Skoltech
Skolkovo Institute of Science and Technology

MRI attributes manipulation

Deep Learning project

Final Extended Report

Team “MRI Mania”

Artem Moskalev
Artur Grigorev
Artem Sevastopolskiy
Denis Prokopenko

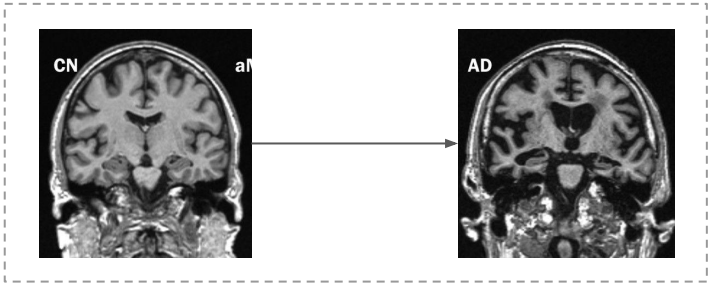
The Task

- **Goal:** simulate the process of brain aging with and without Alzheimer disease on MRI images
  - Fade MRI over **age**
  - Fade MRI over **stages** of Alzheimer

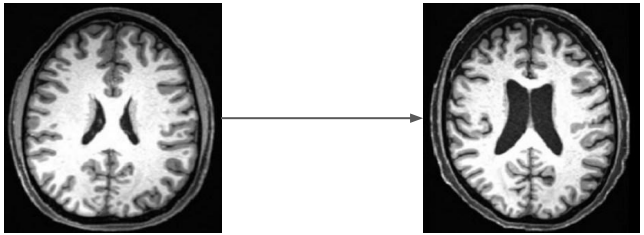
- **Motivation**: observing the process of brain aging and Alzheimer disease progression might help to predict possible complications

- **Methods:** generative and latent models:
  - Deep Feature Interpolation [1]
  - IcGAN [2]
  - Fader Networks [3]
  - InfoGAN [4]
  - CycleGAN [5]

Alzheimer disease



Brain aging



CelebA

- **10,177** number of **identities**,
- **202,599** number of aligned **face images**,
  resolution: 178 x 178,
- 5 landmark locations,
- 40 binary attributes annotations per image.

Plan:

1. try baseline models on CelebA
2. Adopt them for MRIs

CelebA - sample images grouped by attributes.

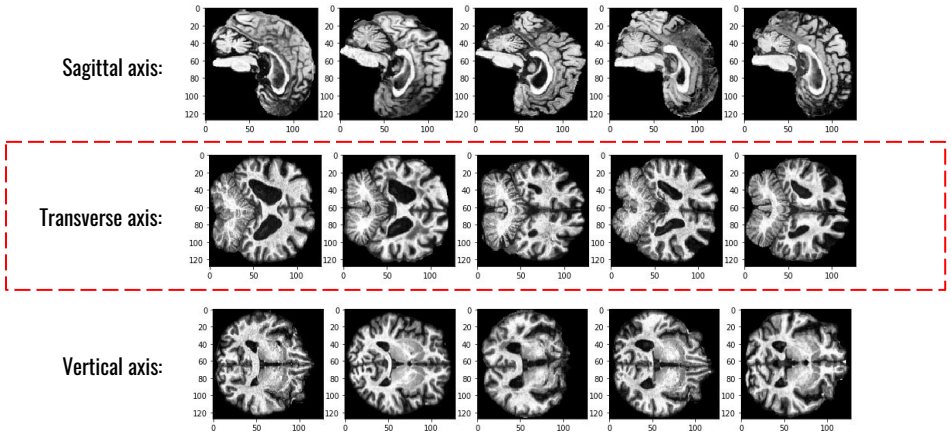
ADNI dataset

Skoltech
Skolkovo Institute of Science and Technology

Co-registration was a pain :((

4968 MRIs **1582** patients **T1** modality

- N4 bias field corrected
- Skull stripped
- Side zeros cropped
- Taking **2D** 128x128 slices

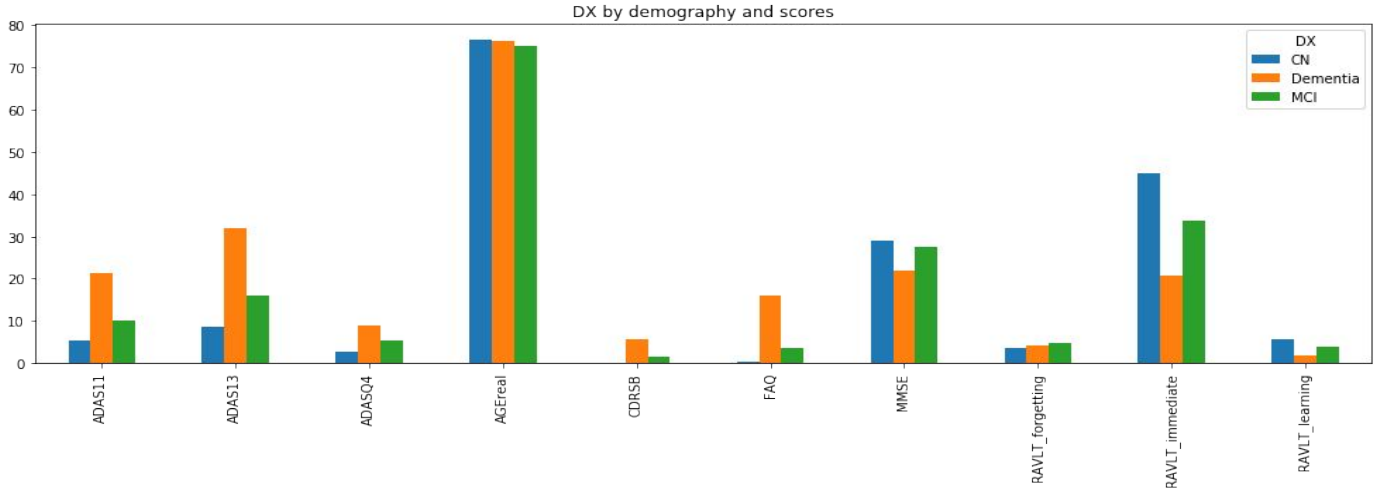


Attribute Selection

Skoltech
Skolkovo Institute of Science and Technology

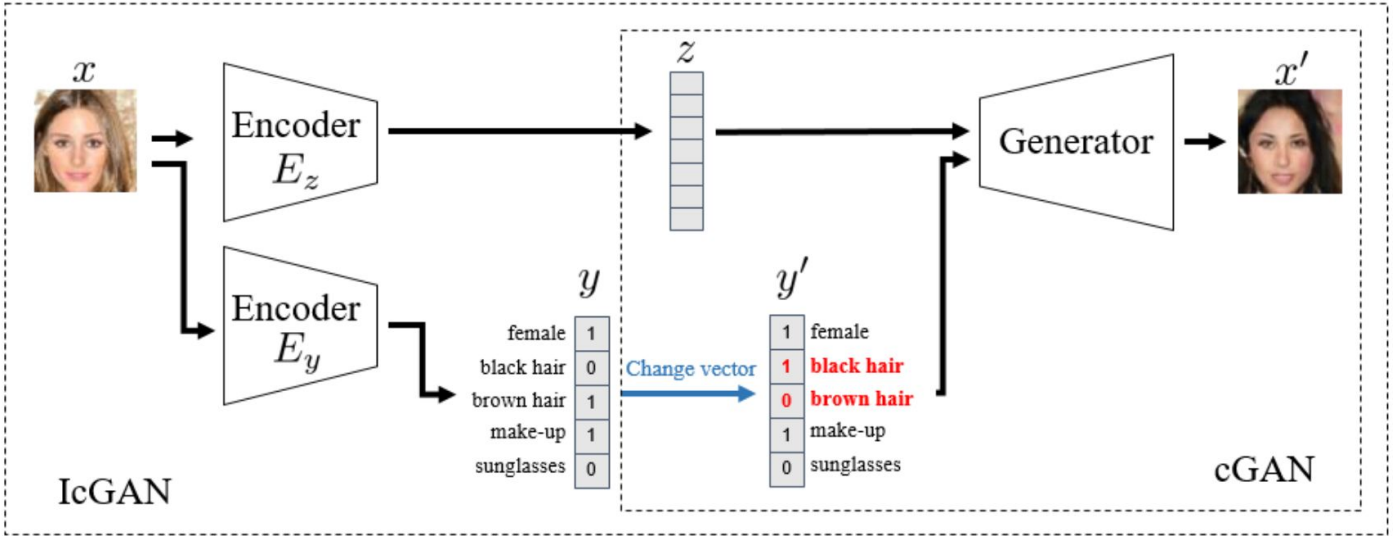
- Alzheimer Disease Assessment Scale (**ADAS**)
- Functional Activities Questionnaire (**FAQ**)
- Rey Auditory Verbal Learning Test (**RAVLT**)
- Diagnosis (**CN/MCI/Dementia**)

*we selected attributes that differ most from one diagnosis to another



Invertible Conditional GAN

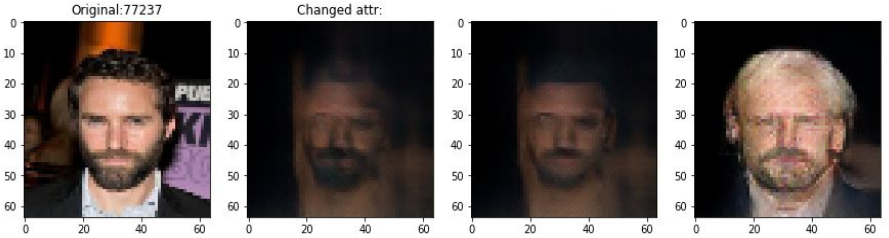
IcGAN: Method Description

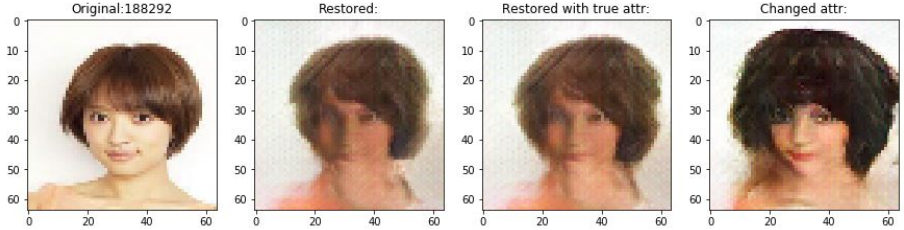


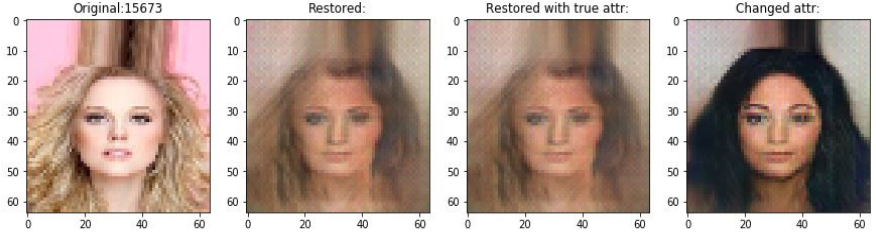
IcGAN = Encoder + conditional GAN

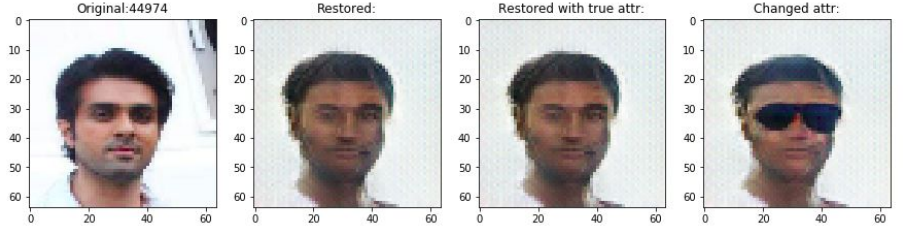
[4] Perarnau, Weijer, et al. "Invertible Conditional GANs for image editing." *NIPS 2016 Workshop on Adversarial Training*. 2016.

