis
WiSe 2021-22

Entropy Guided Unsupervised Domain Adaptation for Segmentation of Brain MRI Scans

Rohil Rao, Sanket Shah

Supervisor : Rasha Sheikh

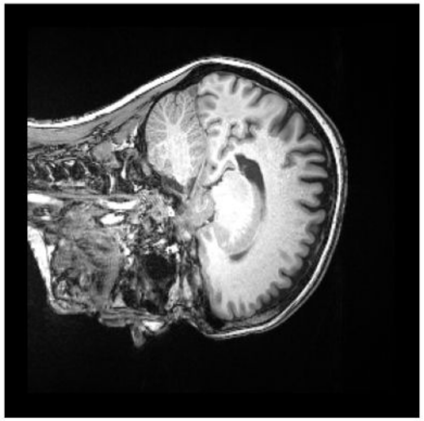
UNIVERSITÄT BONN

Outline

- Motivation
- Overview of Reference Paper
- Architecture
- Methods
- Dataset
- Experiments
- Results & Conclusion

Motivation

- Image segmentation is a crucial part of medical diagnosis and research.
- Brain segmentation used for detecting brain diseases. (eg: Alzheimers and Parkinsons)
- Manual segmentation is accurate but time consuming, expensive, impractical and non-uniform.
- Automating brain segmentation is hence important.



Brain MRI Scan

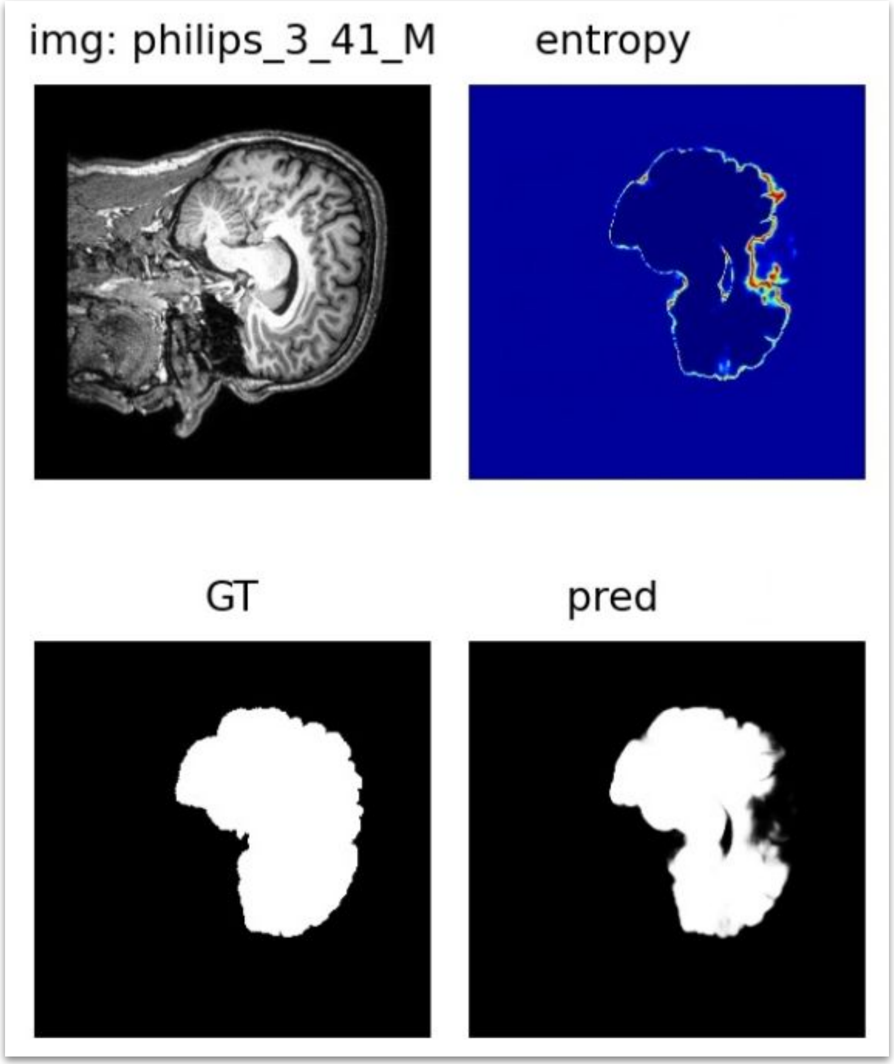
Segmentation mask

Motivation

- Feature based Pattern Recognition
  - Simple to implement.
  - Requires feature extraction from images.
  - Not accurate.
- Deep Learning based methods
  - Highly accurate.
  - Require large amount of data.
  - Expensive to collect labelled data.
  - Suffers from domain shift problem.

Motivation

- What is Domain Shift:
  - Deep learning model trained on scans from one center perform poorly on scans from other centers.
- Why does it occur?
  - Difference in imaging devices
  - Difference in image acquisition process.



Terminology

- Source Domain
  - Refers to the center from where the data is used for training the original image segmentation model.
- Target Domain
  - Refers to the other centers from where the data is used for evaluating the model trained on source domain.
- Domain Adaptation
  - Adapt model trained on source domain to perform equally well on target domain
- Unsupervised Domain Adaptation
  - Uses only source images, labels and target images.

Related Work

- Two major approaches solve domain shift :
  - Image Adaptation
    - Generate source like images from target images.
    - Image synthesis quality is bottleneck
  - Feature Adaptation
    - Aligns model features for target domain to features of source domain.
    - Adversarial Training

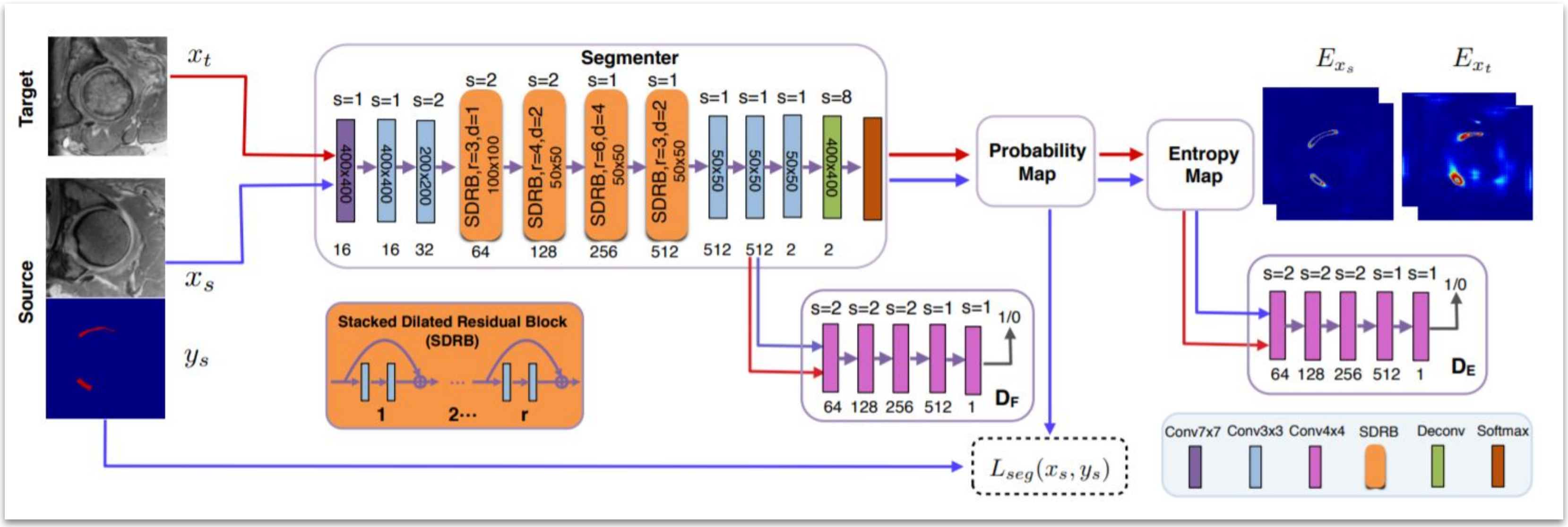
Reference Paper

- “Entropy Guided Unsupervised Domain Adaptation” by Zeng G. et.al. [1] uses feature and entropy based unsupervised domain adaptation (UDA) using adversarial training.
- Use feature and entropy map discriminators to align source and target, and hence reduce domain shift.
- In this lab, we implement the approach suggested in this paper with some changes.

