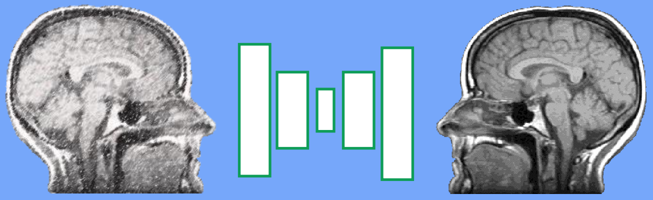


Deep Convolutional Autoencoders for reconstructing magnetic resonance images of the healthy brain

Adrián Arnaiz Rodríguez
Student

Dr. Baris Kanber
Director

Universitat Oberta de Catalunya

Master's in Data Science
Master's Thesis
Medicine Area

January 2021

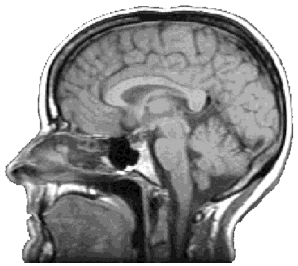
aarnaizr@uoc.edu

[AdrianArnaiz/Brain-MRI-Autoencoder]

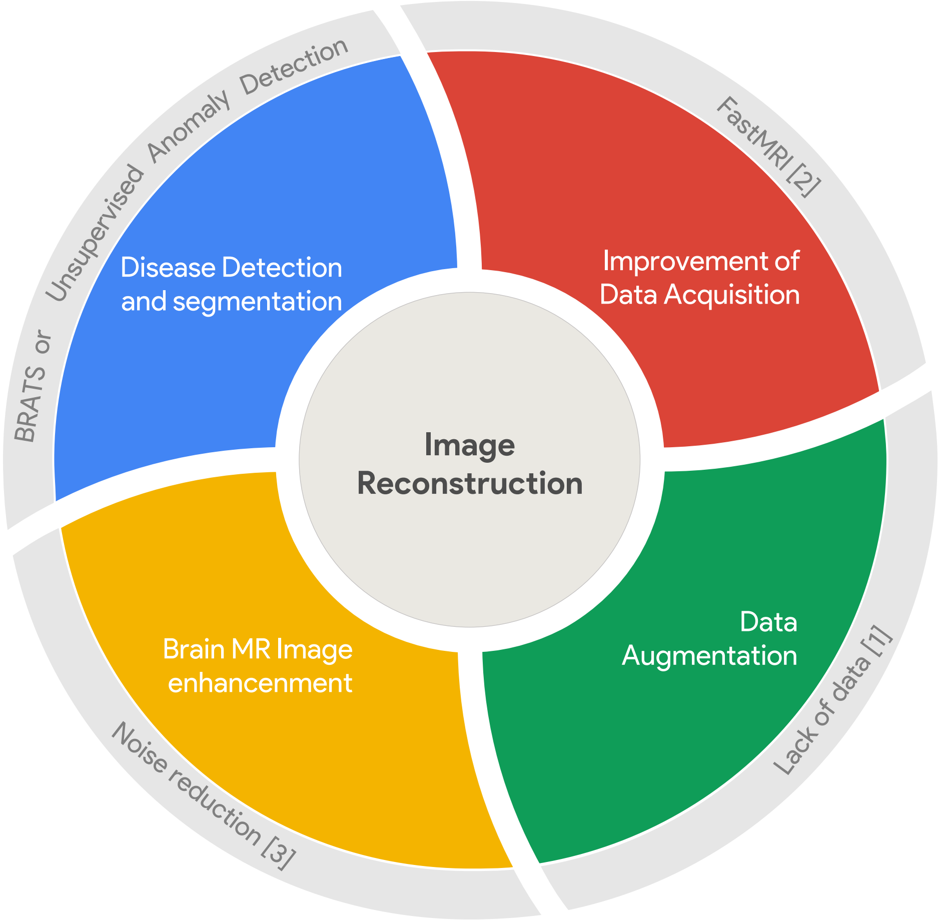
Content

Introduction: Brain MRI and Neuroimaging

Bran Magnetic Resonance Images: brain images obtained through a magnetic resonance.



Neuroimaging: study of **morphological features** of the human brain and its correlations with neurological disorders to **improve medical systems.**



Relevance: **cutting edge** projects in neuroimaging research areas.

Overlapped objectives: advance in some of them leads to the advance in another.

Image reconstruction through **information compression techniques** helps every other problem.