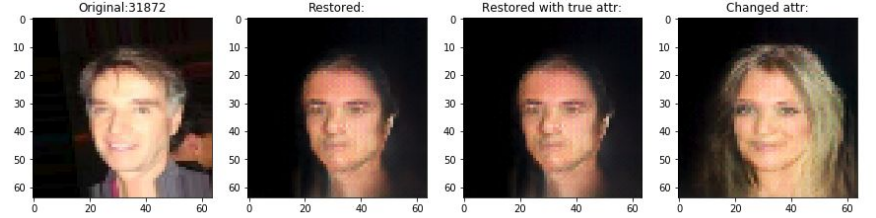
IcGAN: Results





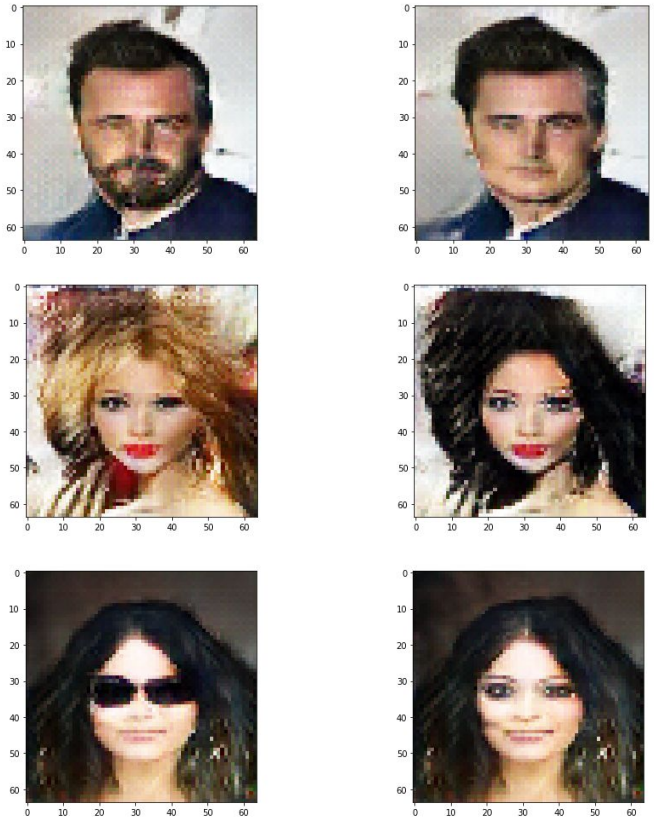






IcGAN: Results on CelebA

- Just cGAN works well
- Encoder suffers
- Images on this slide are just cGAN generated, images from slide 8 are IcGAN results



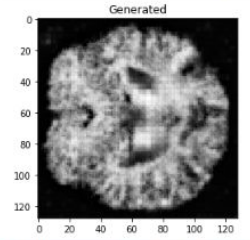
- cGAN trained for 25 epochs in matching-aware fashion exactly as in **[2]**
- Encoder trained with parameters from **[2]**
- hair color, mustache, glasses, gender were selected as attributes
- Implementation based on DCGAN (see used code)

IcGAN: Results on ADNI

Skoltech
Skolkovo Institute of Science and Technology

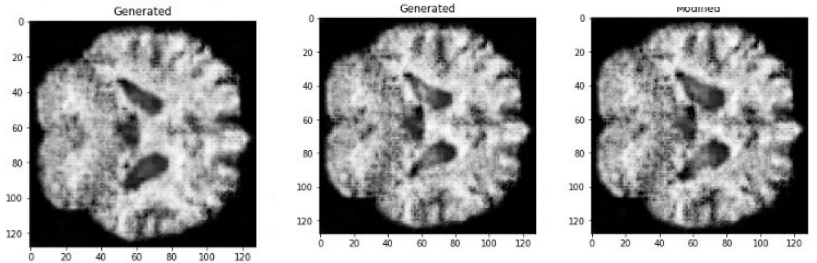
*conditioned on diagnosis

- Appeared to be weak model
- Fail to generate plausible image
- Spends a lot of training time to just reproduce the **shape**



*conditioned on diagnosis & laplacian

- Can more or less reproduce the shape
- Produce results that do not differ much from the **mean**



IcGAN: Training

Skoltech
Skolkovo Institute of Science and Technology

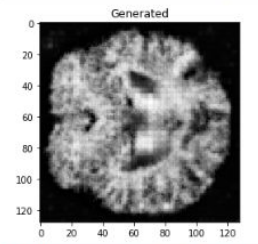
*conditioned on attributes

cGAN trained for 100 epochs in matching-aware fashion with Adam optimizer, lr=0.0001, betas=(0.5,0.999) as in **[2]**. With matching-aware training we ask discriminator to distinguish real images with true attributes from real images with wrong attributes and fake images with real attributes **[7],** 12-th slide Figure 1.

cGAN and respective discriminator conditioned on ADAS, FAQ, Age, Diagnosis. To sample wrong conditions, diagnosis is reverted and the rest of attributes is taken as the means for specific diagnosis.

Produced images are earned by trying to interpolate an image generated by cGAN.

Decided not to spend time to make it invertible and not to train encoders.



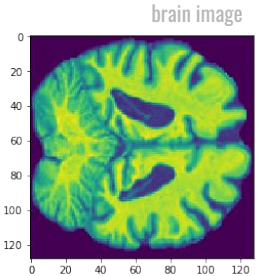
*conditioned on attributes & laplacian

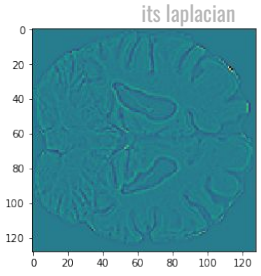
In order to help cGAN to reproduce the shape, laplacian is passed through convolution and concatenated with generator's feature maps.

Trained and conditioned in the same way as described above.

Decided not to spend time to make it invertible and not to train encoders.

Conditioning on several attributes was a bad idea since they at most are highly correlated.





cGAN: Implementation details

Skoltech
Skolkovo Institute of Science and Technology

Generator:

- Same as in **[2]**, but additional [TransposeConv2d, BN, ReLU] block is added to fit 128x128 images, where convolutions are made with (4,4) kernels, (2,2) stride.
- Initially noise vector **z** and condition vector **y** are concatenated and passed through a generator.
- Laplacian (128x128) is passed through 2 convolutions to (32x32) size and concatenated with output of the 4-th generator [TransposeConv2d, BN, ReLU] block.
- Tanh is on the top.
- Up to 1024 feature maps are used.

Discriminator:

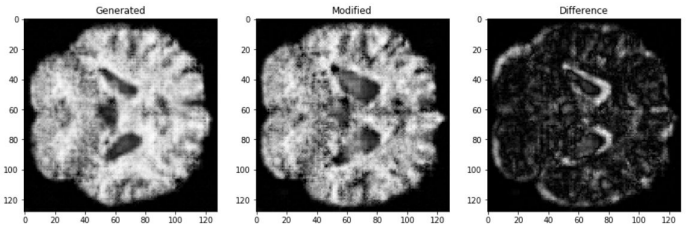
- Same as in **[2]**, but additional [Conv2d, BN, ReLU] block is added to fit 128x128 images, where convolutions are made with (4,4) kernels, (2,2) stride.
- Conditioning attributes are repeated to fit the size of input image, then passed through a [Conv2d, ReLU] to (64x64) and concatenated with other discriminator's feature maps of this size.
- Discriminator outputs **0's** for fake generated images and for real images with wrong conditions.
- Up to 1024 feature maps are used

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z,h)$ {Forward through generator}
  $s_r \leftarrow D(x,h)$ {real image, right text}
  $s_w \leftarrow D(x,\hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x},h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1-s_w) + \log(1-s_f))/2$
  $D \leftarrow D - \alpha\partial\mathcal{L}_D/\partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha\partial\mathcal{L}_G/\partial G$ {Update generator}
end for

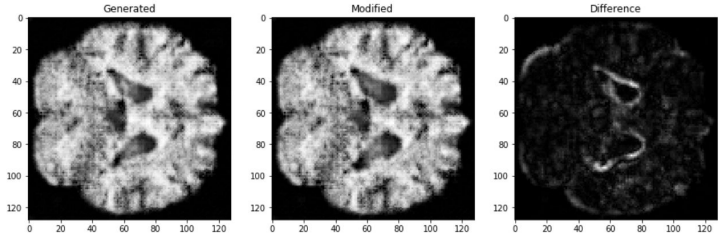
Figure 1. Training algorithm from [7] was adopted. We just substitute "text conditions" with conditioning attributes that we mentioned on 5-th slide.

IcGAN: Results on ADNI

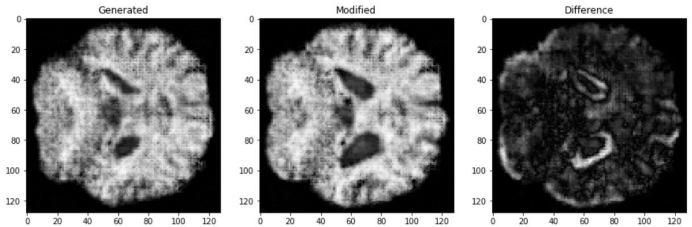
- **ADAS** interpolation



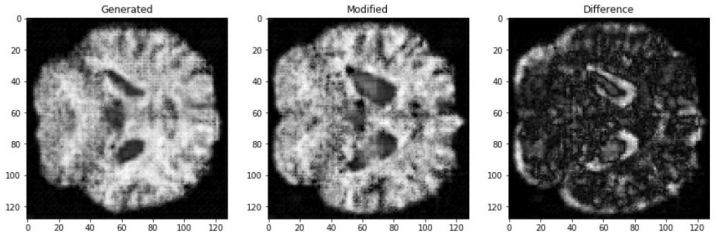
- **Diagnosis** interpolation



- **FAQ** interpolation



- **ADAS&Diagnosis&FAQ** interpolation

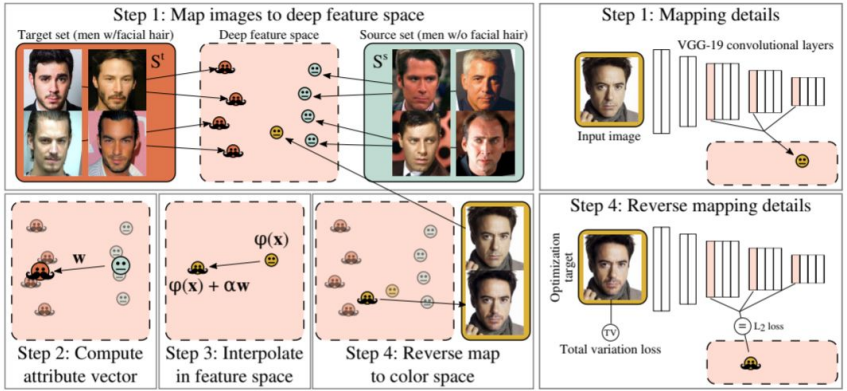


Skoltech
Skolkovo Institute of Science and Technology

Deep Feature Interpolation

Deep Feature Interpolation: Method Description

Optimization-based method, which requires an aligned set of images.
A version described below was used in the project:
it seeks neighbors in the attribute space and does not use scaled step size.



The task is to transform an image $\mathbf{x} \in S^s$ (source set) into an image in S^t (target set). For instance, the source set can represent images of people with facial hair, and the target set - people without facial hair.

1. VGG-19, pretrained on ImageNet, is used as a fixed feature extractor $\phi(\mathbf{x})$ (conv3_1, conv4_1, conv5_1 layers are used)
2. Compute the difference between mean feature vectors from two sets: $\mathbf{w} = \bar{\phi}^t - \bar{\phi}^s$.
3. Reversing the mapping by solving an optimization task:

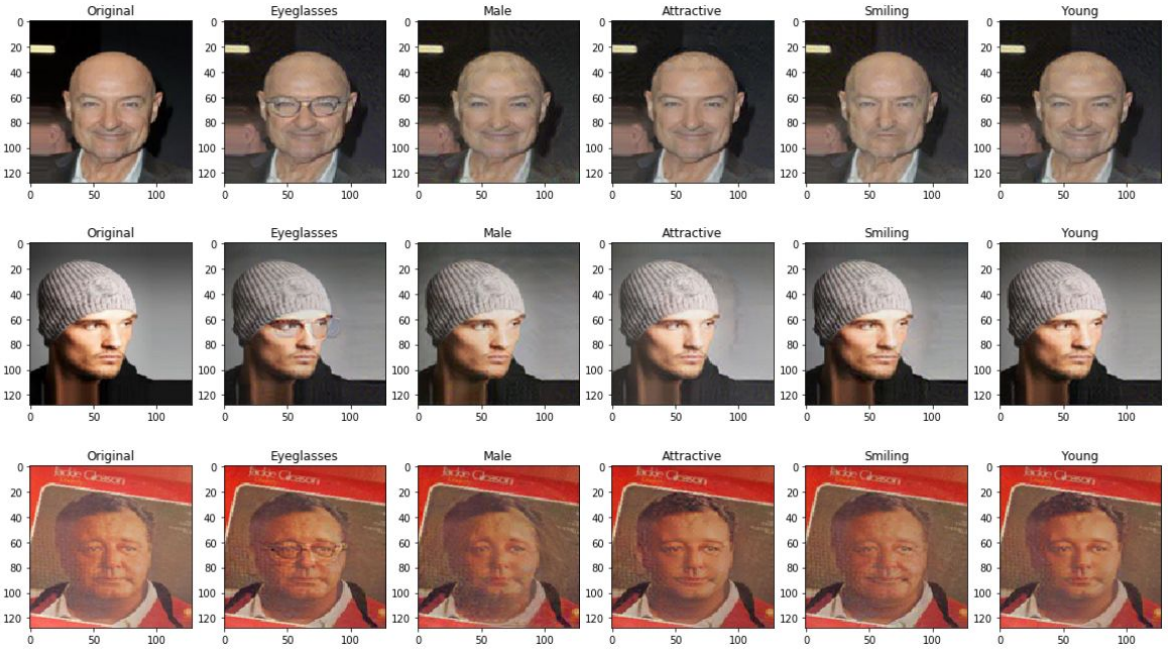
$$\mathbf{z} = \arg\min_{\mathbf{z}} \frac{1}{2}\|(\phi(\mathbf{x}) + \alpha\mathbf{w}) - \phi(\mathbf{z})\|_2^2 + \lambda_{V^\beta} R_{V^\beta}(\mathbf{z}), \qquad \text{where} \qquad R_{V^\beta}(\mathbf{z}) = \sum_{i,j} \left((z_{i,j+1} - z_{i,j})^2 + (z_{i+1,j} - z_{i,j})^2\right)^{\frac{\beta}{2}}$$

The source and target sets are taken as the sets of *K* nearest neighbors in a small-dimensional attribute space:.

[1] Upchurch, Gardner et al. "Deep Feature Interpolation for Image Content Changes." *CVPR 2017*.