Objective

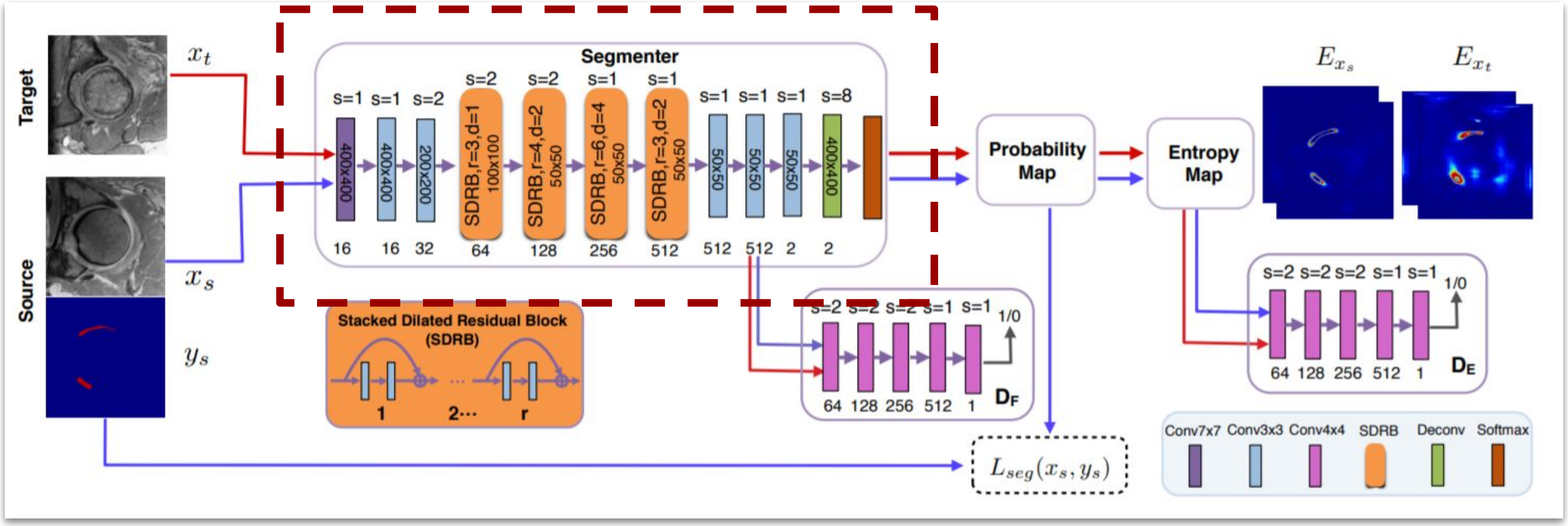
- The objectives of the lab were :
  - Implement the reference paper with some differences in architecture and using a different dataset.
  - Experiment with the hyperparameters to obtain best results.
  - Add own contributions to implementation.

Overview of Reference Paper



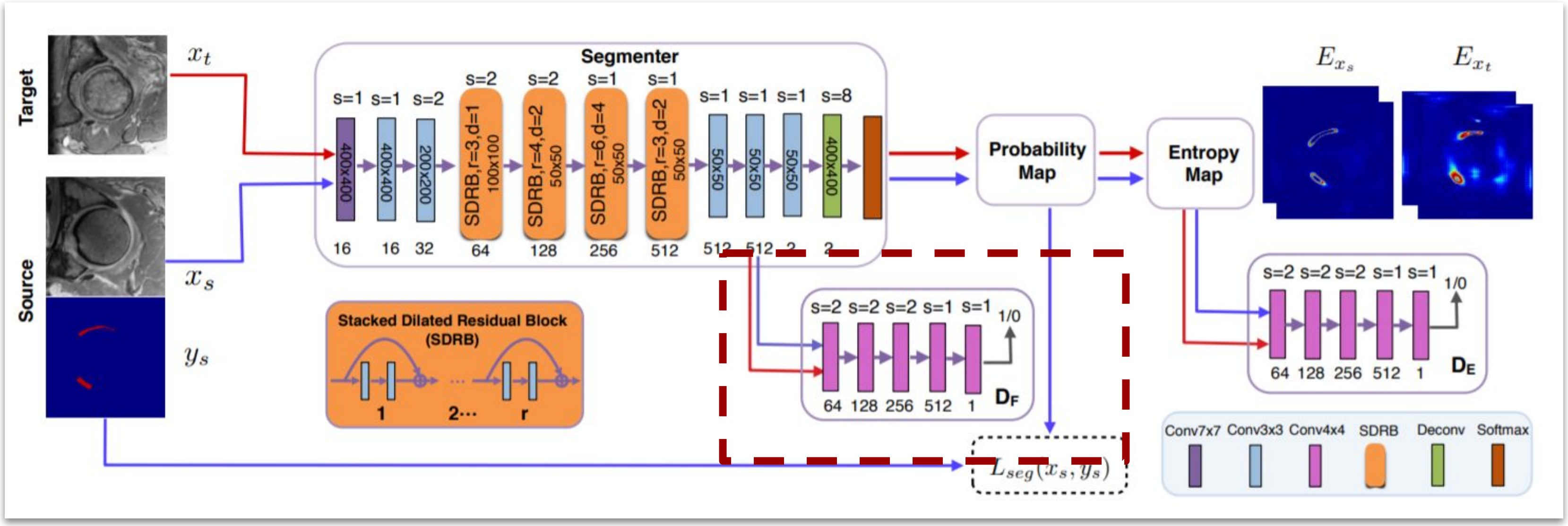
- The architecture consists of three main components :
    - Segmentor
    - Feature Map Discriminator
    - Entropy Map Discriminator
- The model takes source scans (with ground truth) and target scans (without ground truth) as input.

Overview of Reference Paper



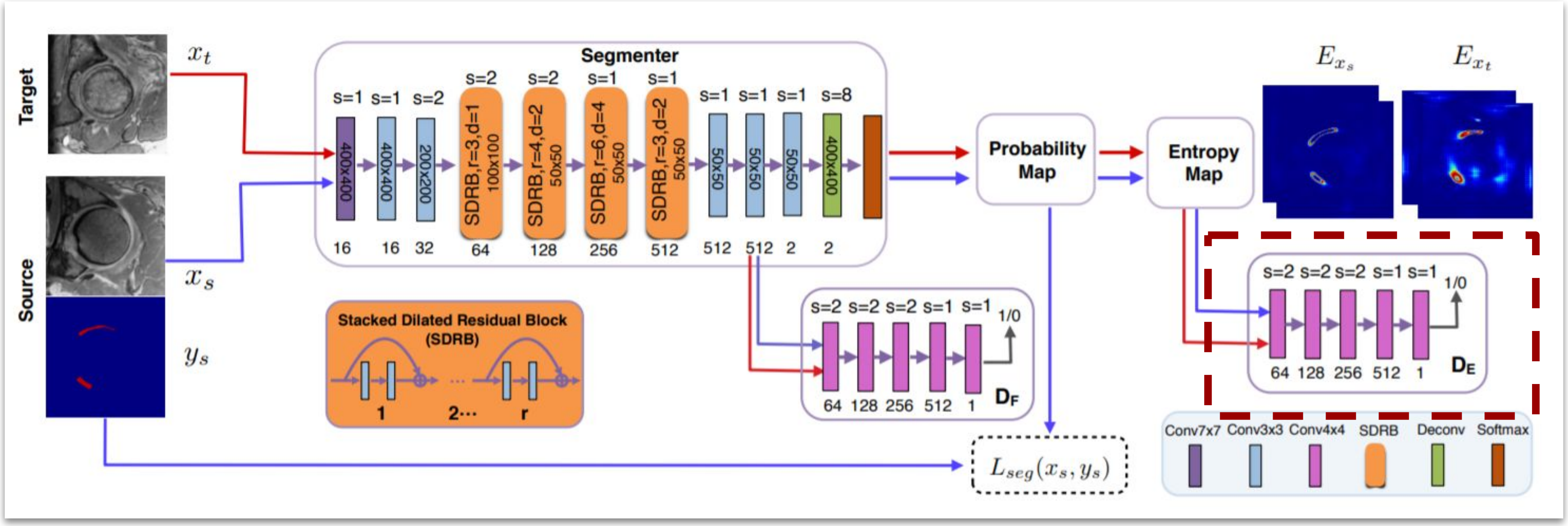
- Segmentor
  - Stacked Dilated Residual Blocks.
  - Segmentation of source and target images.
  - Generates feature maps and entropy maps for both source and target scans.

Overview of Reference Paper



- Feature Map Discriminator
  - Formed of stacked convolution layers.
  - Performs discrimination between features from source and target scans.
  - Used for adversarially training the segmenter to align features.

Overview of Reference Paper



- Entropy Map Discriminator
  - Formed of stacked convolution layers.
  - Performs discrimination between entropy maps from source and target scans.
  - Used for adversarially training the segmenter to align entropy maps.

Overview of Reference Paper

- Losses for training
  a. Segmentation loss :
     - Dice-BCE loss supervised on source ground truth.
  b. Discriminator loss:
     - BCE loss applied on source and target features and entropy maps for discrimination.
  c. Adversarial Loss:
     - Same as Discriminator Loss. However, target domain labels are flipped to “fool” the discriminators.
- Also, Surface Dice used for evaluation

Overview of Reference Paper

- Optimization strategy
  a. Segmentation loss :
     - Updates the segmentor network parameters.
  b. Discriminator loss:
     - Updates the discriminator parameters.
  c. Adversarial Loss:
     - Updates only segmentor network parameters.