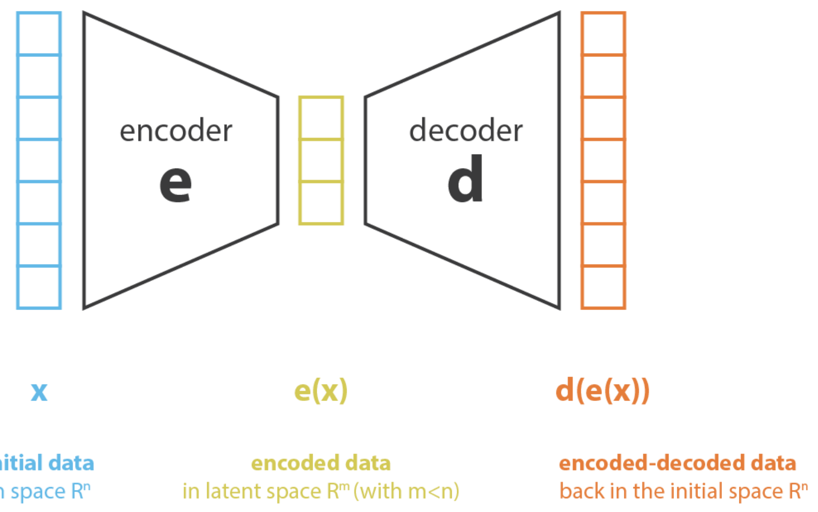
[1] Changhee Han, et. Al. Infinite brain tumor images: Can gan-based data augmentation improve tumor detection on mr images?, 2018.
[2] Florian Knoll, et al. FastMRI: A publicly available raw k-space and dicom dataset of knee images for accelerated mr image reconstruction using machine learning, 2020.
[3] Daiki Tamada. Noise and artifact reduction for mri using deep learning, 2020

Main approach: Reconstruction through lower dimension distribution

Approach: capture the representation of the underlying structure of healthy brains in a lower-dimensional space and reconstruct from this under-sampled data.

ML Topics: Representation learning, dimensionality reduction, Normative model, information compression, representation in latent space, lower dimension distribution...

Convolutional Auto Encoders



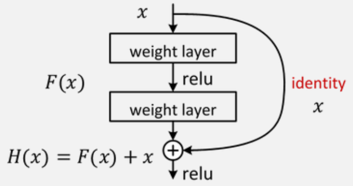
- Train with disease-free neuroimaging data
- This autoencoder, would define a normal range (or distribution) for the neuroanatomical variability for the illness absence.
- It learns the brain structure representation.
- Once trained, it encodes an input image and reconstructs it without corruptions like noise, artifacts or lessons.

State Of The Art: Related works

Architectures

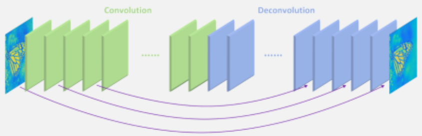
Residual Networks

- Mapping image to target: *H(x)*
- Approximate the residual of this function: *F(x) = H(x) – x*
- Get the original function: *F(x) + x*
- Different kinds of building blocks
- ✓ Addresses vanishing gradient problem
- ✓ Less complexity by learning residuals



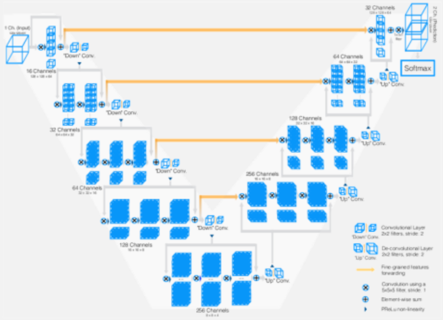
Skip Connection CAEs

- Add or concatenate values from encoder to decoder layers
- ✓ Addresses vanishing gradient problem
- ✓ Better low-detail reconstruction



U-Net and V-Net

- 3D convolutional networks
- Symmetric skip connections with concatenation.
- V-NET adds residual building blocks.



Loss Functions

- Mean Square Error (*MSE*)
- Structural Dissimilarity (*DSSIM*)
- Peak signal to noise ratio (*PSNR*)
- 3D: Dice Loss
- VAE: KL Divergence

MRI Preprocessing

Intensity Normalization

Range [0,1]

Mean 0 and std 1

Downsampling

Reduce the dimension of the input data

Contrast and Bright enhancement

Enhance the image quality before using it as input and output

Could lead to destroying some important clinical aspects despite it looks better

Data Augmentation

Random input data modifications to improve model performance

- Noise
- Rotations
- ...