Deep Feature Interpolation: Results on CelebA

- The method yields relatively meaningful and high-quality reconstructions.
- Method does not require training, but the inference time is very slow (*~25 s* on Geforce GTX 1080).
- K=100 neighbors were sought in the CelebA attribute space, consisting of 40 binary attributes (Eyeglasses, Smile, Gender, etc.)
- Attribute vector scaling techniques from the original paper and λ_{V^β} scalar fine-tuning did not improve the results.



Attribute swapping on random images from CelebA.

Deep Feature Interpolation: Modifications for ADNI

Skoltech
Skolkovo Institute of Science and Technology

- K=100 neighbors were sought in the following attribute space:
  ADAS13, FAQ, diagnosis, education level, race, ethnicity
  using Mahalanobis distance instead of L2 / cosine, as some attributes are extremely correlated.

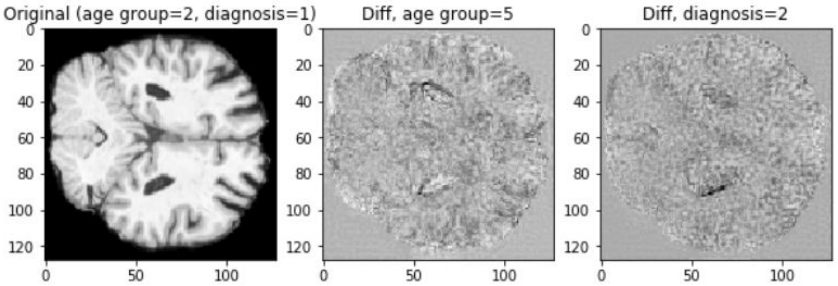
- L-BFGS used instead of GD, due to its faster convergence.

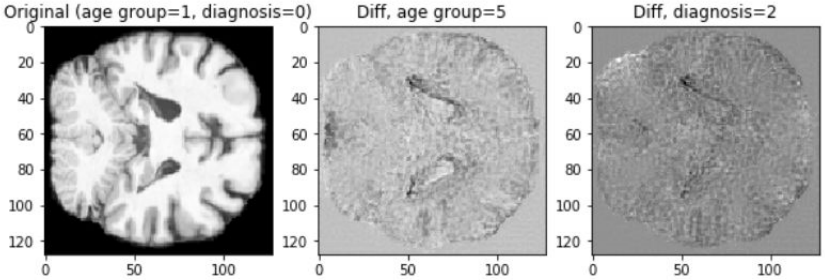
- Since ImageNet did not contain MRI-like images, two types of encoder were employed:
  - VGG, pretrained on ImageNet,
  - VGG, trained on a separate training set from ADNI to predict *age* and *DX diagnosis score*.

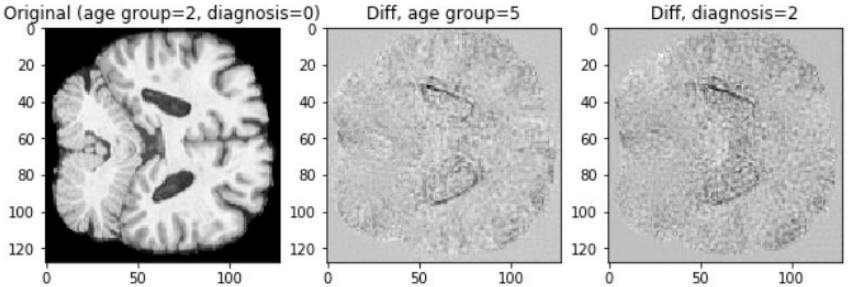
Deep Feature Interpolation: Results on ADNI

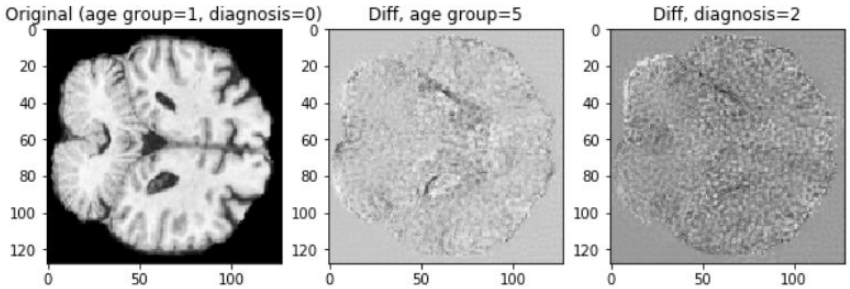
Skoltech
Skolkovo Institute of Science and Technology

Features: conv3_1, conv4_1, conv5_1 from ImageNet-pretrained VGG-16





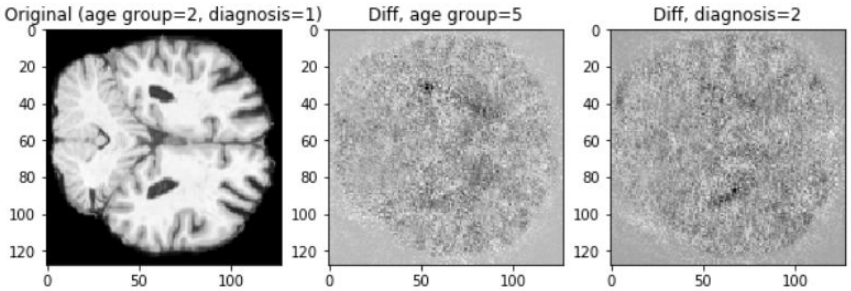


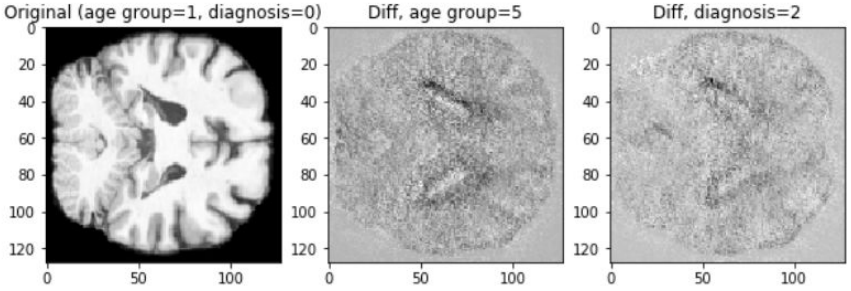


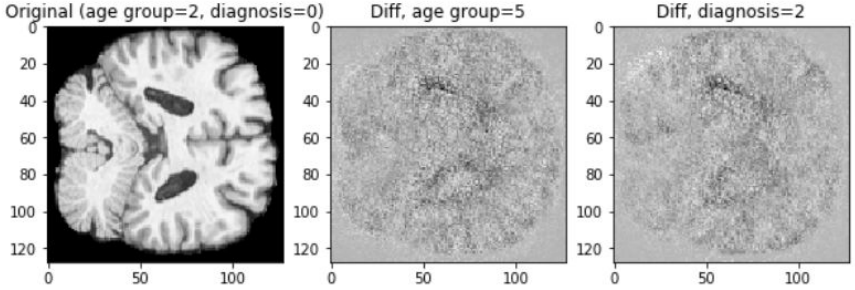
Deep Feature Interpolation: Results on ADNI

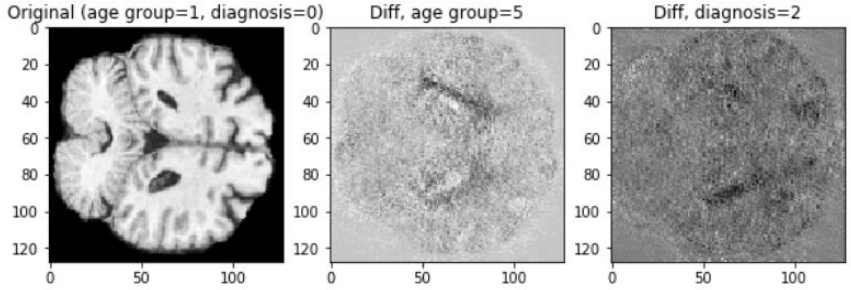
Skoltech
Skolkovo Institute of Science and Technology

Features: conv3_1, conv4_1, conv5_1 from VGG-16, trained on ADNI (separate training set) to predict *age* and *diagnosis*







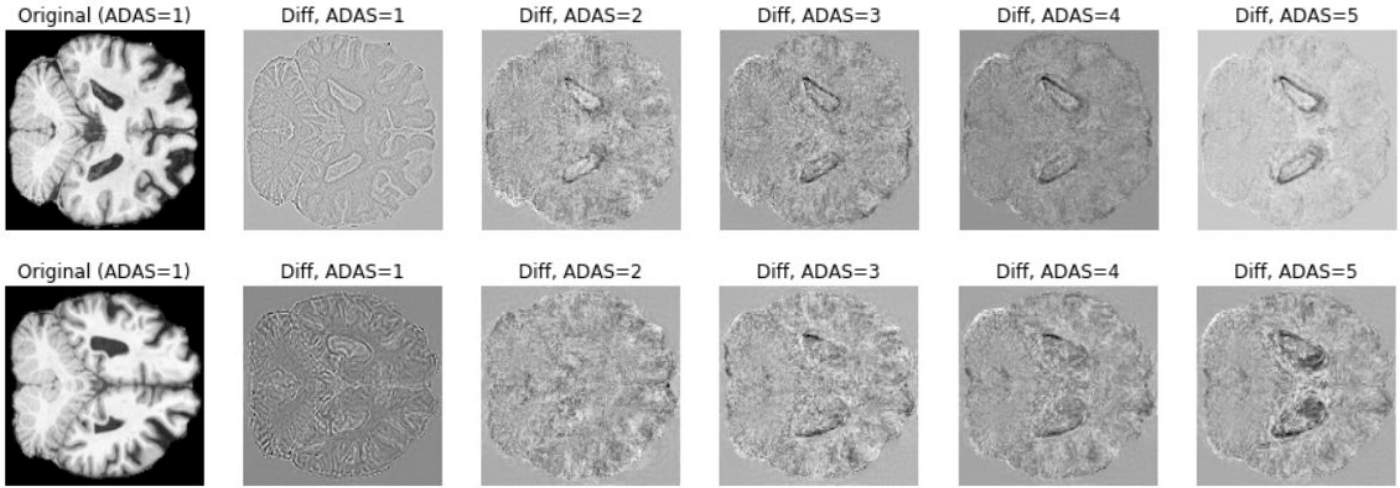


Deep Feature Interpolation: Results on ADNI

Skoltech
Skolkovo Institute of Science and Technology

Encoder: conv3_1, conv4_1, conv5_1 from VGG-16, trained on ADNI (separate training set) to predict *age* and *diagnosis*

Interpolation over ADAS diagnosis score:

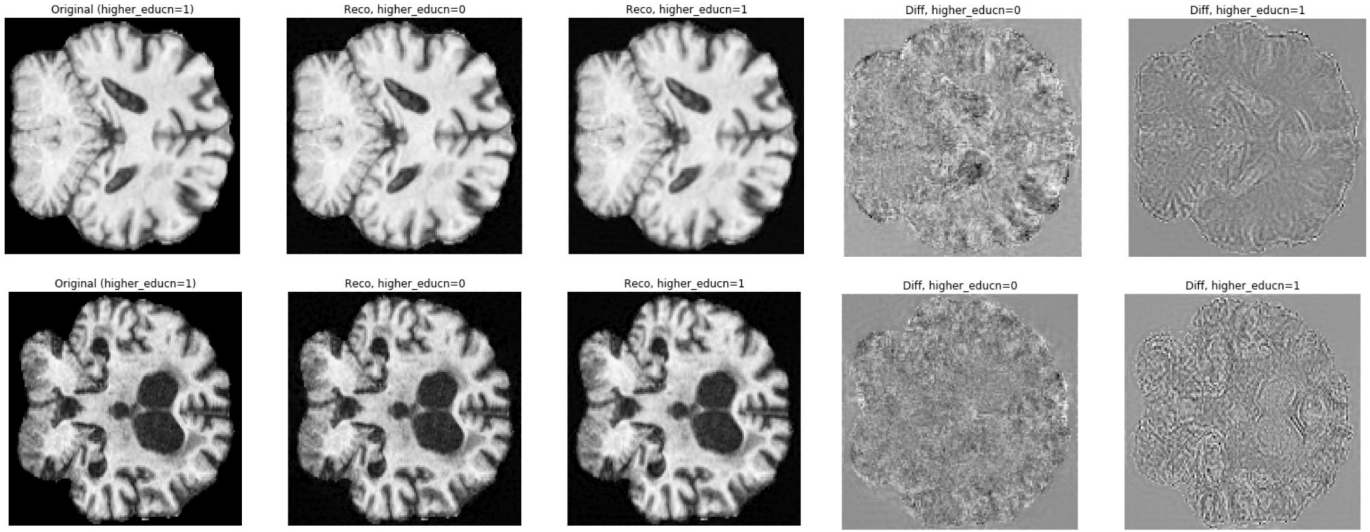


Deep Feature Interpolation: Results on ADNI

Skoltech
Skolkovo Institute of Science and Technology

Encoder: conv3_1, conv4_1, conv5_1 from VGG-16, trained on ADNI (separate training set) to predict *age* and *diagnosis*

with / without higher education:



The resulting difference is close to random noise., so this experiment can be considered a sanity check.

Deep Feature Interpolation: Conclusion

Skoltech
Skolkovo Institute of Science and Technology

- With both encoders, the method discovers the borders of ventricles as the Alzheimer-related features, and the same features are emphasized for age change.

- However, this transformation is very far from expected (for instance, no Alzheimer-related atrophy is captured). Hence, some of the features are captured by the method, but the reconstructions can not be considered satisfactory.

  As similar features of Alzheimer were discovered by IcGAN, this is likely a consequence of using too small dataset or imprecise slice selection procedure.

