Defaced MRI Detection & Generation

By Madhur Tandon, Mudit Garg & Chaitanya Agrawal

Motivation & Problem Statement

Motivation

- Increase in Research Interests to apply ML/DL to neuroscience
- Huge amounts of publicly datasets available
- Anonymization necessary to protect identity of test subjects
- Also required by GDPR (General Data Protection Regulation)
- Amount of defacing required to guarantee anonymity?

Problem Statement

- Build a binary classifier that can detect if an MRI Scan is defaced or not.
  - Can serve as a lightweight deployable tool that researchers can use as a check!
- Build a generative model that can learn to deface as well as reface MRI Scans
  - Showcases that current methods of anonymization are partially reversible to some extent
  - May not provide adequate protection to one’s identity

Related Work

Related Work

- Preserving Privacy in Structural Neuroimages
  - https://link.springer.com/content/pdf/10.1007/978-3-642-22348-8_26.pdf
- PyDeface Library
  - https://pypi.org/project/pydeface/
- CNNs for MRI Scans
  - https://arxiv.org/pdf/1712.03747.pdf

Dataset & Evaluation

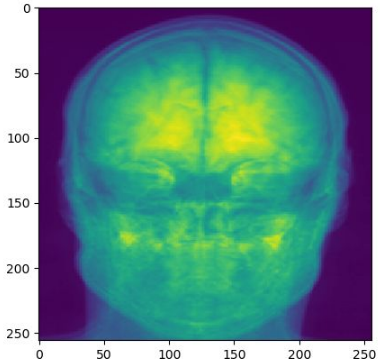
Dataset - I

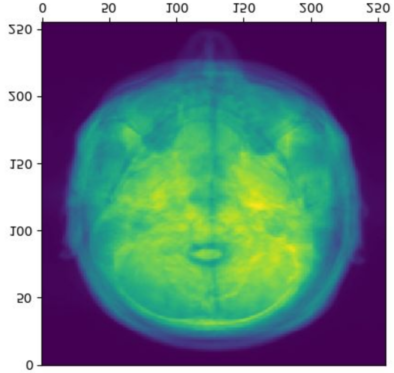
- IXI Dataset (http://brain-development.org/ixi-dataset/)
  - Around 600 MRI Scans from healthy & normal subjects
  - Used T1 images (out of T1, T2, PD-weighted) - [**non - anonymized**]
- Used Augmentation Techniques to increase size
  - Flips, Rotations, Gaussian Blur, Contrast Normalization, Zoom, etc.
- Corresponding Defaced Scan generated
  - With *pydeface* library!
  - Gives us 2 classes → Defaced and Non-Defaced
    - Required for Binary Classification

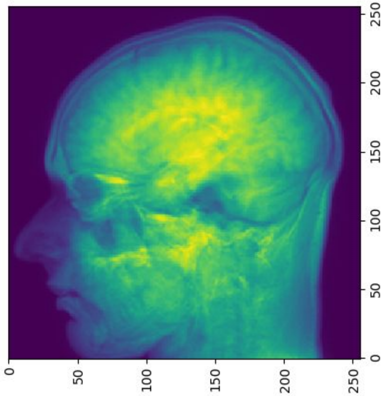
Dataset - II

- MRI Scans are 3D in nature
    - Chose 3 kinds of 2D images from one 3D image (one for each dimension)
        - To reduce complexity and computational overhead
- 2 pre-processing techniques
    - Mean along each dimension
    - Middle Slice of each dimension
- Total 6 = (2 pre-processing * 3 dimensions) 2D images per 3D image
- Each 2D image resized to 256x256 pixel size