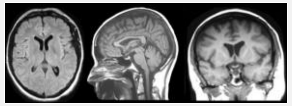
Profile and Relevant Slide Selection

Profile Selection

Most projects works with volumes: 3D

Non-isotropic MRI:

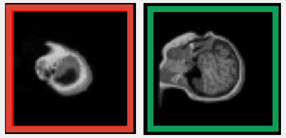
- Use dimension that make slices are high resolution



Relevant Slide Selection

Select slices with relevant information, with some amount of brain

- Middle Slice [3]
- Fixed Range



Applications

FastMRI, 2020 [7]

- Facebook AI and New York University Project
- Accelerate MRI scan process generating MRI from under-sampled data. Invited to NeurIPS

BRATS: Myronenko, 2018 [1]

Residual Variational Autoencoder to regularize residual brain tumor segmentation network.

BRATS: T. Estienne et. al., 2020 [6]

V-NET autoencoder for tumor segmentation and image registration

Yu et. al., 2019 [4]

Residual CAE+STM to learn volumetric shape representations for brain structures

C. Bermudez et. Al., 2018 [3]

Skip connection CAE for denoising and learn brain manifolds

J. Manjón et. al., 2020 [2]

UNet based network for lesson inpainting for improve of brain analysis pipelines

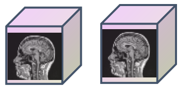
W. Pinaya et. al., 2019 [5]

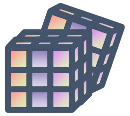
CAE for unsupervised anomaly detection with normative model

[1] Andriy Myronenko. 3d MRI brain tumor segmentation using autoencoder regularization. 2018. | [2] José V. Manjon et. al. Blind MRI brain lesion inpainting using deep learning. 2020. | [3] Camilo Bermudez et. al. Learning implicit brain MRI manifolds with deep learning. 2018. | [4] E. M. Yu et. al. A convolutional autoencoder approach to learn volumetric shape representations for brain structures. 2019. | [5] Walter HL Pinaya et. al. Using deep autoencoders to identify abnormal brain structural patterns in neuropsychiatric disorders: A large-scale multi-sample study. 2019. | [6] Theo Estienne et. al. Deep learning-based concurrent brain registration and tumor segmentation. 14, 2020. [7] Florian Knoll, et al. FastMRI: A publicly available raw k-space and dicom dataset of knee images for accelerated mr image reconstruction using machine learning, 2020.

Development: Overview of Experiment Pipeline

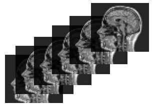
All Original IXI Volumes





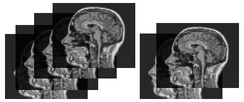
Slice extraction and selection

Bunch of slices with relevant information



Train – Test Split and Normalization

Data Splited



We have all the original IXI brain T1 weighted MRI volumes

- Profile Selection
Select profile which give us high-definition images: SAGITTAL
- Orientation checking
Check all the volumes have the same orientation
- Slice extraction and selection
Select slices with relevant information:
✓ DeepBrain: Brain segmentation

We store in a folder all jpg files of selected MRI images

- Data split
80-20 train/validation and test
Stratified by sex, age and ethnic.
- Preprocessing
We normalize the pixels values in pictures on range [0-1]

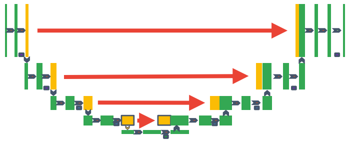
We downsample the images to 128x128
Both Will be made on the fly

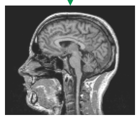
Stored in separate folders.

These images are our input and also target images.

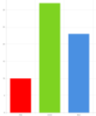
Experiments

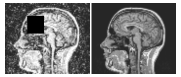
Data augmentation





Results





Quantitative and Qualitative evaluation

- Training

- Data Augmentation
- Blank-out regions
- Noise
  - Gaussian
  - Dropout pixels
- Blur images

Random level of each Augmentation in every picture

- Architectures
CAE based on:
  - Resnet-based building blocks
  - Cae + skip-connections
  - Resnet + Skip connections (V-Net based)
- Loss functions:
  - MSE
  - DSSIM
- Regularization:
  - L2
  - Batch Normalization

- Get metrics on test set
Compare models quantitatively and qualitatively

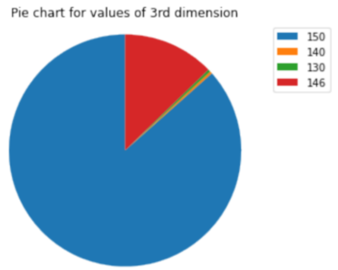
Development: Dataset Exploration

- IXI BRAIN T1- Weighted Dataset from Imperial College of London

General Data

IXI BRAIN T1- Weighted Dataset from Imperial College of London

- 584 3D MRI volumes → Nifti Format
- Voxel dimension: 1^{st} and 2^{nd} dimension → Isotropic
  - 0.9375 × 0.9375 × 1.2mm3 → 576 volumes
  - 0.9766 × 0.9766 × 1.2mm3 → 5 volumes
- Volume dimension (in voxels):
  - 256 x 256 x [150, 146, 140, 130]