Fader Networks

Fader networks: Method Description

Skoltech
Skolkovo Institute of Science and Technology

The architecture described in the paper [3] consists of three main parts: encoder, decoder and discriminator (see next slide).

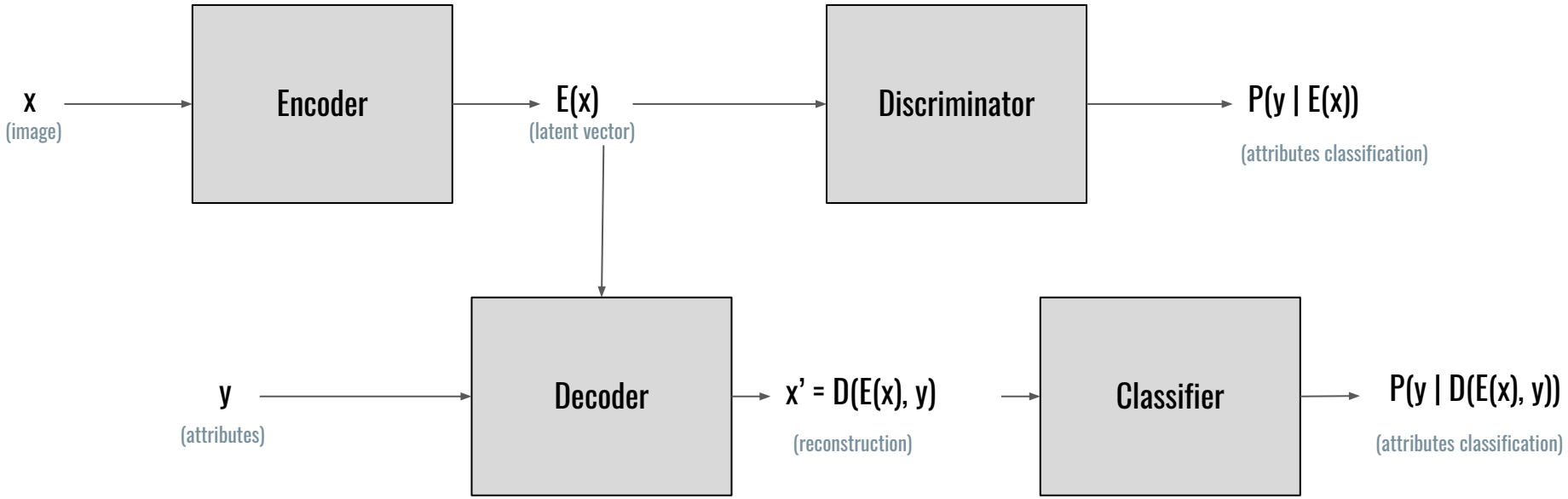
- **Encoder** maps images into latent space.
- **Decoder** reconstructs image given latent vector and attributes vector.
- **Latent discriminator** tries to classify attributes of an image by its' latent vector.

The network is trained in adversarial manner where not only encoder-decoder part aims to draw good reconstructions but also to fool discriminator. Due to such training no information about attributes is stored in latent vectors, therefore for good reconstructions it is necessary for decoder to use attribute vector. And so, changing attribute passed into decoder, we can alter visual appearance of reconstruction

Apart from the architecture presented in paper, it turned out that a good way to improve performance of the network is to add a classifier to it.

- **Classifier discriminator** is trained on both real images and reconstruction to classify their attributes, it encourages decoder to reconstruct more attribute-dependent images.

Fader networks: Architecture



Fader networks: Training Process

Skoltech
Skolkovo Institute of Science and Technology

Prior to training the fader network itself, we've trained an evaluation classifier that classifies attributes based on images from training set. It is used then to evaluate how good the network reconstructs images with altered attributes, its' weights do not change during the fader network training.

Each iteration of fader network training consists of the following consecutive steps:

- **Latent discriminator step**: a batch of images is mapped into latent space and then latent discriminator trains to classify correct attributes. Loss function for the latent discriminator can be written down as follows:

$$\mathcal{L}(\theta_{lat\_dis}) = -logP_{\theta_{lat\_dis}}(y|E_{\theta_{enc}}(x))$$

- **Classifier discriminator step**: classifier discriminator trains to predict attributes by real images:

$$\mathcal{L}(\theta_{clf\_dis}) = -logP_{\theta_{clf\_dis}}(y|x)$$

- **Autoencoder step**: updating weights of the encoder-decoder part. Loss function of autoencoder consists of three parts:

  - reconstruction loss: mean squared error between real image and reconstruction with original attributes
  - latent discriminator loss: cross-entropy between discriminator predictions and invalid attributes
  - attribute reconstruction loss: encoded latent vectors passed into decoder with altered attributes, then cross-entropy between classifier predictions and those attributes calculated.

$$\begin{aligned}\mathcal{L}(\theta_{enc}, \theta_{dec}) = & mse(D_{\theta_{dec}}(E_{\theta_{enc}(x)}, y), x) - \\ & \lambda_{lat} logP_{\theta_{lat\_dis}}(1 - y|E_{\theta_{enc}}(x)) - \\ & \lambda_{clf} logP_{\theta_{clf\_dis}}(1 - y|D_{\theta_{dec}}(E_{\theta_{enc}(x)}, 1 - y))\end{aligned}$$

Fader Networks: Experiments on CelebA

Skoltech
Skolkovo Institute of Science and Technology

First, CelebA dataset was split into train and validation set, size of validation set was taken as 0.05 of the whole dataset, or 1012 images.

We've trained **evaluation classifier** on the whole celebA set for two attributes: sex and smile. It reached accuracy of 97.4% and 92.3% for those attributes on the validation set respectively.

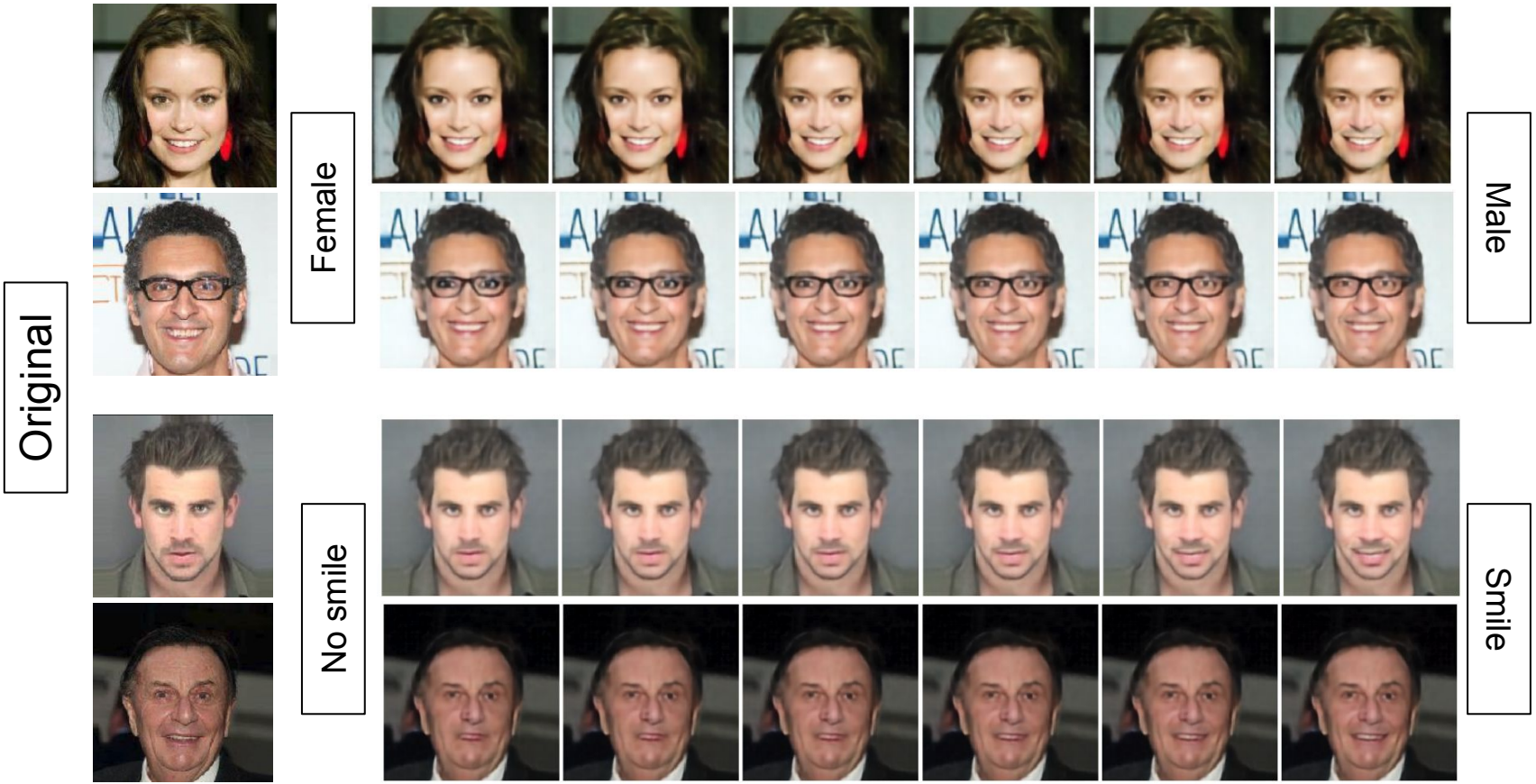
The main architecture was trained during 2.2 million steps with batch size = 32, it took around 20 hours to train on GeForce 970. After the training, the network learned to reconstruct faces with altered attributes with accuracy higher than 90% for evaluation classifier to confirm altered attributes:

- **97.1%** accuracy for sex attribute
- **91.1%** accuracy for smile attribute.

MSE for original attributes is 1,14e-3.
Examples of attribute modification can be seen on the next slide.

Fader Networks: results on CelebA (~200K images)

Skoltech
Skolkovo Institute of Science and Technology



Fader Networks: Architecture Tuning

Skoltech
Skolkovo Institute of Science and Technology

While the original architecture can be used only for binary labels, to work with ADNI dataset it was necessary to tune the network so that it also works with multiclass labels and real-valued labels. So now output vectors of discriminators have one element for each binary or real-valued attribute, plus the number of elements equal to the number of classes for each multiclass label.

To calculate loss, those output vectors are now split into parts that refer to each attribute and appropriate loss functions applied to each of them:

- mean squared error for real-valued attributes
- sigmoid + binary cross-entropy for binary
- softmax + cross-entropy for multiclass

Also to train classifier discriminator to classify altered attributes we introduce different ‘flip_attribute’ procedure for each attribute type:

- for binary: 1 - y
- for multiclass: randomly choose another label
- for real: preemptively compute mean and standard deviation for that attribute in the dataset and draw values from gaussian distribution with those parameters.

Fader Networks: Experiments on ADNI

Skoltech
Skolkovo Institute of Science and Technology

We've trained fader network to interpolate for two attributes:

- Diagnosis (or DX) — multiclass (healthy, MCI or Dementia)
- Age — real-valued (in years)

The first problem we faced with ADNI due to comparatively small dataset size (4968 MRIs) was severe overfitting. In order to solve it, we, first, added dropout layers with p=0.5 to the architecture and, second, augmented the dataset with random brightness, contrast and saturation jitter.
Those resorts allowed for evaluation classifier to get better metrics on validation dataset, but not much, here are best validation metrics we got:

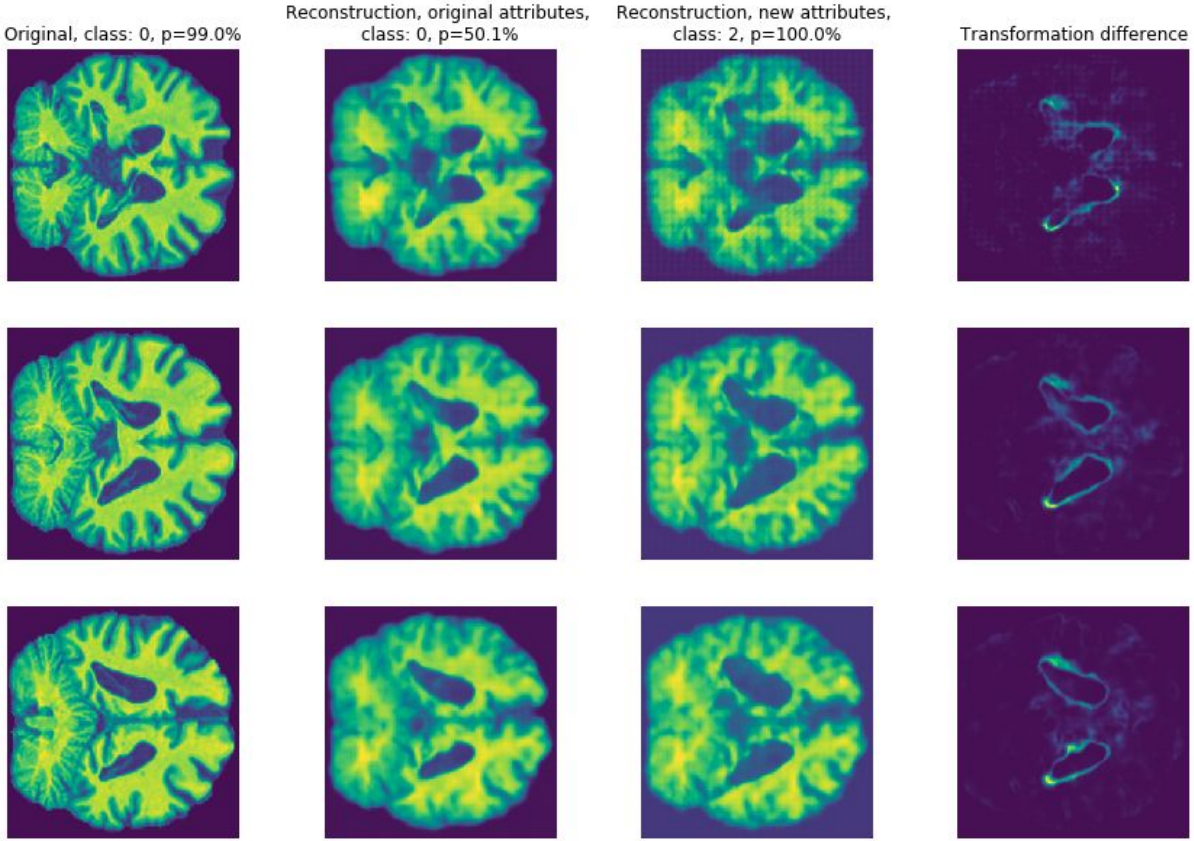
- **98.1%** accuracy for Diagnosis and **3.3** rmse for Age on train set,
- **64.3%** accuracy for DX and **5.6** rmse for Age

For ADNI dataset fader network was trained during 850K iterations with batch size 32 (~10 hours on GeForce 970).
On the next few slides you can see results of altering attributes for MRIs from train set. Each row corresponds to a separate slice of MRI.