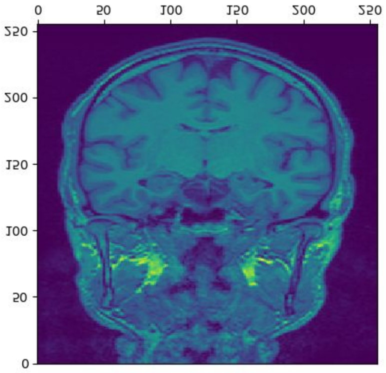
Dataset - III

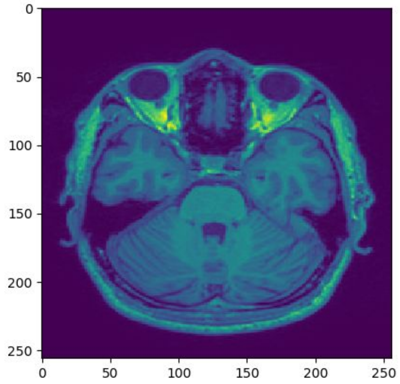


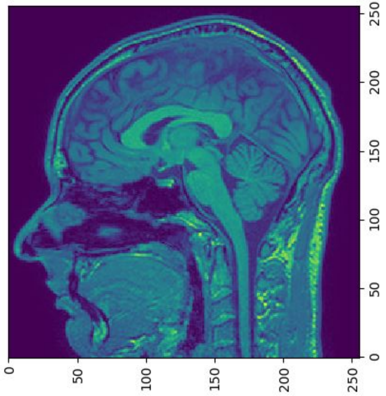




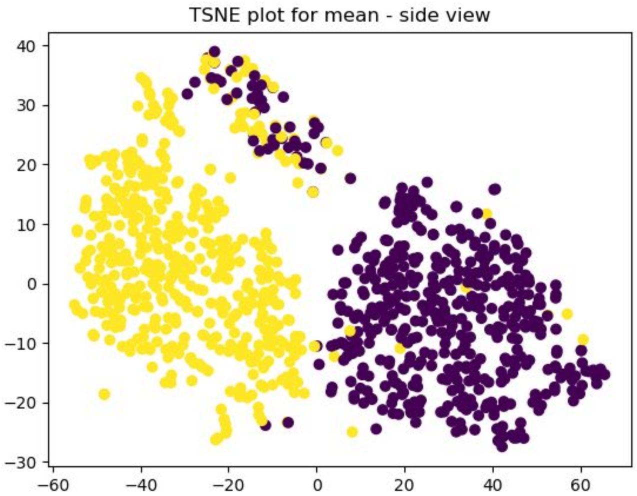
Dataset - IV



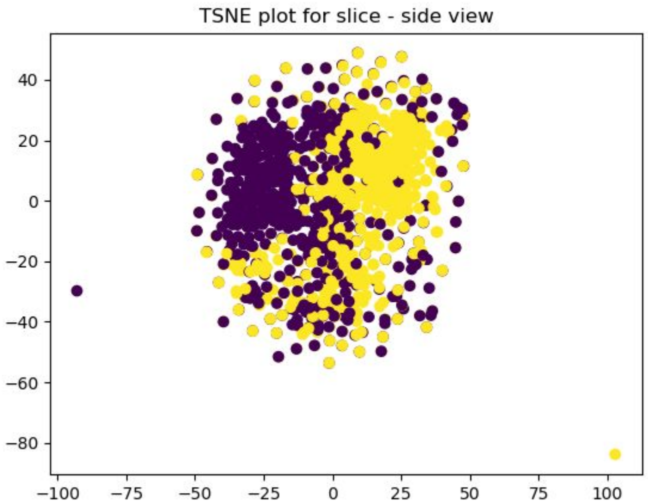




Dataset - V



Mean: Blurs out the intricate details of the underlying content within the MRI Image but preserves the actual shape and the structure of a person’s head, which is essential for detecting a potential defaced image.



Slice: gives us the underlying neural anatomy.

Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-Score

Classification Methods & Analysis

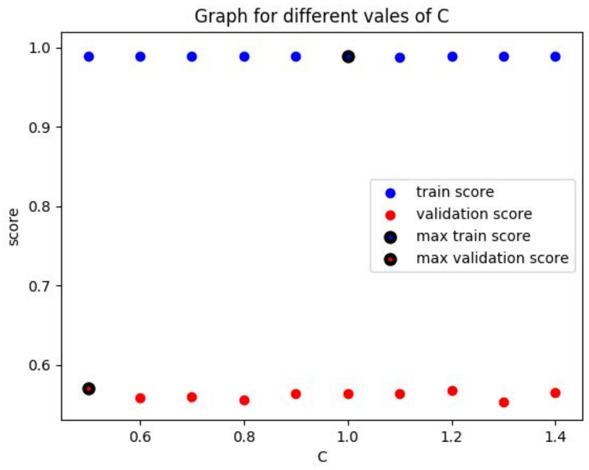
Baseline - Logistic Regression

Logistic Regression - I

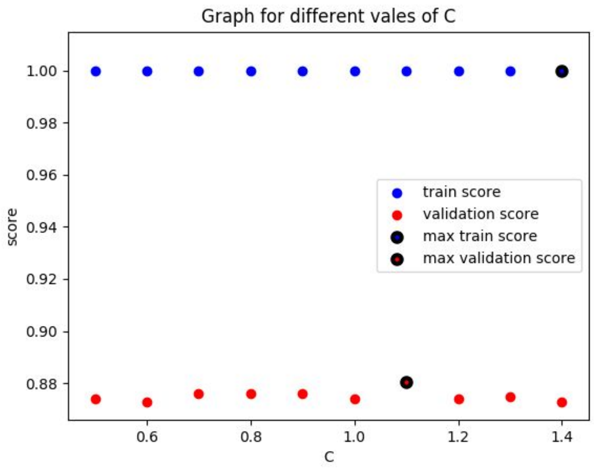
- 6 Logistic Regression Classifiers trained for 6 different 2D Images
- Voting done for final prediction