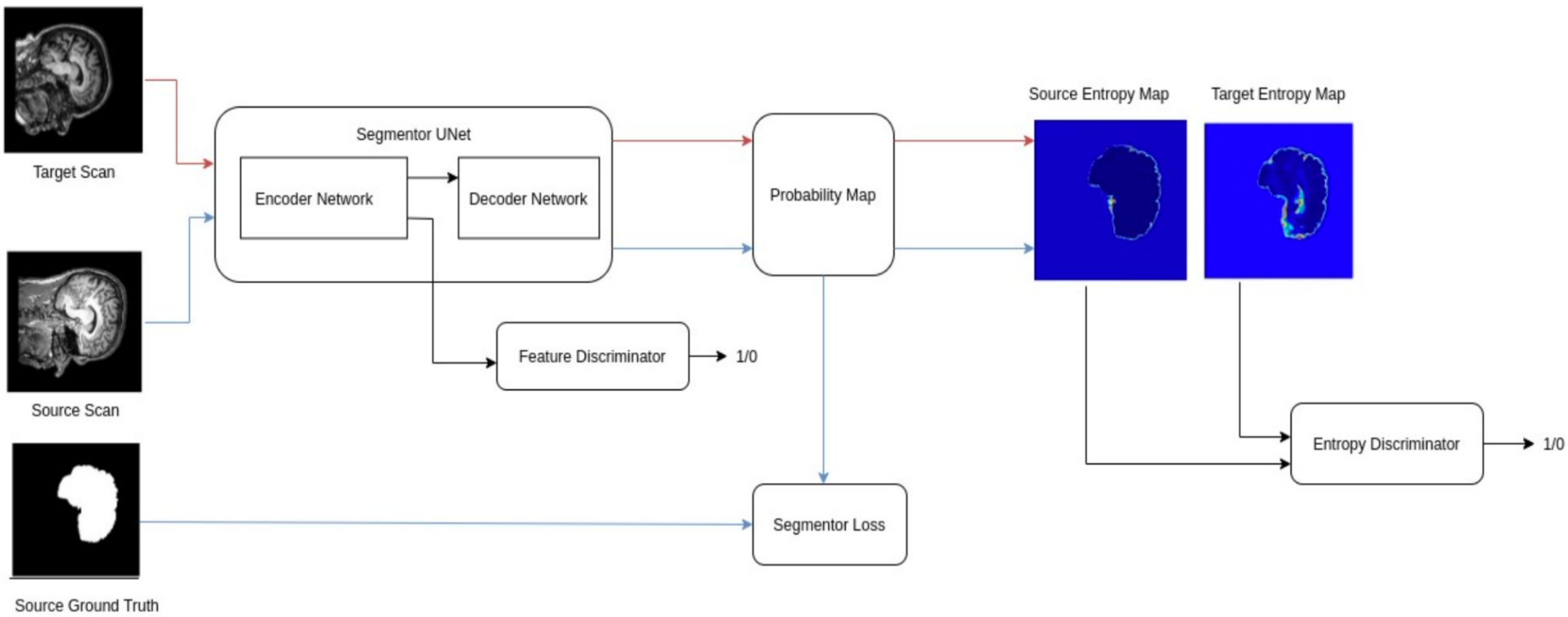
Overview of Reference Paper

- Method Summary:
  a. Segmentor trained in a supervised manner on source domain as well as in an unsupervised manner with adversarial loss on target domain (labels flipped).
  b. Both Feature and Entropy Discriminator are trained in a supervised manner on both domains.

Overview of our Method

- Problem Definition :
  - Given input :
    - Labelled source scans (xs,ys) and unlabelled target scans xt.
  - Expected output :
    - Predicted segmentation mask for target scans.
  - Labels for Discriminators:
    - Source : 1, Target : 0

Architecture

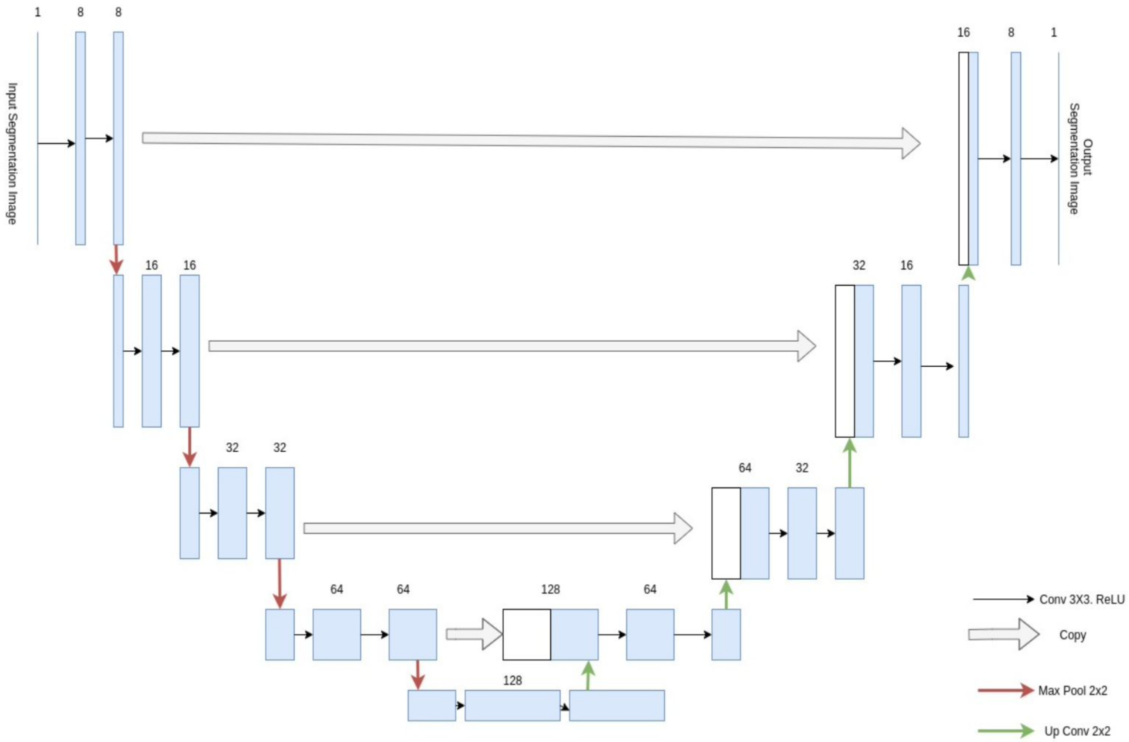


- The architecture consists of :
  a. Segmentor
  b. Feature Map Discriminator
  c. Entropy Map Discriminator

Segmentor

- U-Net architecture used as backbone for learning image segmentation. (unlike the reference)
- Architecture same as used by Shirokikh et al. [2]
- Trained with supervised loss on source and adversarial loss on target.
- Goal: Generate similar feature maps for both source and target.

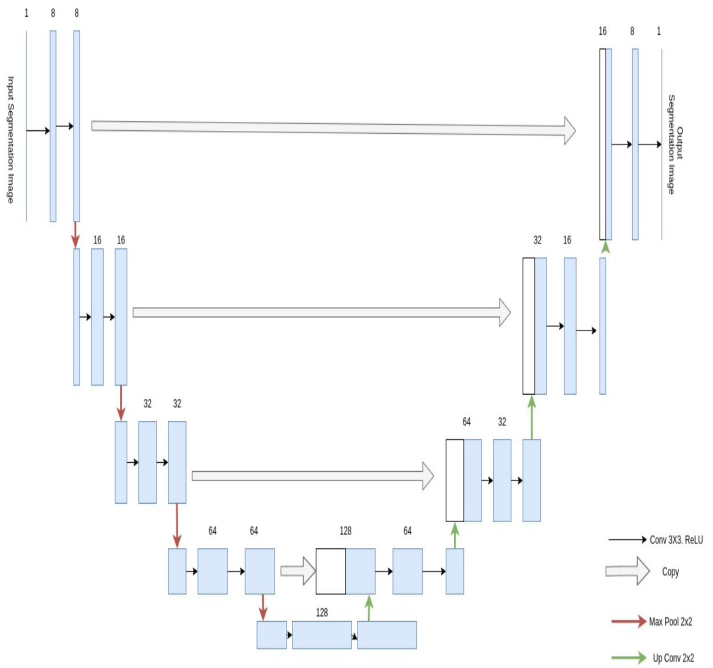
UNet Architecture



- Consists of two halves :
  - Encoder Network : Increases channels, reduces feature map size.
  - Decoder Network : Decreases channels, increases feature map size.