Fader Networks: Results on ADNI (~5K images)

Skoltech
Skolkovo Institute of Science and Technology

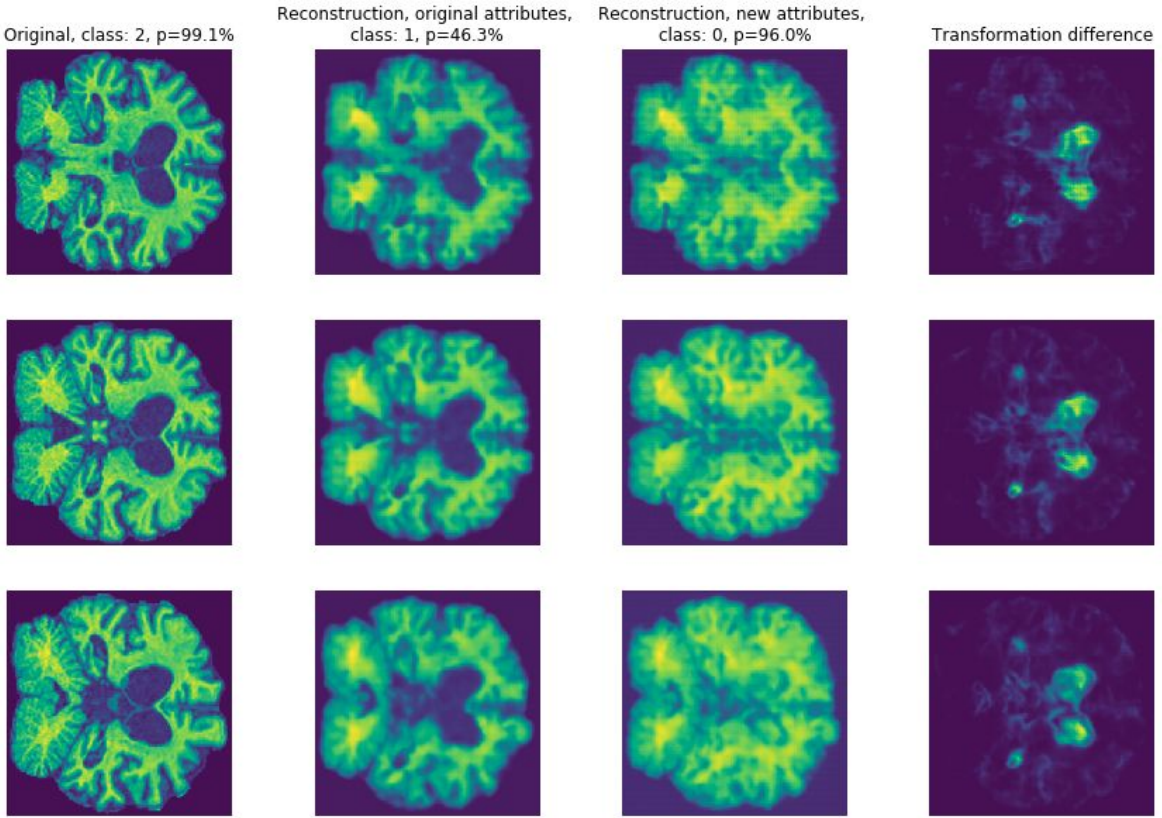
Healthy -> Dementia



Fader Networks: Results on ADNI (~5K images)

Skoltech
Skolkovo Institute of Science and Technology

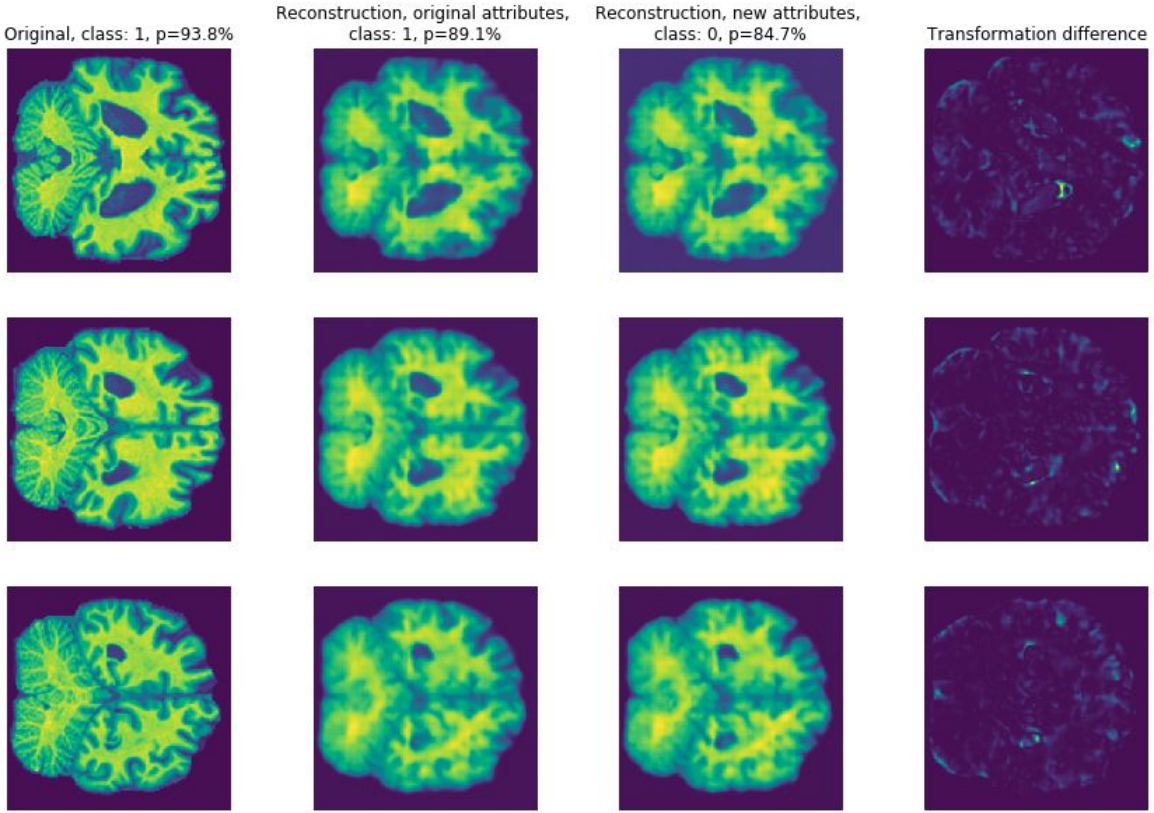
Dementia -> Healthy



Fader Networks: Results on ADNI (~5K images)

Skoltech
Skolkovo Institute of Science and Technology

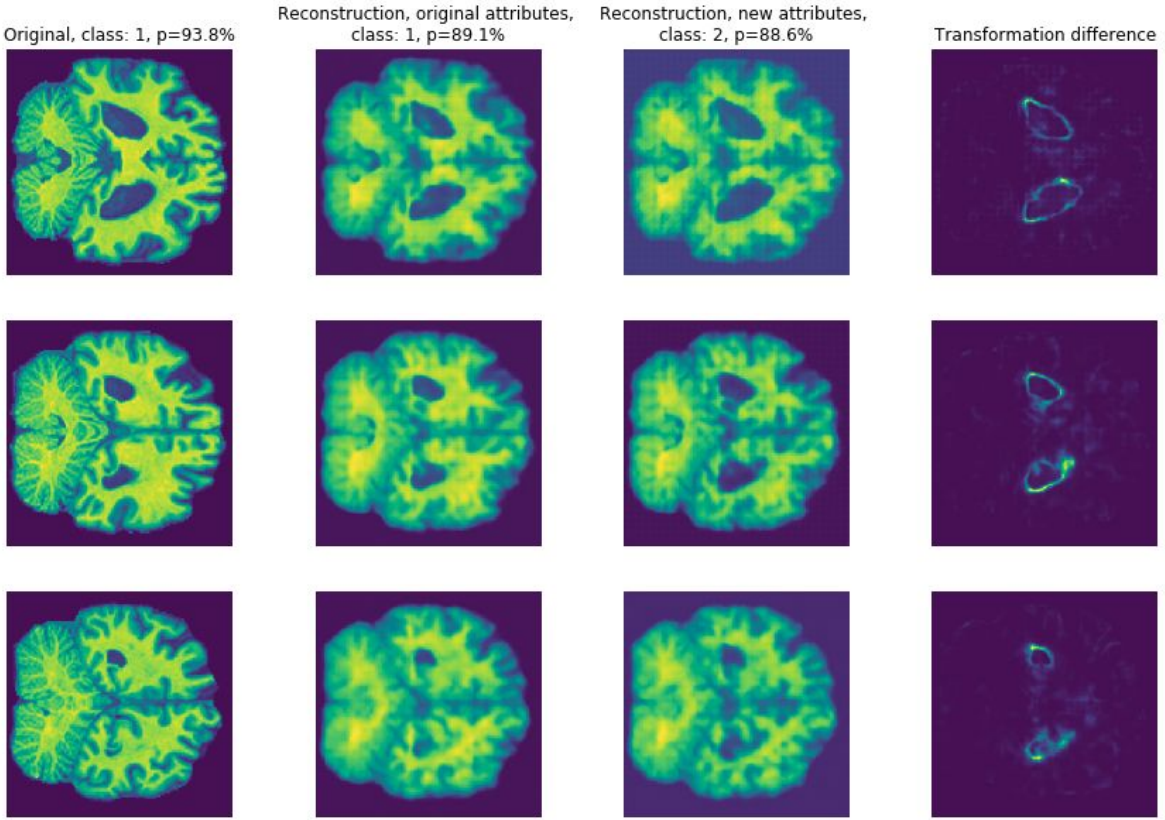
MCI -> Healthy



Fader Networks: Results on ADNI (~5K images)

Skoltech
Skolkovo Institute of Science and Technology

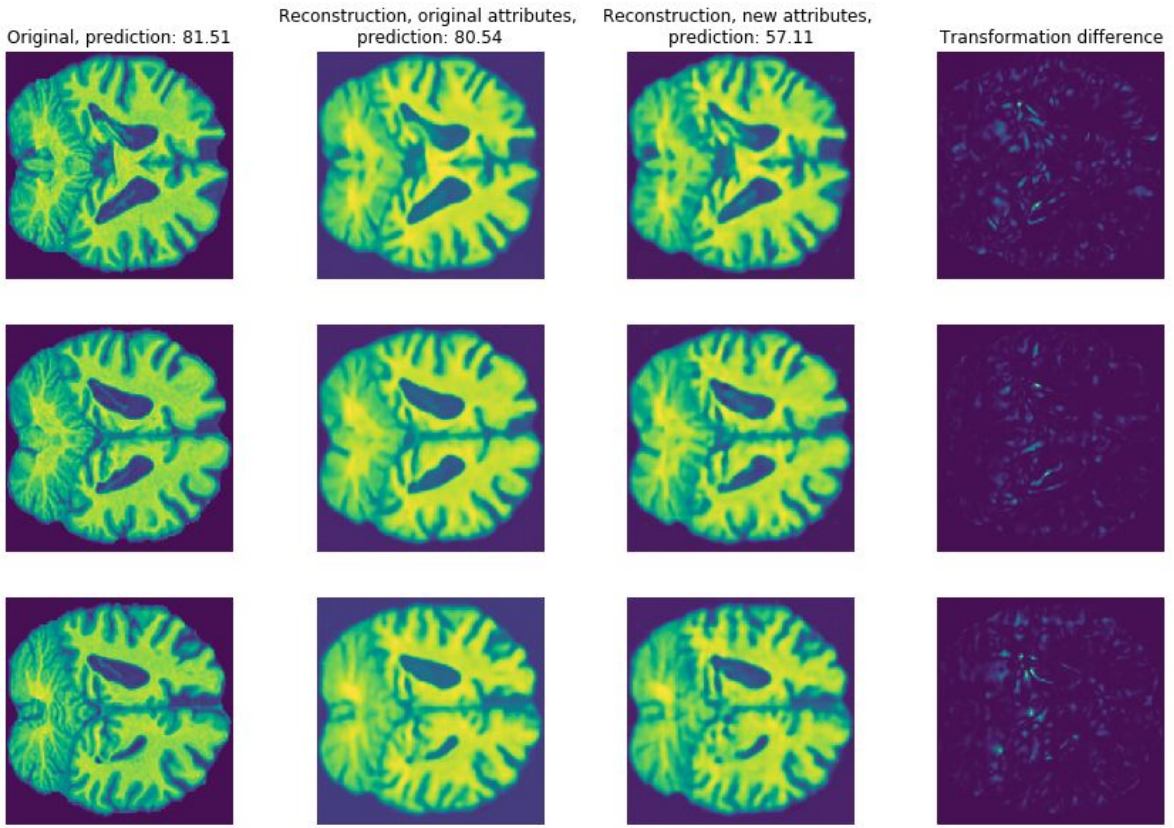
MCI -> Dementia



Fader Networks: Results on ADNI (~5K images)