Logistic Regression - II

Validation Curve Slice,
Dimension = Top View

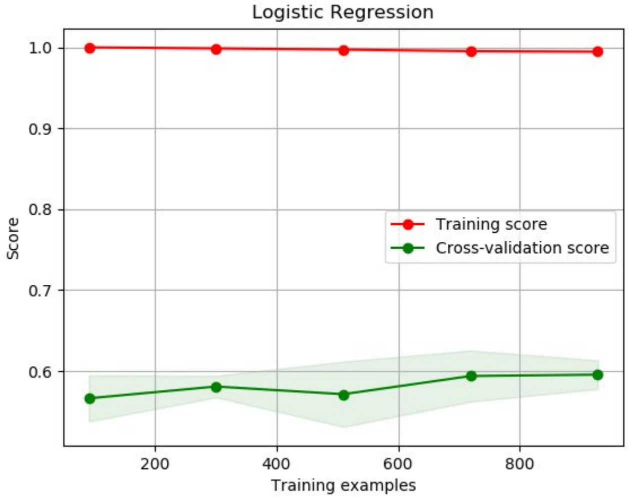


Validation Curve Mean,
Dimension = Top View

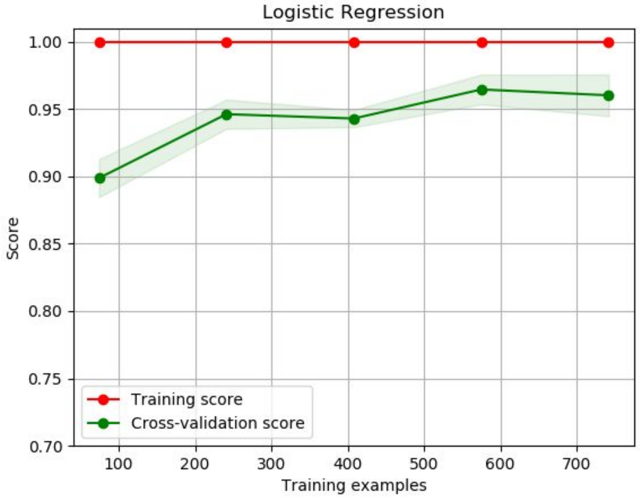


Logistic Regression - III

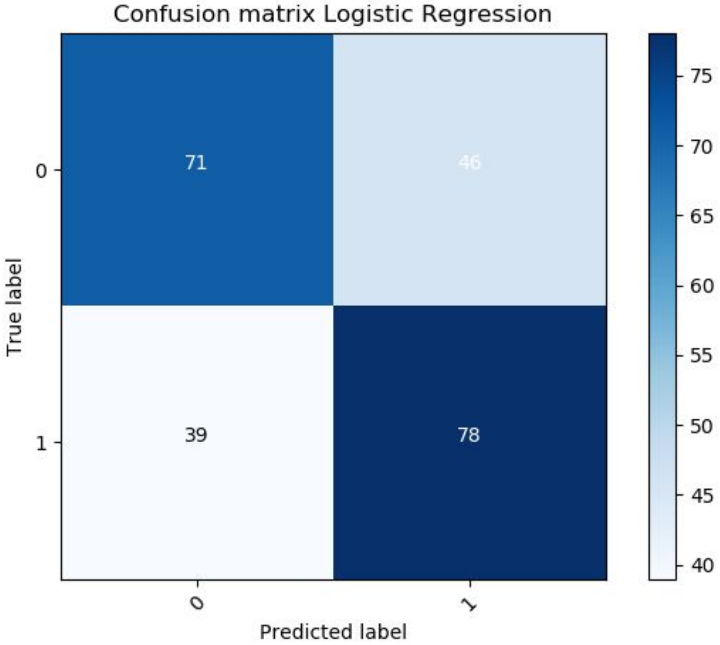
Learning Curve Slice,
Dimension = Top View



Learning Curve Mean,
Dimension = Top View



Logistic Regression - IV



Logistic Regression - Results/Analysis

- The logistic regression model did not perform well on the test set despite giving 1 accuracy on train set as evident from the graph.
- Therefore, the model results in overfitting.
- Hyperparameters like the value of regularization parameter were selected using grid search CV.

Advanced - Kernelized SVMs, Random Forests, CNN

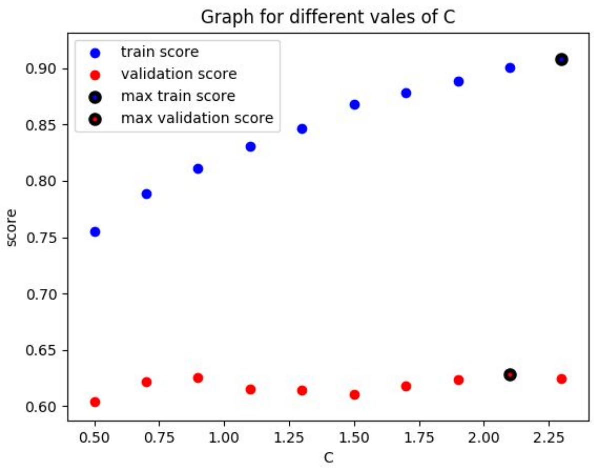
SVMs

SVM - I

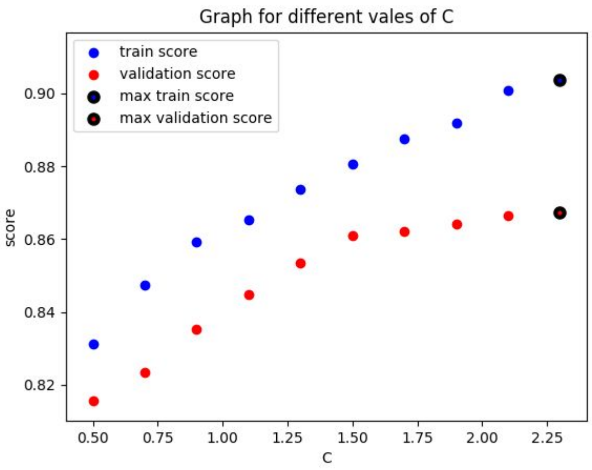
6 RBF kernelized SVMs trained for 6 different 2D Images