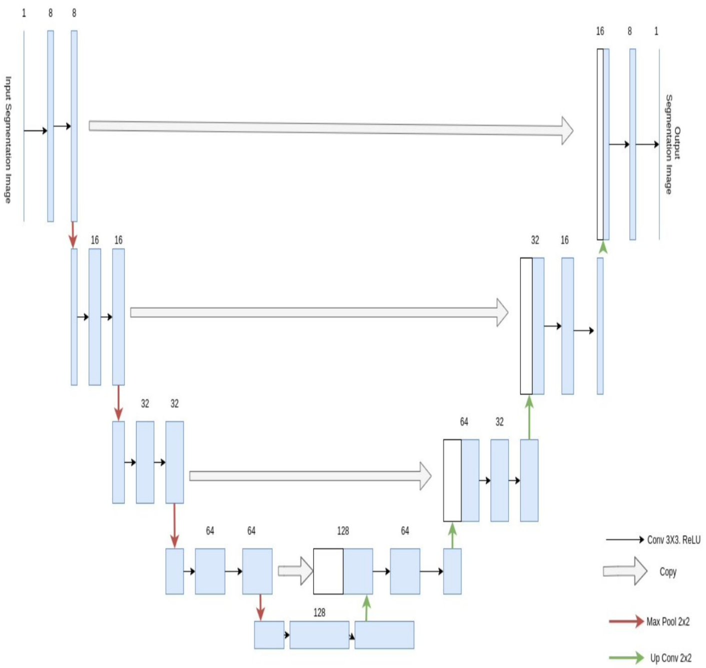
Encoder Network

- **Encoder Network :**
  - Takes segmentation image as input.
  - Each block consists of:
    - 3X3 kernels with ReLU.
    - 3 Resnet
  - Channels doubled with every block.
  - 3 such blocks in total
  - Channels become 8 times.
  - Each block has dropout layers.



Decoder Network

- **Decoder Network :**
  - Number of channels are halved at each block.
  - Transposed Convolution to increase feature map sizes.
  - Outputs a single channel with height and width similar to original input.



Training the Segmentor

- The weights of segmentor are changed by combination of Binary-cross entropy loss and the dice coefficient loss :

$$L = -\sum y_s^{(h,w,c)} . log(p_s^{(h,w,c)}) -$$

$$\lambda \sum \frac{2\hat{y_s}^{(h,w,c)} . 2y_s^{(h,w,c)}}{\hat{y_s}^{(h,w,c)} . 2y_s^{(h,w,c)} + \hat{y_s}^{(h,w,c)} . 2y_s^{(h,w,c)}}$$

Feature Map Discriminator

- Learns to discriminate between feature maps between source and target domains.
- Takes features from first half of UNet as input.
- Binary Cross Entropy Loss Used.

$$L_{D_F} = \frac{1}{|X_s|}\sum_{X_s} L_D(S_F(x_s), 1) + \frac{1}{|X_t|}\sum_{X_t} L_D(S_F(x_t), 0)$$

Feature Disc. Architecture

- Consists of 3 convolution layers followed by ReLU.
- Also consists of 64 dimensional fully connected layer mapped to a single neuron, followed by sigmoid.
- Ideally, the network should predict 1 for source and 0 for target.

Entropy Map Discriminator

- Learns to discriminate between entropy maps between source and target domains.
- Takes entropy maps from UNet as input.

Entropy Discriminator Architecture

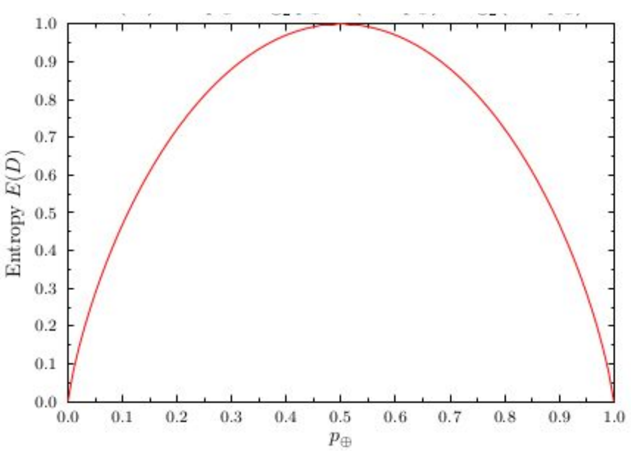
- Consists of 5 convolution layers followed by ReLU.
- Also consists of two fully connected layers of size 512 and 64 respectively.
- Finally, it maps to a single neuron and sigmoid giving output [0,1]
- Ideally, the network should predict 1 for source and 0 for target.

Entropy

- Entropy is calculated from the probability map from the U-Net decoder.

$$E_x^{(h,w,c)} = -p_x^{(h,w,c)}.log(p_x^{(h,w,c)})$$

- The entropy is min when probability is 0 or 1 and max when it is 0.5.



- The model gives low entropy for source and high for target.

$$L_{D_E} = \frac{1}{|X_s|}\sum_{X_s} L_D(E_{x_s}, 1) + \frac{1}{|X_t|}\sum_{X_t} L_D(E_{x_t}), 0)$$

Adversarial Loss

- Adversarial loss applied on output of feature map discriminator and entropy map discriminator.
- The labels of target and source are flipped.
- Propels feature alignments between source and target.

Adversarial Loss for Feature Map Disc.

- Feature discriminator takes target scan features as input will be fooled to produce output 1 (instead of 0).
- The adversarial loss from feature map discriminator only changes the segmenter encoder weights.
- Encoder forced to generate similar features for source and target.
- This hence, causes domain adaptation.

Adversarial Loss for Entropy Map Disc.

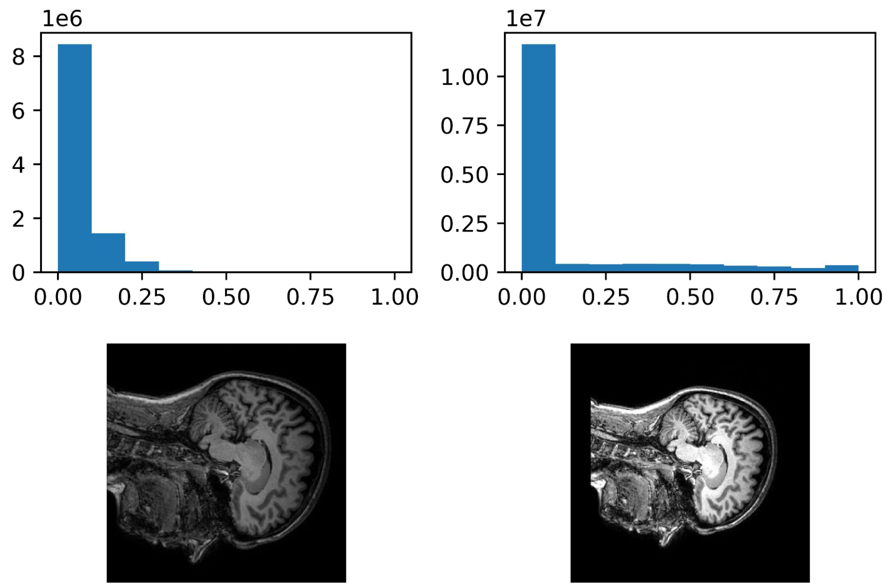
- Feature discriminator takes target scan features as input will be fooled to produce output 1 (instead of 0).
- The adversarial loss from entropy map discriminator only changes the segmenter weights.
- The adversarial loss from entropy map discriminator forces the segmentor to produce source-like low entropy outputs.
- This hence, causes domain adaptation.

Dataset

- Calgary-Campinas (CC-359) Public Brain MRI Dataset [4].
- Consists of 6 domains:
  - Different Vendors (GE, Philips, Siemens)
  - Different Magnetic field strengths (1.5T and 3T)
- Different domains have different image sizes.
- Different intensity ranges within and across domains.
- We arbitrarily chose GE-3 as our source domain. Also, after preprocessing, the data was split into 70%-20%-10% (train-val-test) for all domains.

Dataset Preprocessing

- Steps performed per scan volume:
  1. Clipping intensity values to range between 1st and 99th percentile
  2. Min-max scaling
  3. Zero-padding all scans to same size (288x288)
- Step 1,2 same as Shirokikh et. al. [2].