Skoltech
Skolkovo Institute of Science and Technology

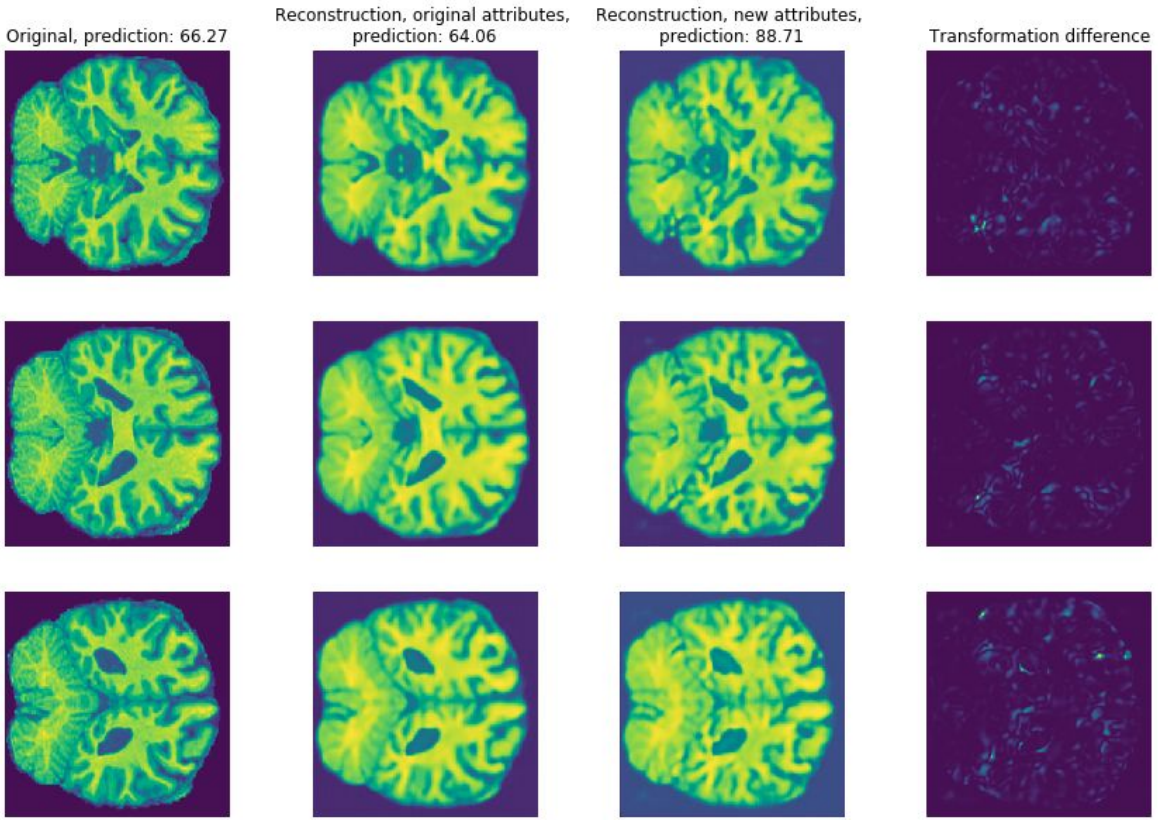
81y.o. -> 57y.o.



Fader Networks: Results on ADNI (~5K images)

Skoltech
Skolkovo Institute of Science and Technology

66y.o. -> 88y.o.



Fader Networks: Conclusion

- Works well on large datasets
- Reconstruction quality suffers on small datasets (ADNI)
- Overfits severely on 5000 images set
- Shows semantically meaningful transformation

InfoGAN

InfoGAN: Method

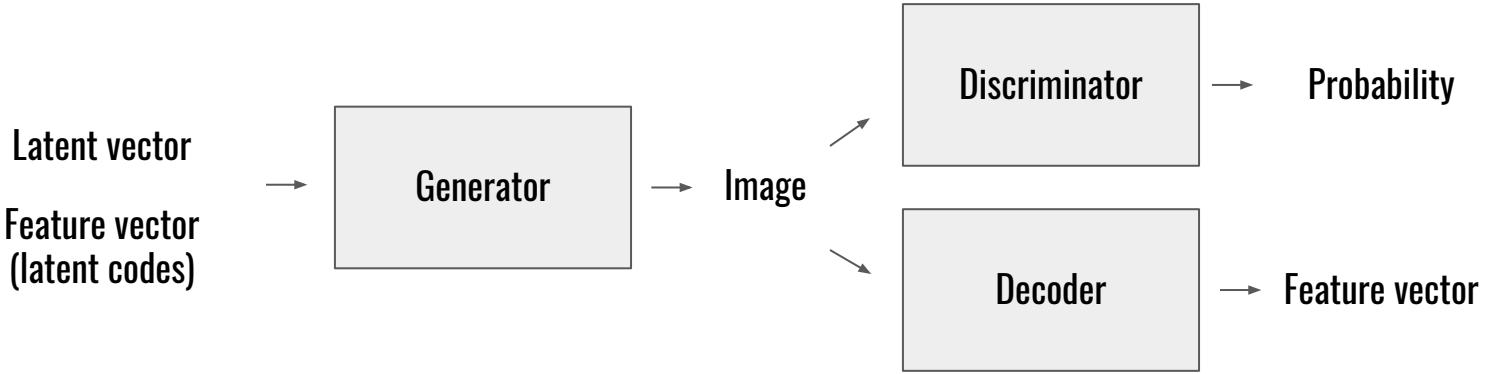
Skoltech
Skolkovo Institute of Science and Technology

InfoGAN is an extension of GAN that is able to learn representations in a completely unsupervised manner. The main difference between infoGAN and GAN is that the first one maximizes the mutual information between a small subset of the latent variables and the observation. InfoGAN architecture consists of:

- Generator — create image from random sampled latent vector with latent meaningful variables (i.e. features)
- Discriminator — evaluate the probability of credibility of generated image
- Decoder — derive the features from generated image.

Such as any GAN, infoGAN uses adversarial learning, where generator not only tries to fool discriminator but creates images with features which are extracted by decoder. Therefore, generator will use latent features instead of random variables from laten vector to generate an image with appropriate attributes.

InfoGAN: Method



InfoGAN: Training Process

Skoltech
Skolkovo Institute of Science and Technology

Let's start with GAN. As we know, it uses minmax game.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim \text{noise}}[\log(1 - D(G(z)))]$$

The first step to infoGAN is to use input vector which consists of two parts: source of incompressible noise (z) and latent code (c), which will target the salient structured semantic features of the data distribution. So we have independent $c_1, c_2, \ldots, c_L$ latent codes. Hence, $P(c_1, c_2, \ldots, c_L) = \prod_{i=1}^{L} P(c_i)$.

The generator now depends on noise and latent code: $G(z,c)$. Now we need to add regularization in order to increase the importance of information in latent codes. Thus, we use and try to maximize mutual information between latent codes and generated image: $I(c; G(z,c))$. Therefore, we add another term to the minmax game:

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c; G(z,c))$$

InfoGAN: Training Process

Skoltech
Skolkovo Institute of Science and Technology

However, it is quite complex task to calculate the mutual information because the posterio $P(c|x)$ is required. But we can use the lower bound of it using auxiliary distributior $Q(c|x)$ to approximate the posterior.

$$
\begin{aligned}
I(c; G(z,c)) &= H(c) - H(c|G(z,c)) \\
&= \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log P(c'|x)]] + H(c) \\
&= \mathbb{E}_{x\sim G(z,c)}[\underbrace{D_{\mathrm{KL}}(P(\cdot|x) \parallel Q(\cdot|x))}_{\geq 0} + \mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\
&\geq \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c)
\end{aligned}
$$

As a result we get lower bound, where latent code entropy is treated as constant.

$$
\begin{aligned}
L_I(G,Q) &= E_{c\sim P(c),x\sim G(z,c)}[\log Q(c|x)] + H(c) \\
&= E_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\
&\leq I(c; G(z,c))
\end{aligned}
$$

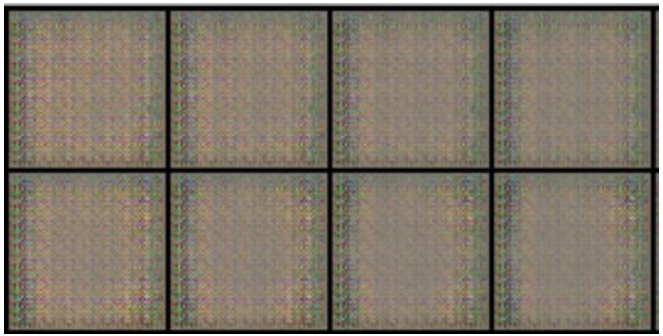
And minmax game becomes: $\min_{G,Q}\max_{D} V_{\mathrm{InfoGAN}}(D,G,Q) = V(D,G) - \lambda L_I(G,Q)$

InfoGAN: CelebA

Skoltech
Skolkovo Institute of Science and Technology

We implemented the infoGAN. We changed the architecture of Generator, Discriminator and Decoder in order to increase the size of images, added our DataLoader to simplify the processing of CelebA dataset. Also, we have started with fewer number of latent codes — 2: the first one is discrete, the other — continuous [-1, 1].

The first attempt were unsuccessful, because we used too big sizes. As a result we get the following results (1). Then we decreased the size of the image to 64. Trained it on 202598 faces, 5 epochs and got better results (2).



(1)

(2)

InfoGAN: Tuning for ADNI

Skoltech
Skolkovo Institute of Science and Technology

Now, move on ADNI. Firstly, we have to filter MRI scans with *T1 modality* and write them into *npy file* using open source lib "*pydicom*". It helps us to construct our own dataloader. Another difference is to use *all axial slices* of ADNI dataset. The MRI images are not normalized. Each MR scan consists of pixel values in [0, ~2000] instead of standard [0,255] or [0,1]. Therefore we had to normalize each MR volume and then use it for training.

Next, we handled the issue connected with size of the scans. First attempts were made with scans scaled to 32x32 and after that with 128x128. So we change generator with discriminator CNN architecture and add flexibility for size of input latent vector. In order to use all information from hidden layers, we used Leaky ReLU in Generator and Discriminator both.

Used losses:

- Binary Cross Entropy (Discriminator)
- Cross Entropy (Decoder for discrete latent code)
- Log Gaussian (Decoder for continuous latent code)

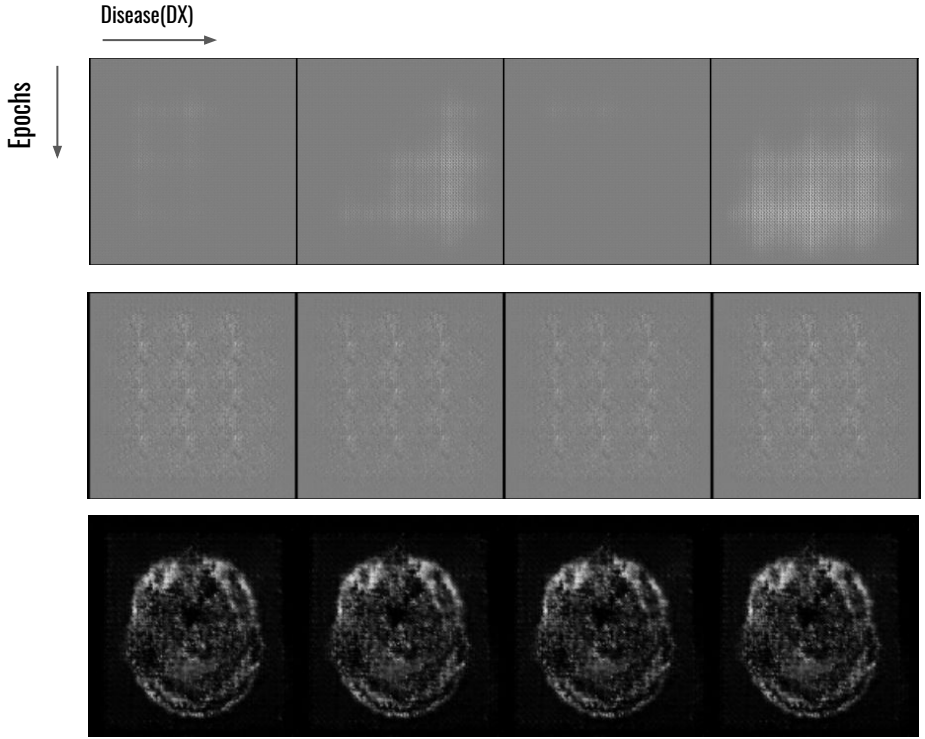
Features to interpolate:

- Slice
- Diagnosis (DX)

Training up to 200 epochs, batch size 500, dataset size ~13000, lr for generator and decoder 0.001, lr for discriminator - 0.0005.