SVM - II

Validation Curve Slice,
Dimension = Top View

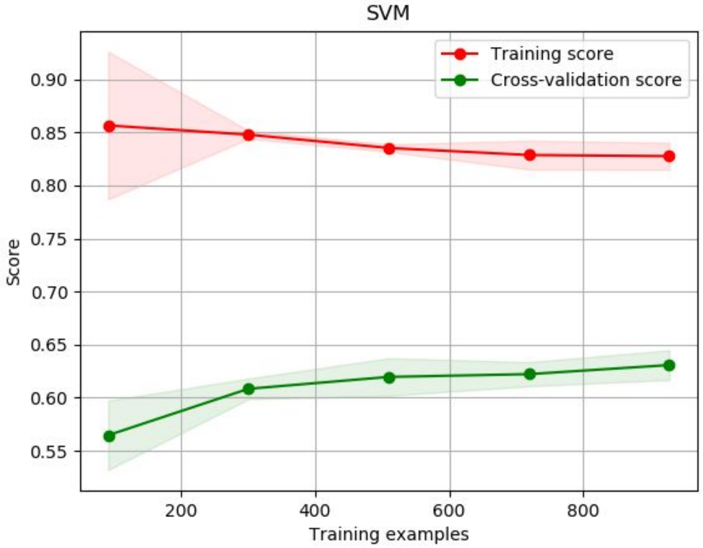


Validation Curve Mean,
Dimension = Top View

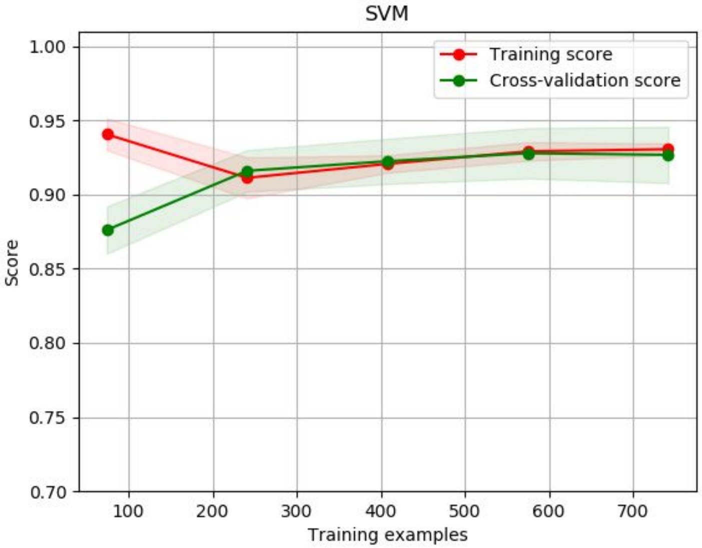


SVM - III

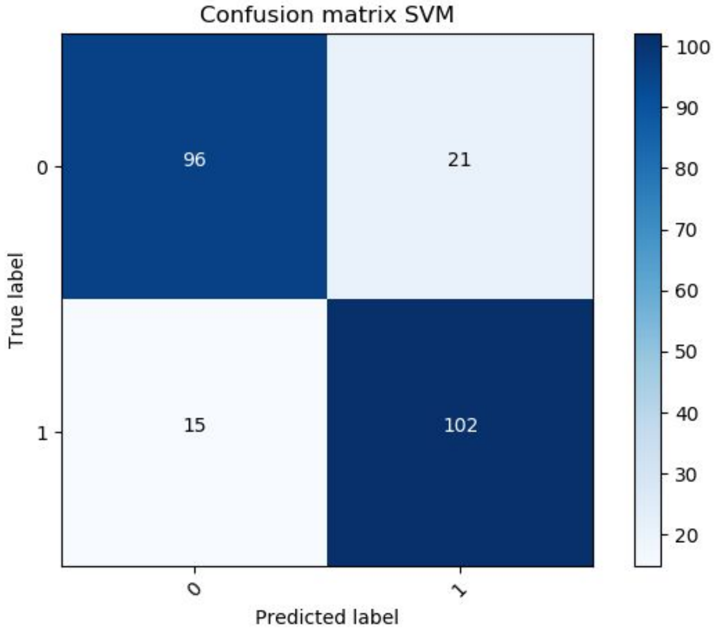
Learning Curve Slice,
Dimension = Top View



Learning Curve Mean,
Dimension = Top View



SVM - IV



Confusion matrix SVM
96
21
15
102
0
1
True label
0
1
Predicted label
100
90
80
70
60
50
40
30
20

SVM - Results/Analysis

- A kernel adds non-linearity to the decision surface by mapping each data sample to a higher dimension.
- Therefore, we expect SVMs to find more complicated decision surfaces than Logistic Regression.
- Since SVM can learn complex decision boundaries, there is a possibility of overfitting. However, both the train and test the accuracy of the SVM model are almost the same, so we are assured that overfitting has not occurred.
- The hyperparameters are selected by using grid search CV.

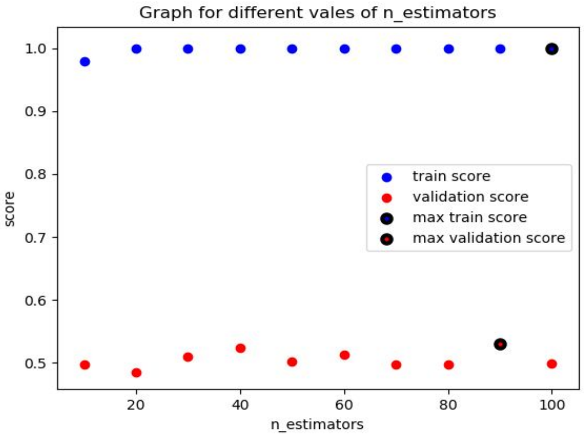
Random Forests

Random Forests - I

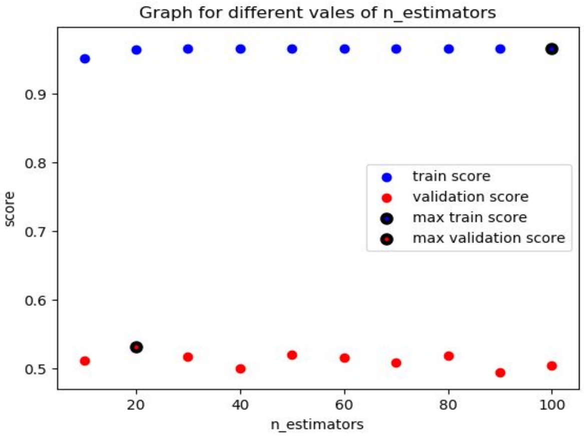
6 Random Forests Classifiers trained for 6 different 2D Images