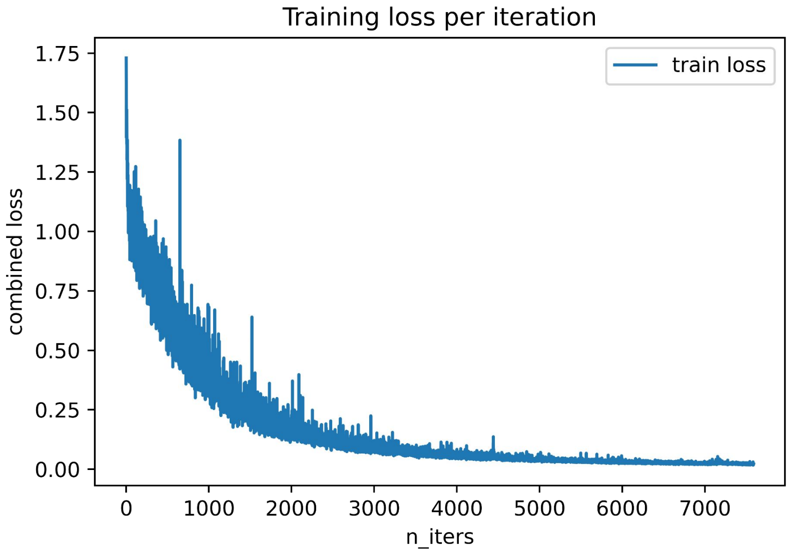


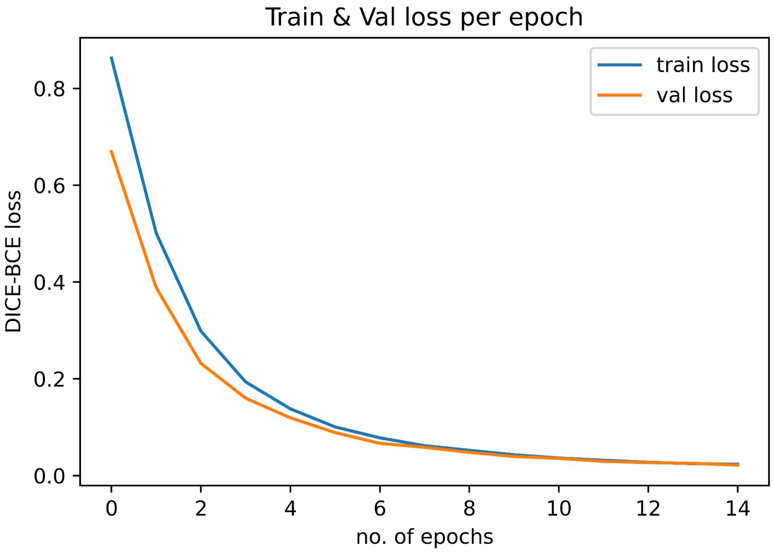
Overview of Experiments

- **Stage I:** Baselines
- **Stage II:** Implement Original Paper
- **Stage III:** Modified Adv. Loss
- **Stage IV:** Training from scratch
- **Stage V:** Direct Entropy Minimization

Stage-I: Segmenter training

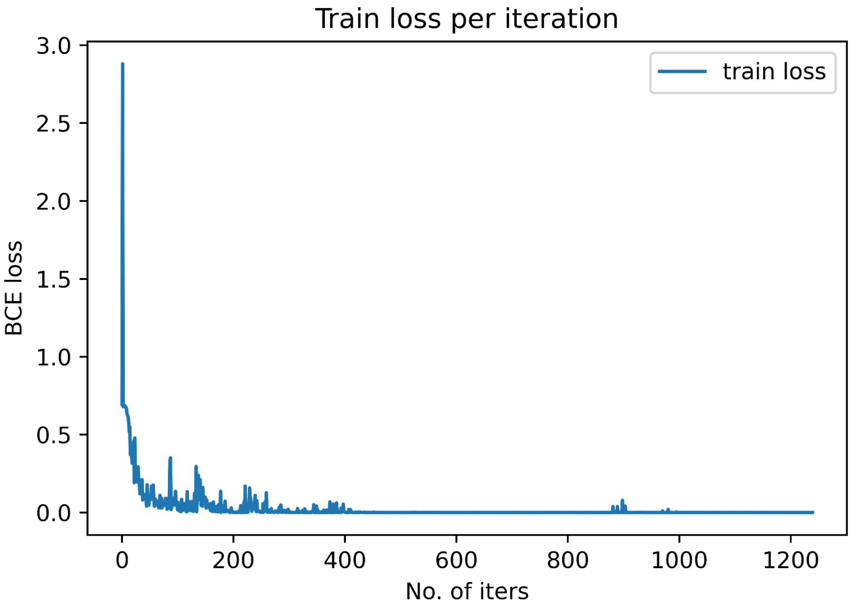
- 15 epochs
- Adam Optimizer (Default)
- Batch Size: 16
- Dice-BCE Loss

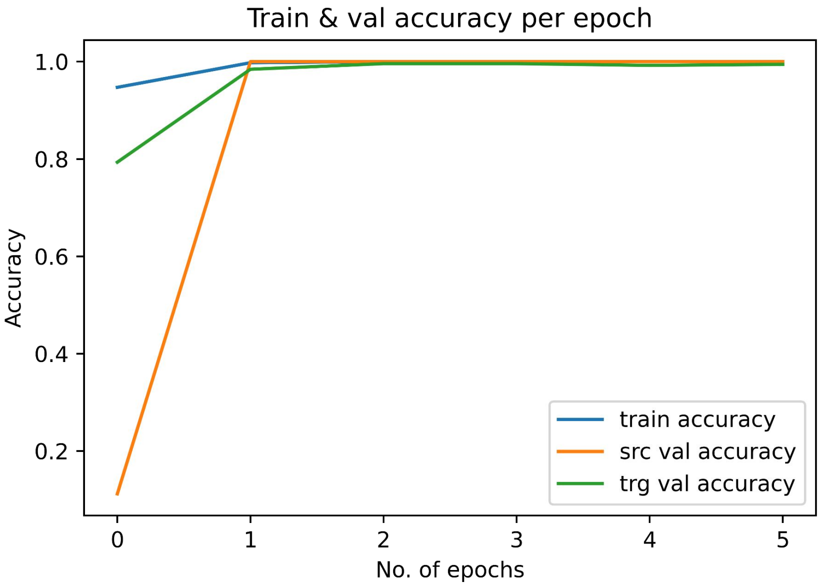




Stage I: Feature Disc.

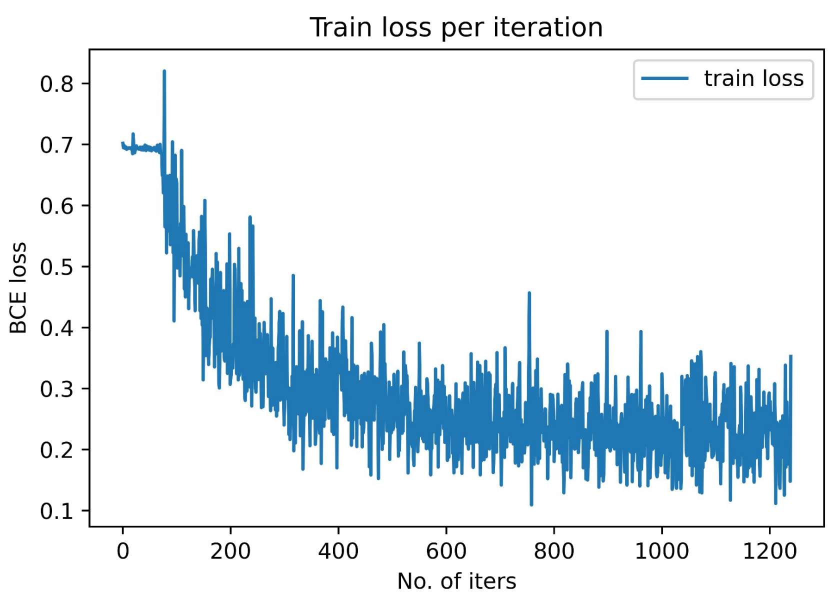
- No. of epochs: 5
- Adam Optimizer (Default)
- Batch size: 16
- BCE Loss

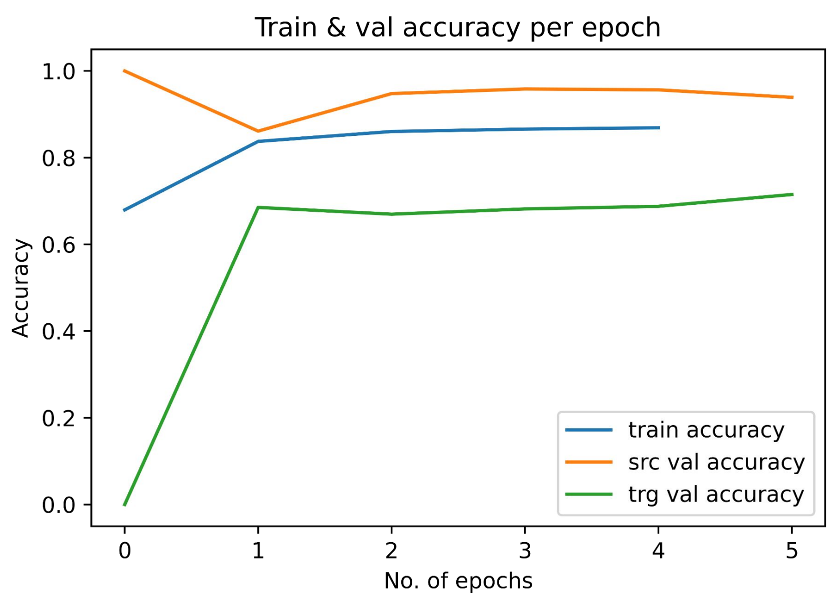




Stage I: Entropy Disc.