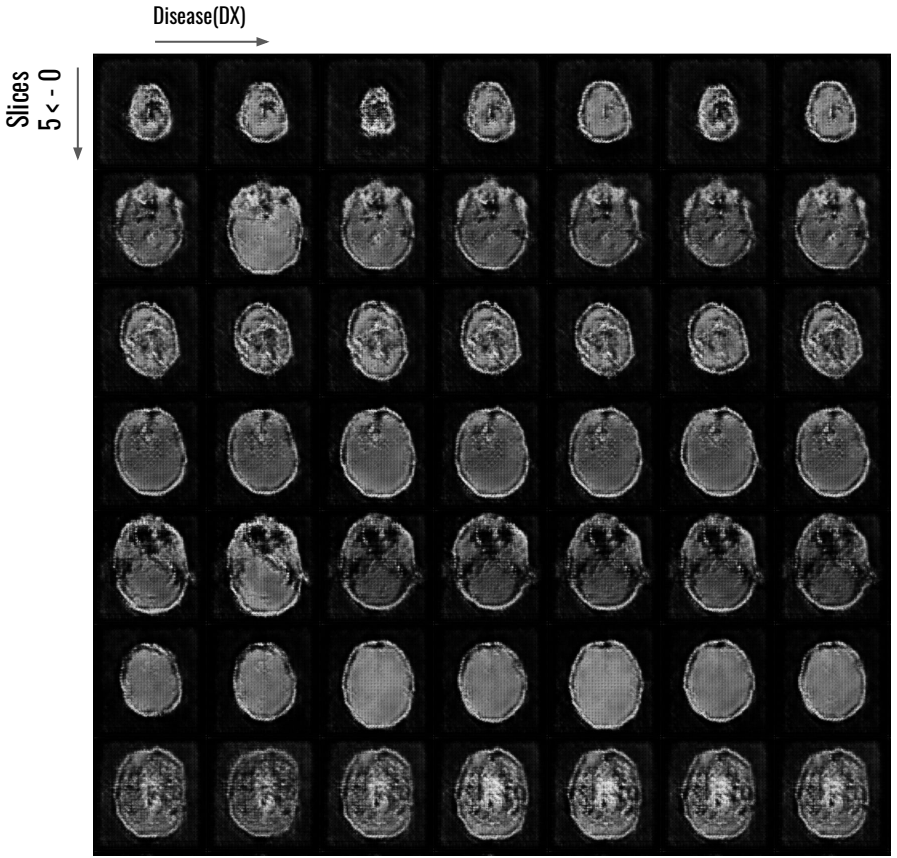
InfoGAN: Results on ADNI

- 128x128
- 1, 7, 23 Epochs
- Able to reproduce shape
- Poor quality of inner structures



InfoGAN: Results on ADNI

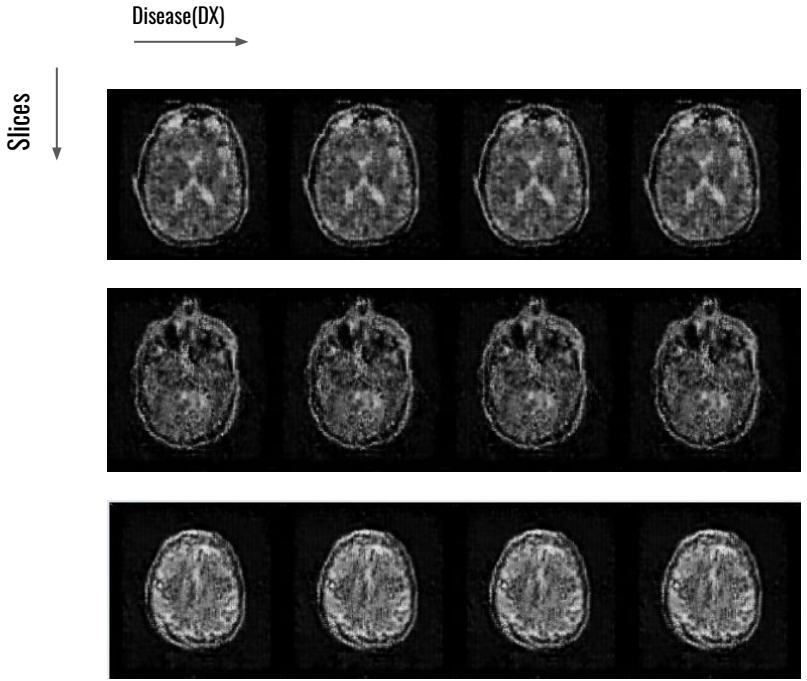
- 128x128
- 23 Epochs
- Able to reproduce shape
- Poor quality of inner structures



InfoGAN: Results on ADNI

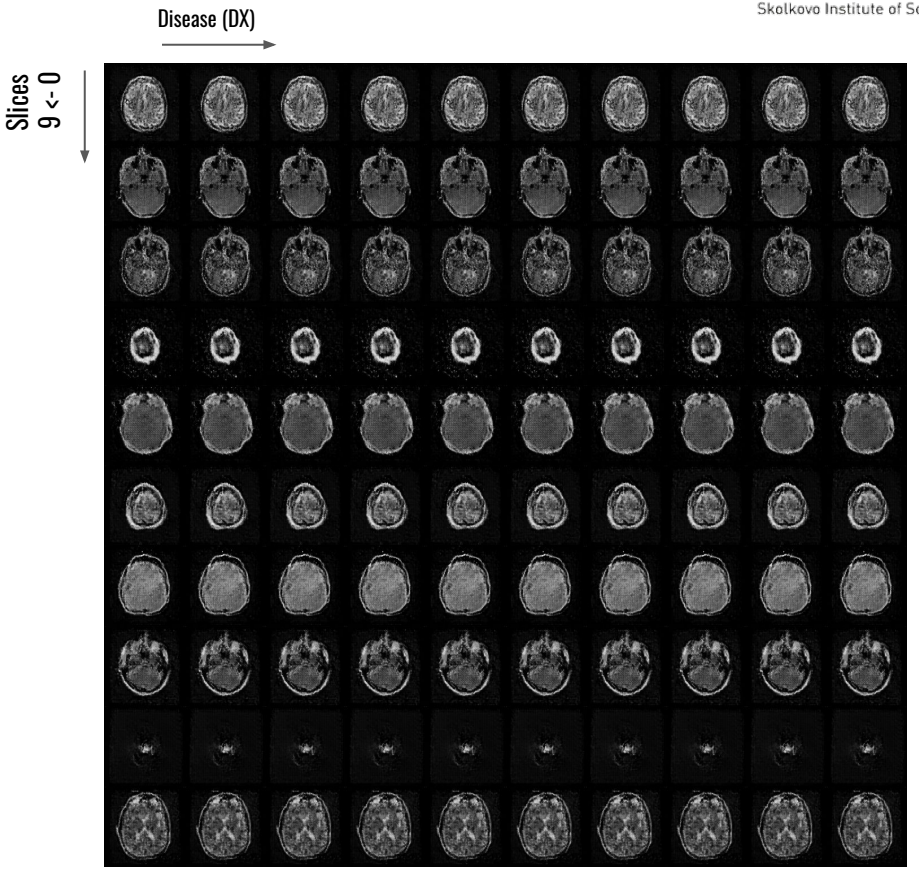
Skoltech
Skolkovo Institute of Science and Technology

- Able to reproduce shape
- Able to extract the slices with unique structure
- The structure inside is still blurred and contains artefacts (hallucinations), but coincides with slice number
- Therefore, it is hard to extract features connected with inner structure, i.e. disease
- ~100-150 Epochs



InfoGAN: Results on ADNI

- Able to reproduce shape
- Able to extract the slices with unique structure
- The structure inside is still blurred and contains artefacts (hallucinations), but coincides with slice number
- Therefore, it is hard to extract features connected with inner structure, i.e. disease
- ~100-150 Epochs

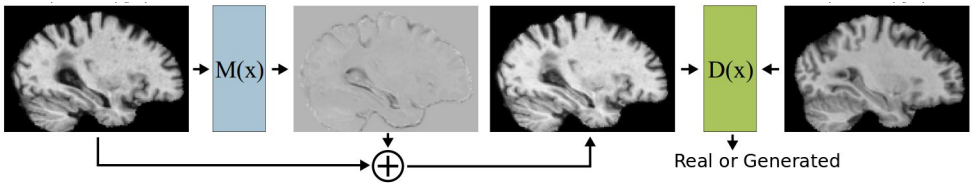


Visual Attribution GAN

Visual Attribution GAN

Skoltech
Skolkovo Institute of Science and Technology

Many of generative models that we tried struggle to reproduce details of brain shape and frequently, generated image appears to be just blurry and not-plausible. We thought then, why could not we build a model that does not aim to generate whole brain, but will be more focused on how we should change given image to make it own certain attributes. Indeed, for our task we don't need to generate a whole brain, it will remain the same either with or without required attributes, so let's try to generate specific changes of this brain while others its parts remain the same. From that perspective, Visual Attribution GAN [6] seemed to be suitable model to try.



$$\mathcal{L}_{GAN}(M, D) = \mathbb{E}_{x \sim p_d(x|c=0)}[D(x)] - \mathbb{E}_{x \sim p_d(x|c=1)}[D(x + M(x))].$$

- **WGAN-GP** with **UNet** type generator
- Generate the **mask** that moves an image to another category
- Originally **not** supposed to be **conditioned**

[6] Christian F. Baumgartner, Lisa M. Koch, et al. "Visual Feature Attribution using Wasserstein GANs." *arXiv preprint arXiv:1711.08998* (2017).

Visual Attribution GAN: Modifications & Implementation Details

Skoltech
Skolkovo Institute of Science and Technology

Originally VA-GAN was used to convert brains with MCI to brains with Alzheimer. During training generator was given MCI brains and discriminator was trained to distinguish MCI->Alzheimer generated images and Alzheimer images. Thus, generator was not conditioned and was trained for only one specific pair.

Since, there was no time for additional preprocessing and data preparation, we decided to directly condition generator with ADAS score and ask it to produce a mask, that given a brain image, will mimic an ADAS score when added to that image and we will use pre-trained regressor to predict ADAS for generated samples.

Thus, we a bit modified original VA-GAN and our objective can now be expressed as:

$$\mathcal{L}_{GAN}(M, D, \mathbb{C}) = \mathbb{E}_{x\sim p_r}\left[D(x)\right] - \mathbb{E}_{x,y\sim p_r}\left[D(x + M(x, y))\right] + \lambda \mathbb{E}_{x\sim p_r}\left[(\|\nabla_M D(x + M(x, y))\|_2 - 1)^2\right] + \mathbb{E}_{x,y\sim p_r}\left[(\mathbb{C}(x + M(x, \hat{y})) - \hat{y})^2\right]$$

*our objective

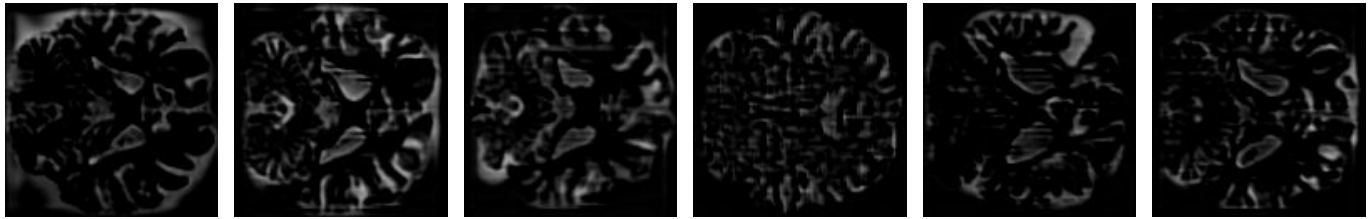
- We use pre-trained ADAS regressor (80,1% r2 score | 160 brains test dataset)
- We additionally penalize generator with Mean Squared Error of predicted ADAS for produced images and true ADAS scores that it was conditioned with
- Discriminator is used only for detecting fake images
- Gradient penalty is taken with respect to the mask added to an image

Visual Attribution GAN: Training

Skoltech
Skolkovo Institute of Science and Technology

- Train generator and discriminator as WGAN with 10*gradient penalty + use auxiliary loss, produced by pre-trained ADAS regressor.
- Training for 50 epochs with Adas optimizer with lr=0.001, betas=(0,0.9), as well as in **[6].**
- L1-regularization for generating minimalistic masks as we want to make as small changes as possible.
- Generator is conditioned by replicating conditioning attributes and concatenating them with given image
- While concatenate feature maps in an up-stream of our UNet generator, conditions are not concatenated
- Generator is conditioned with randomly permuted ADAS scores of batch samples

Masks during training:

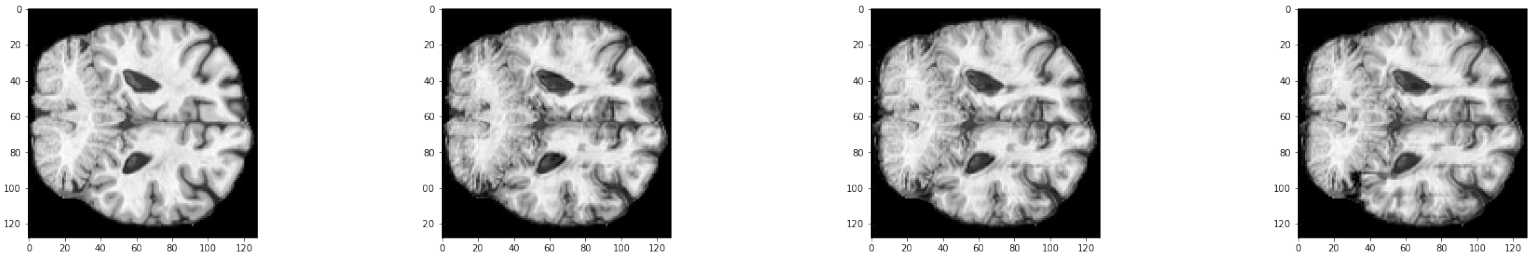


P.S. I will post the code on my github.com/ferumchrome as soon as I clean it and make it readable