Random Forests - II

Validation Curve Slice Dimension: Top View

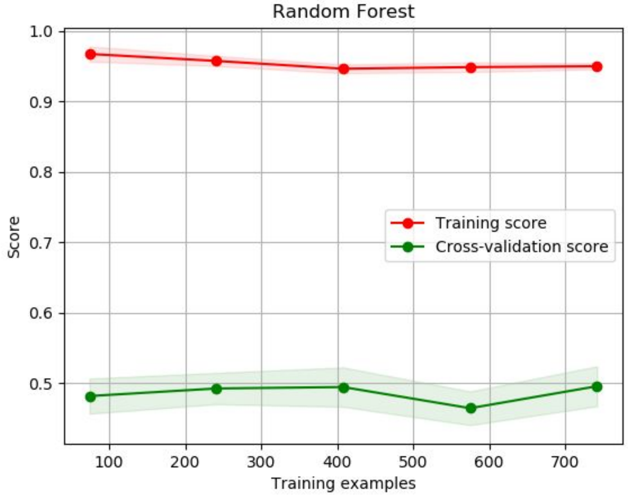


Validation Curve Mean Dimension: Top View

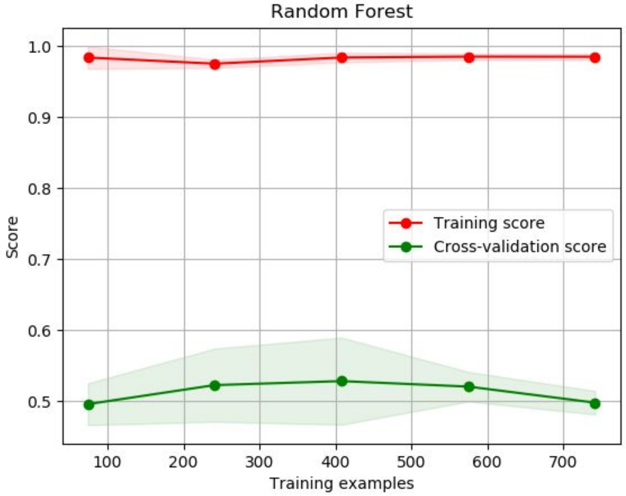


Random Forests - III

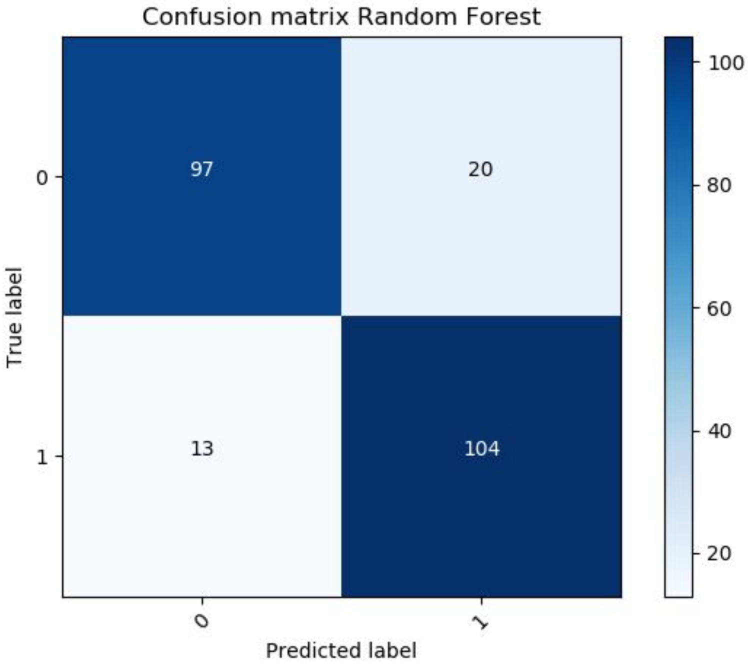
Learning Curve Slice
Top View



Learning Curve Mean
Top View



Random Forests - IV



Random Forests - Results/Analysis

- Random Forests performed best with side section means
- They had very bad results with other angles
  - Shows how it is transform invariant (Lots of augmentation on training/test data)
- Overfit a lot on almost all types of sections as expected with random forests
- We tried to remedy this with different parameters and grid search, but couldn't

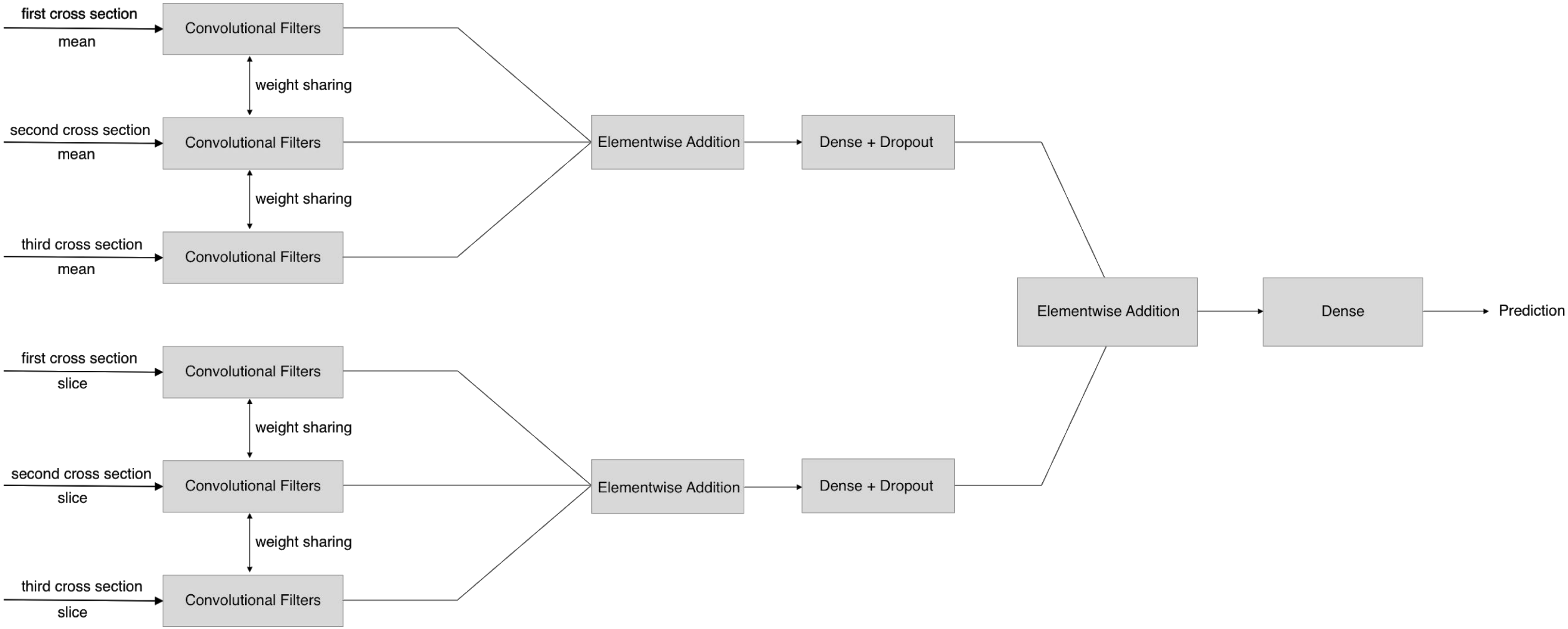
CNNs

CNNs - I

- 6 - Branch CNN with each branch taking in one of the 2D Scans as an input image

- Architecture trained on 32x32 pixel scans to reduce computational overhead and complexity of the model

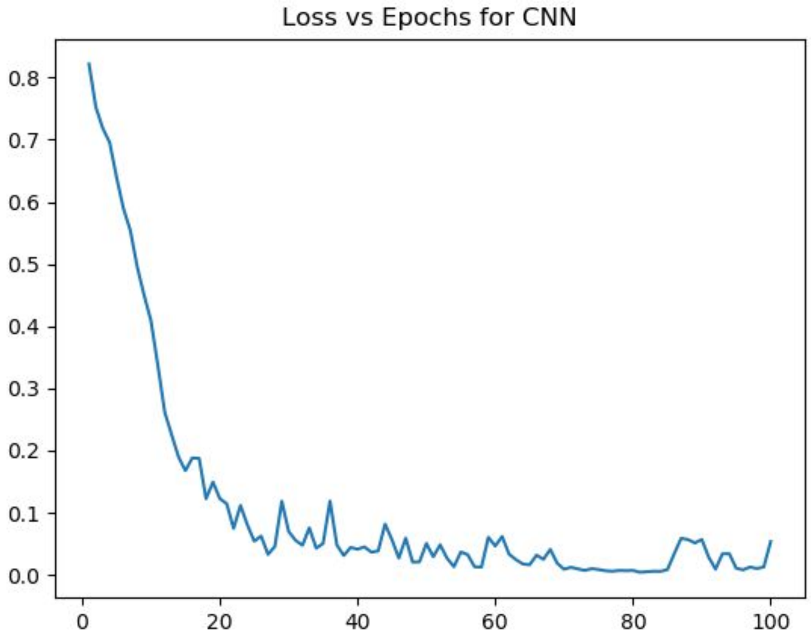
CNNs - II (Model Architecture)