- No. of epochs: 5
- Adam Optimizer (Default)
- Batch size: 16
- BCE Loss





Stage-II: Original Paper

Mentioned training settings:

- Pre-trained Segmenter
- Untrained Discriminators
- Only details for training segmenter given:

> **Implementation details.** The proposed method was implemented in Python using the Pytorch framework on a desktop with a 3.6 GHz Intel(R) i7 CPU and a GTX 1080 Ti graphics card with 11 GB GPU memory. We trained our network from scratch, and the parameters were updated by the stochastic gradient descent(SGD) algorithm (momentum=0.9, weight decay=0.005).The input image size was 400×400 and the batch size was 4. We trained the network for a total of 30 epochs. The initial learning rate was 1×10^{-3} and halved by every 5 epochs.

- Adversarial Weights and other hyperparameters not specified

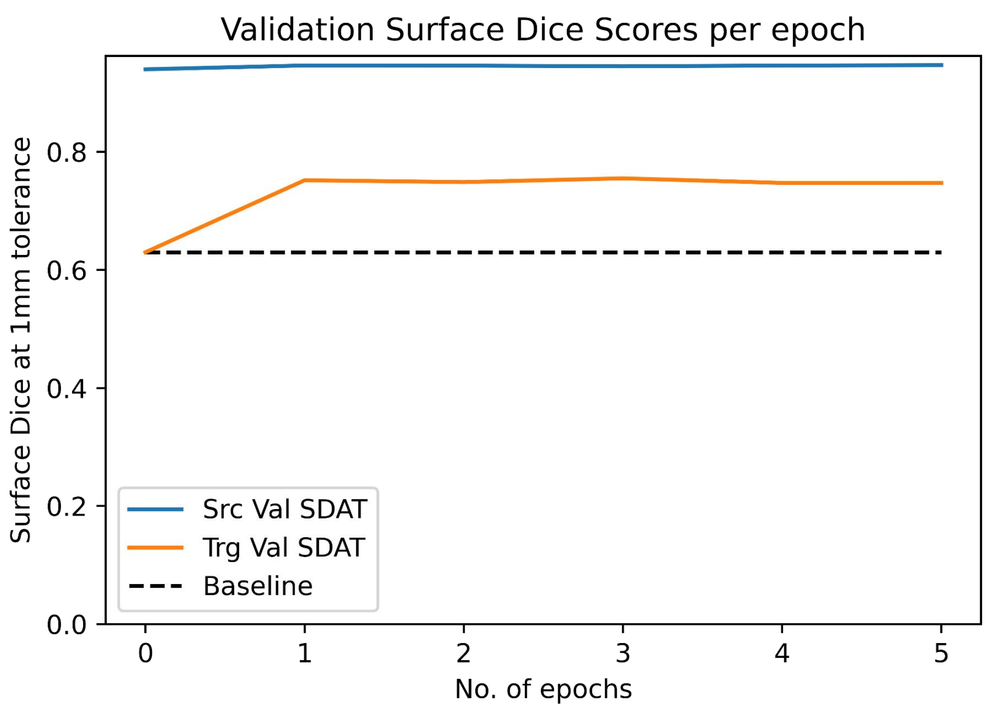
Stage-II

What we tried:

- Different learning rates in the range 1e-3 to 1e-6
- Different weights for adversarial loss components in the range 1e-5 to 1
- Different optimizers: SGD, AdamW, Adam
- Using pre-trained discriminators
- Alternatively training the components (like GANs)

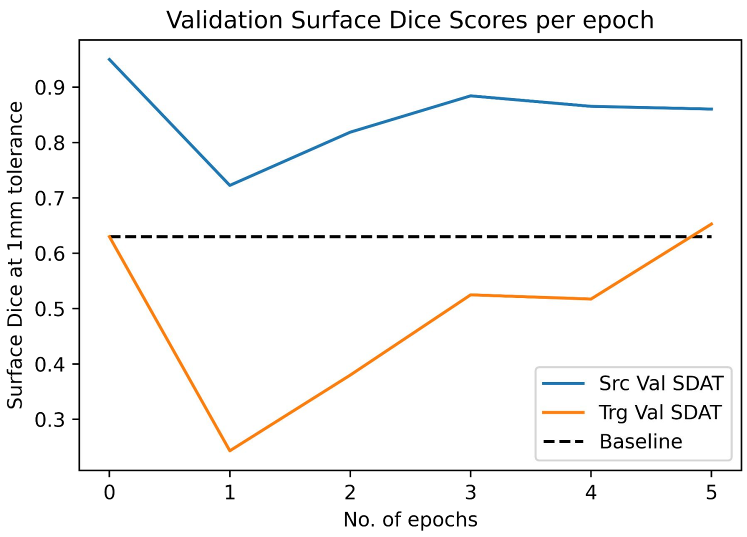
Stage II: Insights

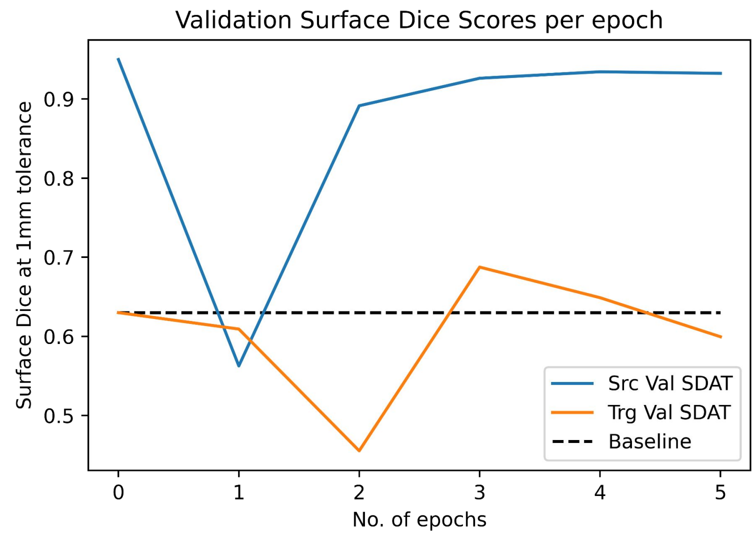
- Training with zero adversarial loss also shows consistent increase in surface dice scores on target.
- Results with setting all loss components to 0 shown below. (Only updates in the model parameters were Batch Norm statistics like running mean, variance etc.)



Stage II: Insights

- Initial focus was to fool both discriminators.
- Higher weight values for adversarial loss prevents discriminator from learning.
- Discriminator biased towards single domain.
- Below are examples when adversarial loss for feat. disc. was weighted 0.1 and 0.2 respectively:

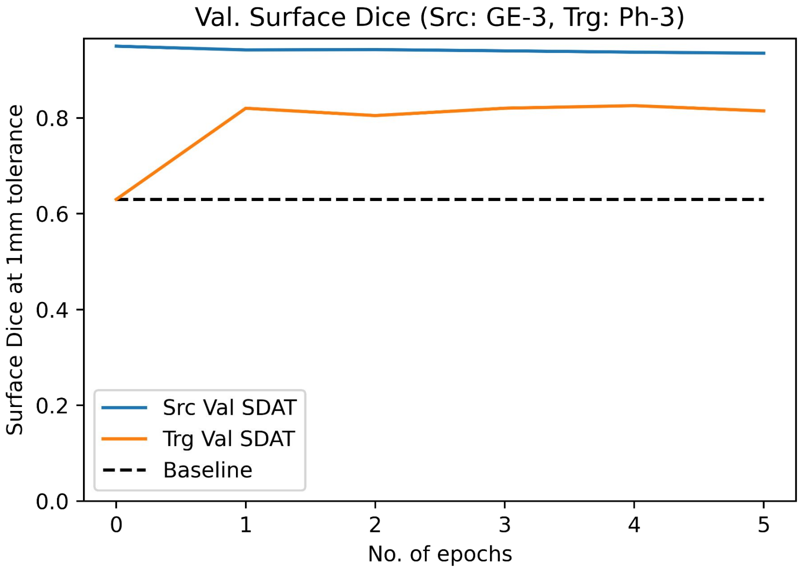


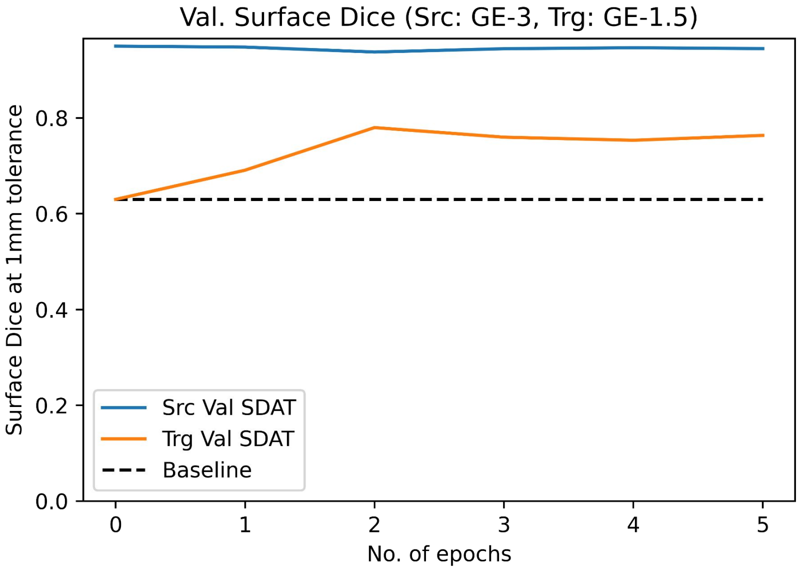


Stage II

What worked:

- Learning Rate 1e-4 for all components
- Adversarial weights (Lambdas): 1e-3 (For low domain shift: 1e-4)
- No. of epochs: 5 (For low domain shift: 3)
- Learning Rate Scheduler with 0.75 Decay and step size 1





Stage III: Modified Adv. Loss

- Objective of adversarial loss:
  - produce domain invariant features
  - produce low entropy source-like outputs

- Original Adv. Loss for feature disc. is calculated only for target domain:

$$L_S = \frac{1}{|X_s|} \sum_{x_s \in X_s} L_{seg}(x_s, y_s) + \frac{1}{|X_t|} \sum_{\underline{x_t \in X_t}} (\lambda_2 L_D(E_{x_t}, 1) + \lambda_3 \underline{L_D(S_F(x_t), 1)}) \tag{5}$$

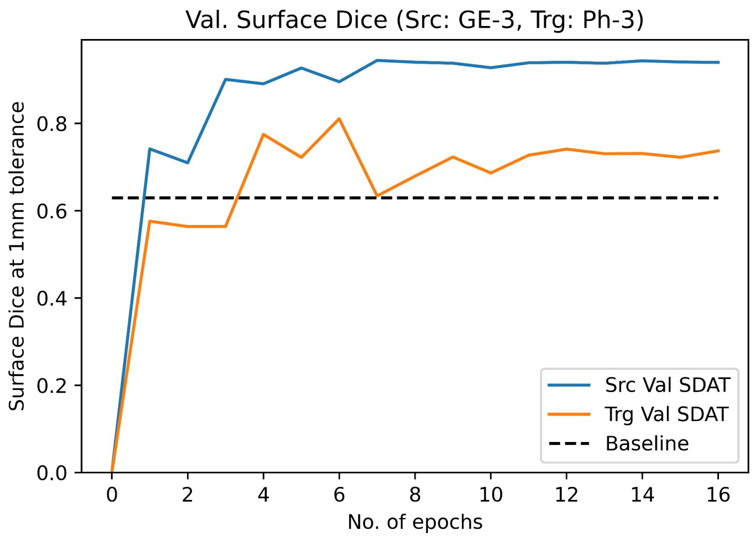
- Instead we calculate Adv. Loss on feature disc. for both domains.
- Adv. Loss on entropy disc remains same

Stage IV: Train from scratch

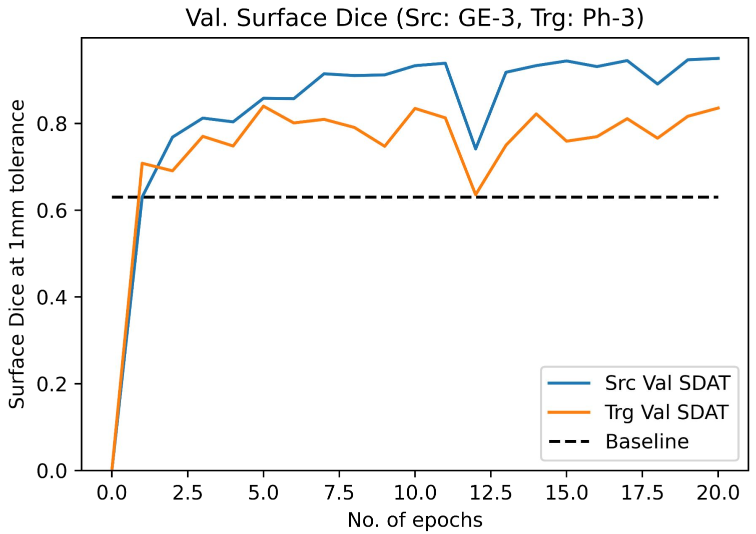
- Original paper suggests fine-tuning a pre-trained segmenter
- Fine-tuning can sometimes reduce performance on source domain
- Our hypothesis was: jointly training the entire network from scratch could be a robust way to produce domain invariant feature
- We try this using the approach in Stage III

Stage IV:

- Only Feature Discriminator: LR scheduler with 0.5 decay at every step. LR scheduler steps only after min. src. validation surface dice score of 0.85 and entropy less than 0.15



- Both Discriminators: LR scheduler that decays the learning rate by a factor of 0.5 at each step (with steps at epochs 15 and 18)



Stage V: Entropy Minimization

- Original paper suggests indirect entropy minimization using adversarial loss
- Objective is to produce low entropy source-like outputs
- As proposed by Vu et. al [3], we examine the effect of further adding direct entropy minimization

$$L_{ent_x} = \sum_{h,w} E_x^{(h,w)}$$

$$E_x^{(h,w,c)} = -p_x^{(h,w,c)} . log(p_x^{(h,w,c)})$$

- Try this using approach in Stage II, III (Not IV)

Overview of Experiments

Stages	**Summary**
Stage-I	Baseline
Stage-II	Original Model
Stage-III	Same as Stage II, except Modified Adversarial Loss for Feat. Disc.
Stage-IV	Same as Stage III, except trained from scratch
Stage-V	Add Direct Entropy Minimization. Produce results for approach in Stage II and Stage III

Overview of Experiments

Approach	Feat. Disc. Only	Feat. & Ent. Disc.	Both Disc. & Direct Ent. Min.
Stage-I (Baseline)	-	-	-
Stage-II (Original Model)	Yes	Yes	Yes
Stage-III (Modified Loss)	Yes	Yes	Yes
Stage-IV (From Scratch)	Yes	Yes	-
Stage-V (Direct Ent. Min.)	Yes	Yes	Yes

Results: Baselines

Domain	DICE	Surface Dice at 1mm tol.
Train	0.99	0.96
Val.	0.99	0.95
Test	0.99	0.95

Results on Source Domain (GE 3)

Domain	DICE	Surface Dice at 1mm tol.
GE 3 (SRC)	**0.99**	**0.95**
GE 1.5	0.86	0.51
Philips 3	0.87	0.63
Philips 1.5	0.97	0.83
Siemens 3	0.98	0.93
Siemens 1.5	0.95	0.80

Performance on target domain (Without UDA) using random sample of 20 scans per domain

Results: Notations

Notation	Description
FeatDisc	Only Feature Discriminator used for UDA
BothDisc	Both Discriminators used for UDA
Combined	BothDisc and Direct Entropy Minimization
FS	Network trained jointly 'From Scratch'
FT	Network trained using only 'Fine-Tuning'
(OG)	If specified, the loss function is used as specified in original paper. Else, our modified loss.

Results

Approach	**DICE**	**Surface Dice**
No domain adaptation	0.80	0.49
FeatDiscFS	0.90	0.61
BothDiscFS	0.95	0.73
FeatDiscFT(OG)	0.91	0.63
BothDiscFT(OG)	0.92	0.64
CombinedFT(OG)	0.94	0.70
FeatDiscFT	0.92	0.64
BothDiscFT	0.94	0.71
CombinedFT	**0.96**	**0.75**

Results using source domain GE-3 and target domain Philips-3. Results reported on unseen test set of 7 volumes (10% of available data).