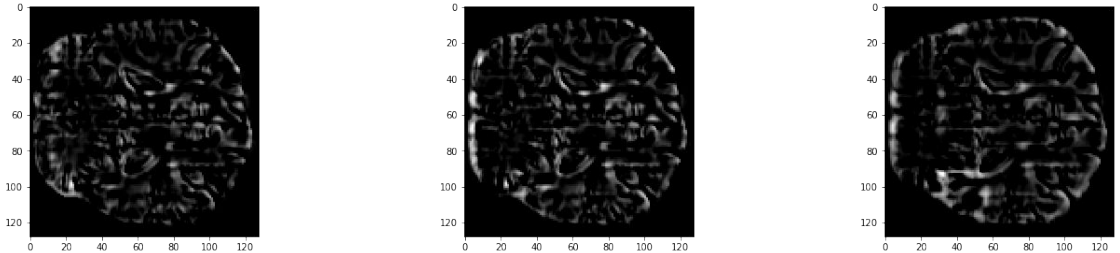
Visual Attribution GAN: Results on ADNI

Skoltech
Skolkovo Institute of Science and Technology

ADAS: **2** ADAS: **10** ADAS: **20** ADAS: **40**



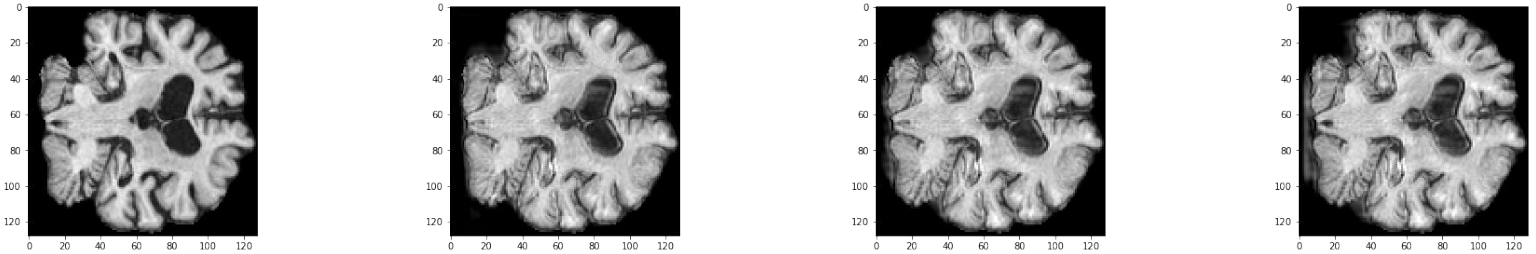
Difference:



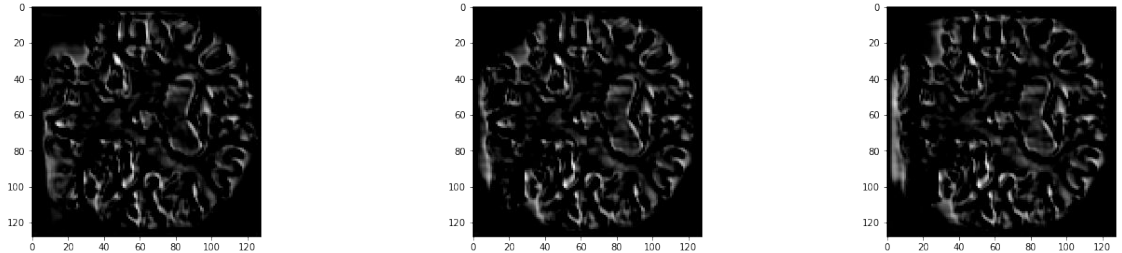
Visual Attribution GAN: Results on ADNI

Skoltech
Skolkovo Institute of Science and Technology

ADAS: **40** ADAS: **20** ADAS: **10** ADAS: **0**



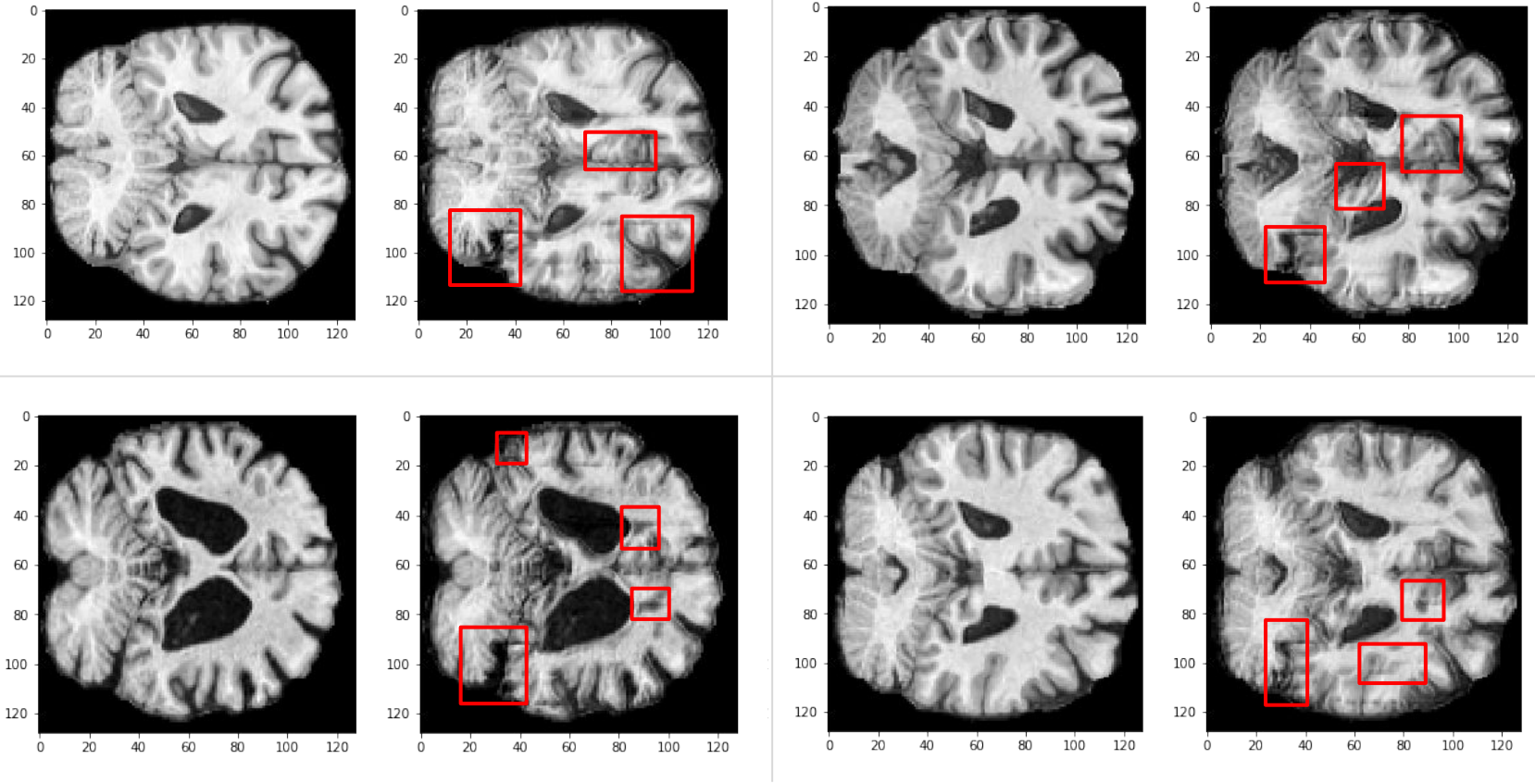
Difference:



Visual Attribution GAN: results on ADNI

Skoltech
Skolkovo Institute of Science and Technology

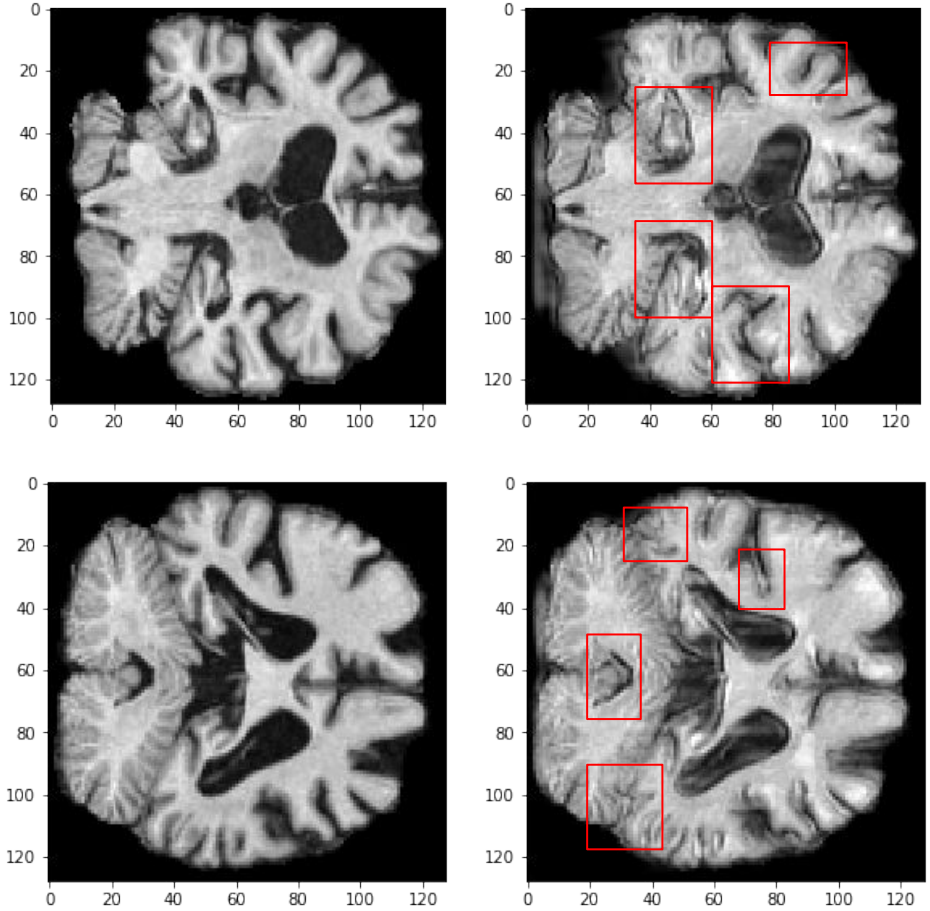
*healthy to injured



Visual Attribution GAN: results on ADNI

Skoltech
Skolkovo Institute of Science and Technology

*injured to healthy



Conclusion

- It was hard to evaluate models quantitatively on ADNI, since the dataset was relatively small, and it was difficult to train a well-performing predictor of age and diagnosis.
- While some of the models fail to yield diverse reconstructions that comply with prior knowledge about the disease, some of them, such as Fader Networks and Visual Attribution GAN, serve a decent proof of concept.
- It is suspected that using 3D as an input domain for all methods would produce much better results, as slices are subject to motion and misalignment. However, using 3D MRI images as input data is very computationally demanding for the most methods.

	Subjective rating on CelebA	Subjective rating on ADNI
IcGAN		
Deep Feature Interpolation		
Fader Networks		
InfoGAN		
Visual Attribution GAN		

Contribution

Skoltech
Skolkovo Institute of Science and Technology

Artem Moskalev: Data preprocessing. IcGAN for CelebA and ADNI, Visual Attribution GAN: implementation, tuning, conducting experiments (slides 4-13, 49-56). Presentations, final report.
Artur Grigorev: Tuning Fader Networks for the task and conducting experiments with this architecture (slides 23-37). Presentations, final report.
Artem Sevastopolskiy: Deep Feature Interpolation - implementation from scratch, evaluation of various decoding techniques (both from the original paper and devised ones). Presentations, final report. (slides 2-3, 14-22)
Denis Prokopenko: Data preprocessing. InfoGAN for CelebA and ADNI: implementation, tuning, carrying out experiments. Presentations, final report. (slides 38-48)

Used code

- Fader networks implementation: https://github.com/facebookresearch/FaderNetworks
- InfoGAN implementation: https://github.com/pianomania/infoGAN-pytorch
- Elements of code from https://github.com/znxlwm/pytorch-MNIST-CelebA-GAN-DCGAN for IcGAN implementation.

References

[1] Upchurch, Gardner et al. "Deep Feature Interpolation for Image Content Changes." CVPR 2017.
[2] Perarnau, Weijer, et al. "Invertible Conditional GANs for image editing." NIPS 2016 Workshop on Adversarial Training. 2016.
[3] Lample, Guillaume, et al. "Fader networks: Manipulating images by sliding attributes." Advances in Neural Information Processing Systems. 2017.
[4] Chen, Xi, et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets." Advances in Neural Information Processing Systems. 2016.
[5] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." arXiv preprint arXiv:1703.10593 (2017).
[6] Christian F. Baumgartner, Lisa M. Koch, et al. "Visual Feature Attribution using Wasserstein GANs." arXiv preprint arXiv:1711.08998 (2017).
[7] S. Reed, Z. Akata, et al. "Generative Adversarial Text to Image Synthesis." arXiv preprint arXiv:1605.05396 (2016).