CNNs - III (Model explained)

- 2 sets of 3 branches, each branch accepting different combinations of cross-sections and preprocessing techniques to account for all 2D images
- Input layer (of each branch) is of size 32x32x1
  - Size is downsampled due to computational overheads
- Each branch is fed into a set of 6 convolutional filters with increasing number of filters (8,16,32 with 3 as kernel size)
- These are then connected to dense fully connected

CNNs - IV



CNNs - Results/Analysis

- CNN uses all of the six 2D scans from the 3D volume
- Thus, accounts for both Mean and Slice
    - Mean
        - blurs out the intricate details of the underlying content within the MRI Image but preserves the actual shape and the structure of a person’s head, which is essential for detecting a potential defaced image.
    - Slice
        - gives us the underlying neural anatomy.

Generation of Defaced and Refaced MRI Images

Initial Study

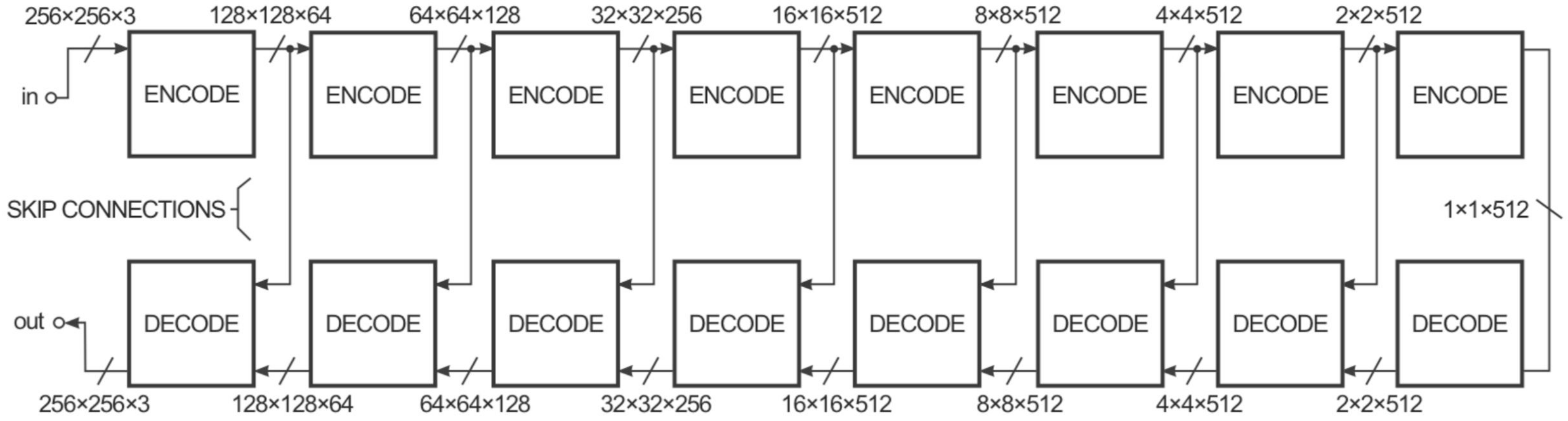
- This is an Image to Image Translation problem
  - Explored CycleGANs and Pix2Pix
- CycleGANs
  - Used for discrete unpaired collections of images
  - Can work on images which are not tightly correlated
  - 2 Generators and 2 Discriminators -- complex model!
- Pix2Pix
  - Uses Patches for believability scores
  - Works well for highly correlated images
  - More suited to our problem!

GANs - I

- Our Dataset has defaced image corresponding to each MRI scan.
- Thus, Pix2Pix is more suitable here
- Also, one generator and one discriminator make the model less complex
- Thus, chose Pix2Pix!

GANs - II

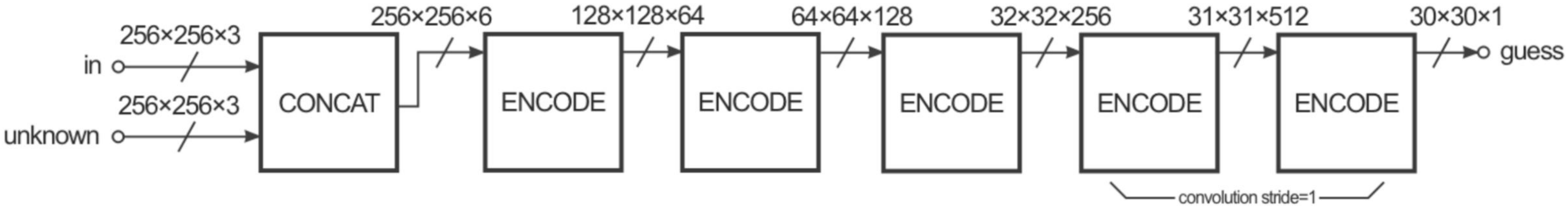
Architecture of Generator: Encoder Decoder Architecture with Skip Connections



Source: https://medium.com/@ManishChablani/cyclegans-and-pix2pix-5e6a5f0159c4

GANs - III

Architecture of Discriminator: PatchGAN as each pixel of 30x30 output corresponds to believability of a 70x70 patch of the input image