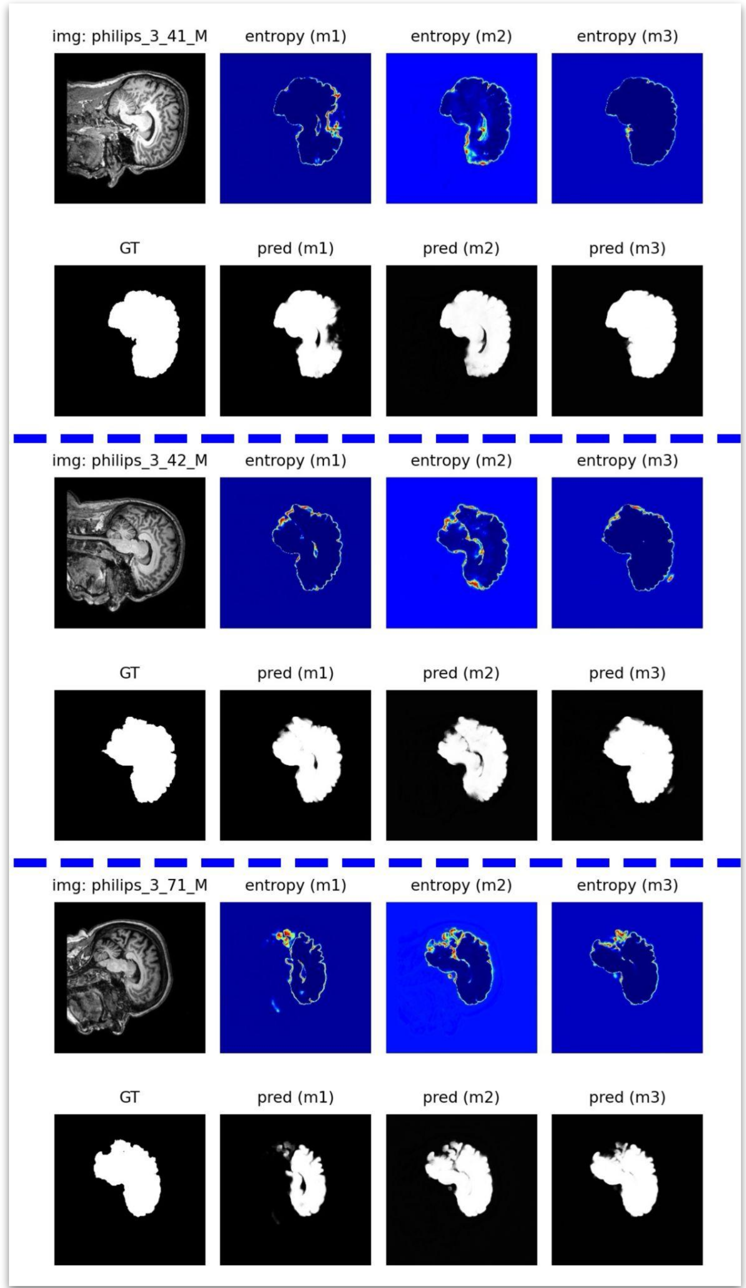
Results

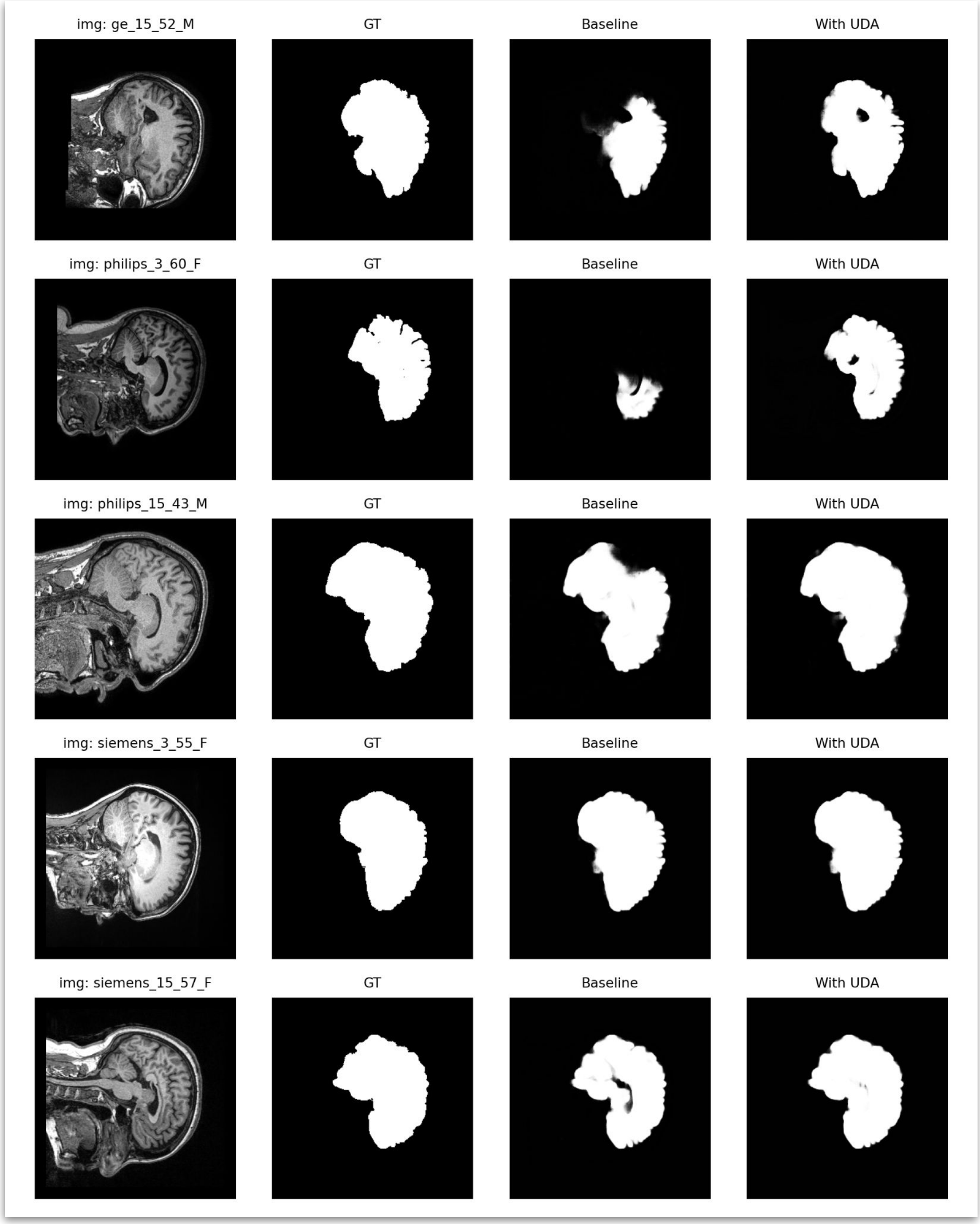
Approach	**Baseline**	**FeatDiscFT**	**BothDiscFT**	**CombinedFT**
GE 1.5	0.624	0.792	0.793	**0.795**
Philips 1.5	0.855	**0.897**	0.891	0.844
Philips 3	0.496	0.645	0.71	**0.759**
Siemens 1.5	0.808	0.818	0.837	**0.862**
Siemens 3	**0.940**	0.937	0.935	0.937

Surface Dice Score using source domain GE-3 and all other target domains. Results reported on unseen test sets (i.e. 10% of available data for each domain).

Results

- Plots for 3 test scans from target domain Philips-3. We see differences in outputs and entropy maps for three approaches from Stage IV (training from scratch):
  1. 1st column: Without adaptation (m1)
  2. 2nd column: With FeatDiscFS (m2)
  3. 3rd column: With BothDiscFS (m3)





Plots comparing UDA results for best models (metrics reported on slide 51) on the test set of all domains.

Conclusion

- Unsupervised Domain Adaptation can successfully reduce the effect of domain-shift.
- Entropy-based methods (both direct and indirect entropy minimization) can improve domain adaptation.
- Adversarial feature discriminator loss on both domains can perform better than original approach
- Fine-tuning methods can perform better than jointly training the network from scratch at a fraction of the training cost

Future Work

- Obtain confidence scores for results.
- Further comparisons of all approaches with different source domains.
- Observe domain shift with regularized and transformer based models.
- GANs or histogram matching based style transfer from target to source.
- Self-supervised learning by augmenting target images and source images.

References

[1] Guodong Zeng (2020) 'Entropy Guided Unsupervised Domain Adaptation for Cross-Center Hip Cartilage Segmentation from MRI'. *MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I.* Available at: https://dl.acm.org/doi/10.1007/978-3-030-59710-8_44 (Accessed: 17 March 2022).

[2] Boris Shirokikh (2020) 'First U-Net Layers Contain More Domain Specific Information Than The Last Ones'. *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings*. Available at: https://arxiv.org/abs/2008.07357 (Accessed: 17 March 2022)

[3] Tuan-Hung Vu (2019) 'ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation'. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.* Available at: https://arxiv.org/abs/1811.12833 (Accessed: 17 March 2022)

[4] Roberto Souza (2018) 'An open, multi-vendor, multi-field-strength brain MR dataset and analysis of publicly available skull stripping methods agreement' *NeuroImage, Volume 170, 2018.* Available at: https://pubmed.ncbi.nlm.nih.gov/28807870/ (Accessed: 17 March 2022)