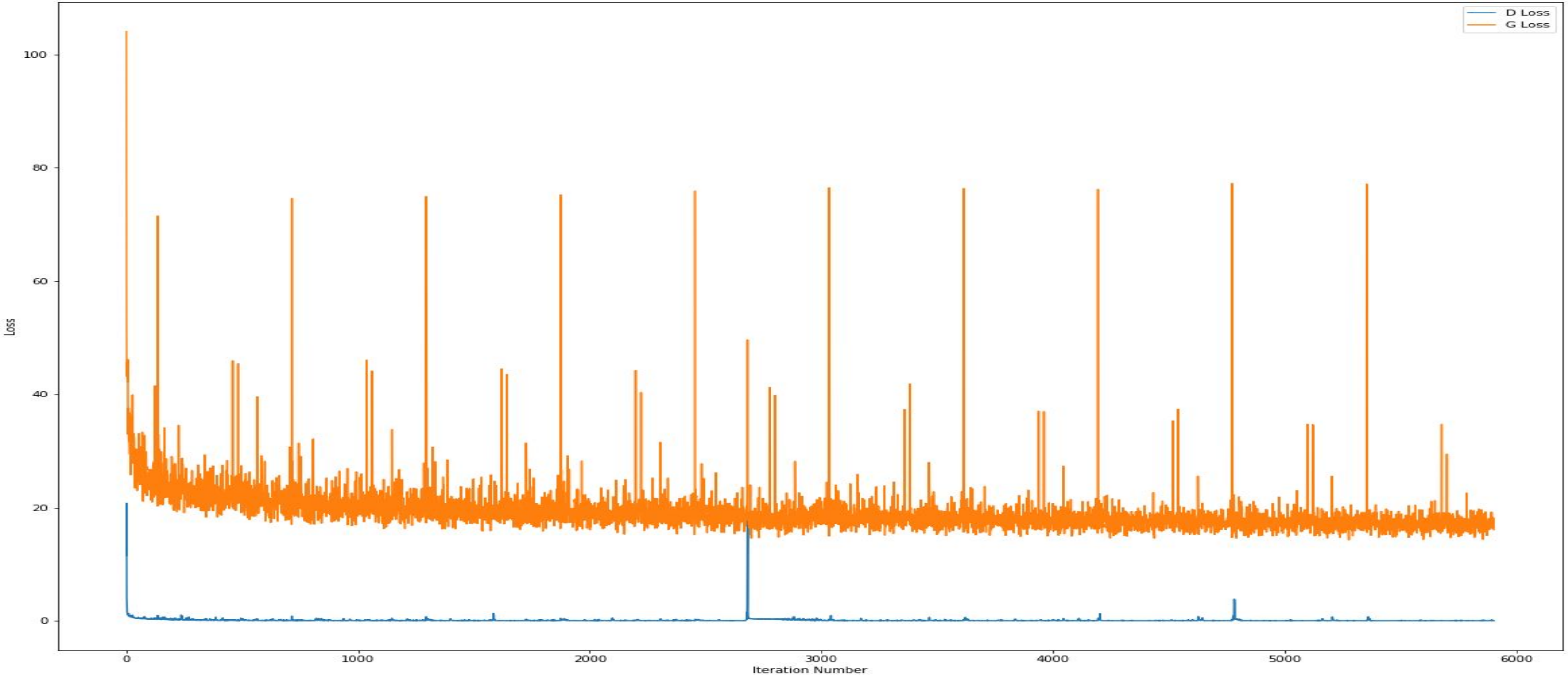


Source: https://medium.com/@ManishChablani/cyclegans-and-pix2pix-5e6a5f0159c4

GANs - IV

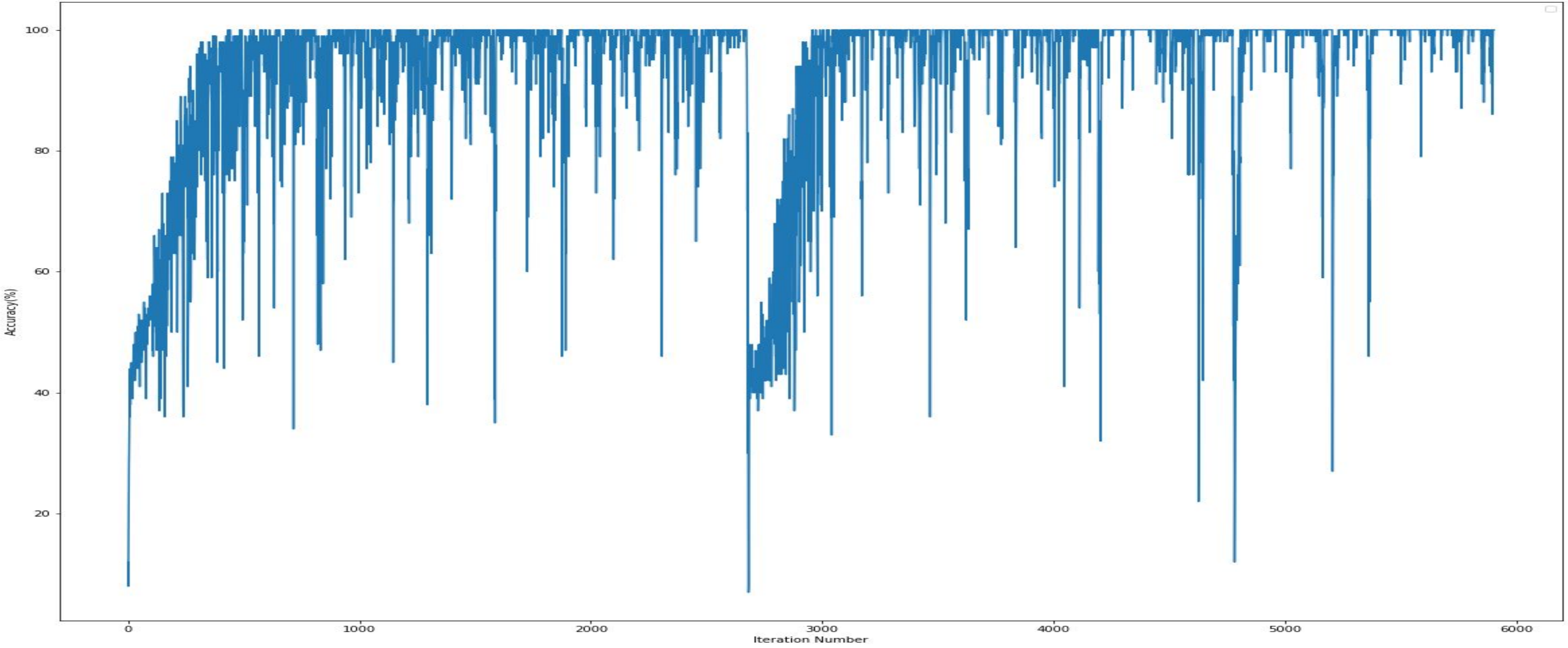
- Model adapted from https://github.com/eriklindernoren/Keras-GAN/blob/master/pix2pix/pix2pix.py
- Used 256x256x1 input aka reduced number of channels from 3 to 1
- Output shape of Discriminator is 16x16x1
- Adam Optimizer used with 0.0002 as learning rate
- Mean Squared Error Loss used for Discriminator
- Mean Squared Error + Mean Absolute Error Loss used for Combined Model
- Loss weights as 1 for MSE and 100 for MAE

GANs - V



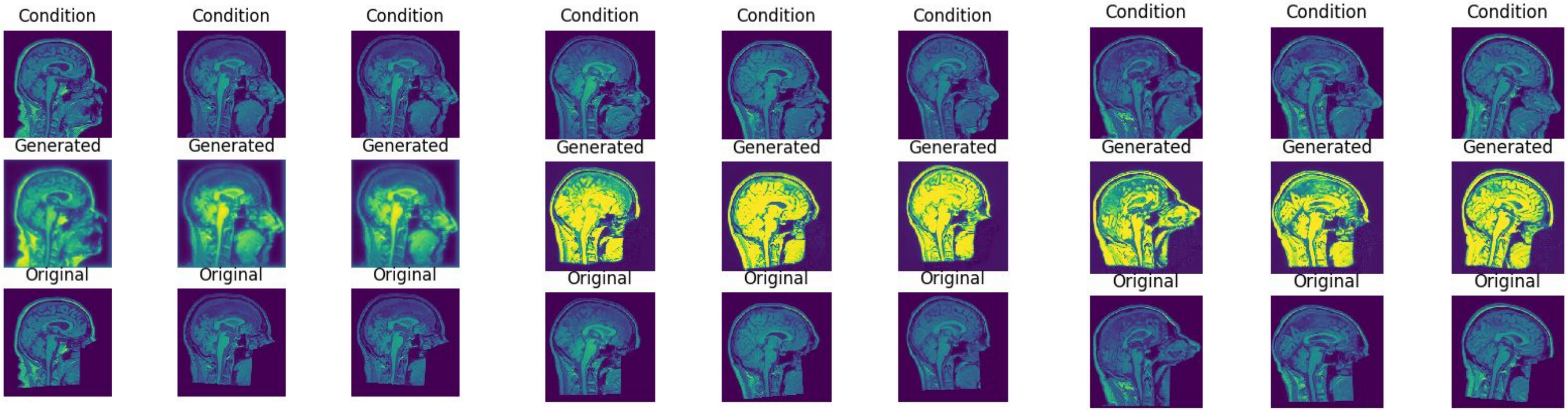
D Loss
G Loss
100
80
60
40
20
0
Loss
0
1000
2000
3000
4000
5000
6000
Iteration Number

GANs - VI

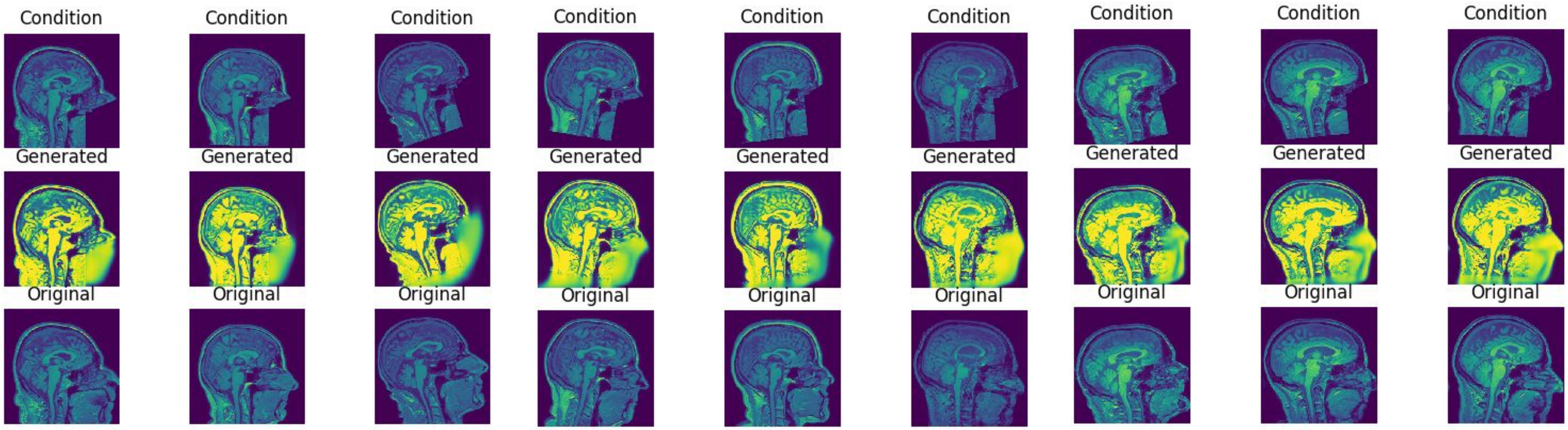


100
80
60
40
20
Accuracy(%)
0
1000
2000
3000
4000
5000
6000
Iteration Number

GANs - VII



GANs - Additional - Attempt at Refacing!



GANs - Results/Analysis

- Anonymization methods are vulnerable
- Although, Restoring complete face is challenging
    - The trajectory traced by the reconstructed face matches
    - Learns to regenerate smooth nose and jaw
- The Defacing algorithm currently employed by tools such as pydeface are partially reversible

Future Extensions

Future Extensions

- Considering T2, PD-weighted images as well
- Explore CycleGANs
- Use SSIM (Structural Similarity Index) for Evaluation
- Try Refacing/Defacing on different slices rather than just the middle slice, and combine those 2D slices to get a 3D volume
- Try using 3D GANs for volume reconstruction
- Make a Web Tool out of the classification model

Fin.