Profile Selection

- Slices must be high resolution: 1x1 mm3 rather than 1x2 mm3
- The 1st and 2nd dimension → Isotropic for all volumes
- We must freeze 3^{rd} dimension to get our isotropic 2d images
- Sagittal view

Orientation checking

- Obtain the same brain slice view for all patients
- Same orientation for every volume: P, S, R.
  - 1st voxel axis goes from anterior to Posterior,
  - 2nd voxel axis goes from inferior to Superior
  - 3rd voxel axis goes from left to Right

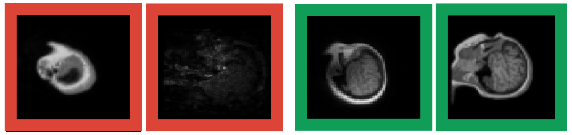
Total

86794

Brain Sagittal 2D images

Development: Relevant slices selection I

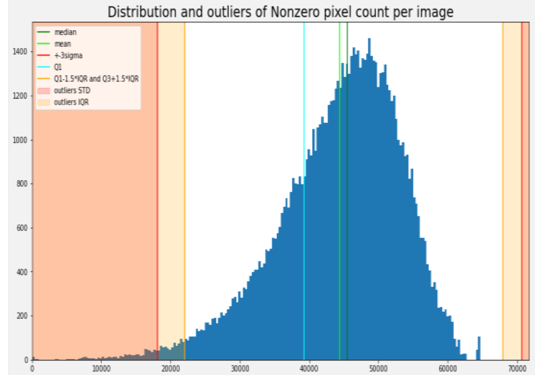
- Not every slice has relevant information
- Middle slice and fixed middle range methods are inefficient approaches

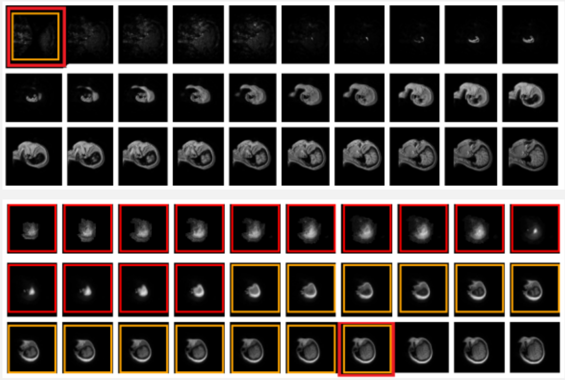


info = 

Non-zero intensity pixel count

- Count pixels with intensity distinct from 0
- Discard slices which are low-outliers in the distribution
  - Low-outliers → Few pixels non-black → no relevant information
  - Thresholds based in outliers : -3std, Q1-1.5*IQR and Q1.

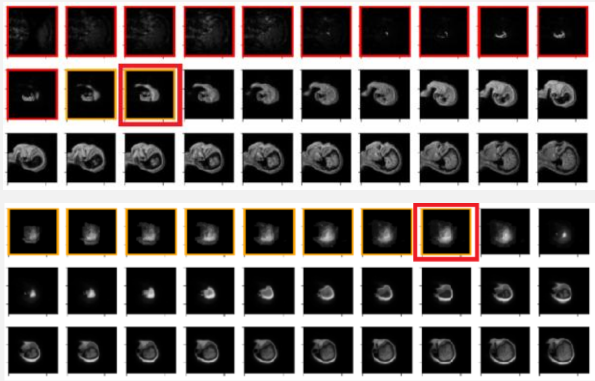


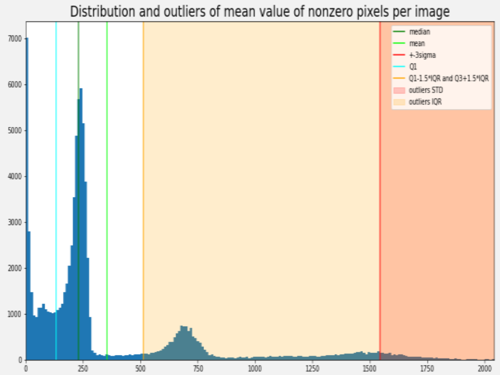


- Not as good as expected
- Weak against noisy points and non-brain structures

Mean of intensity of non-zero pixels

- Compute the mean of pixels with intensity distinct from 0
- Discard slices which are low-outliers in the distribution
  - Low-outliers → Low mean intensity → no relevant information
  - Thresholds based in observations and try and fail.



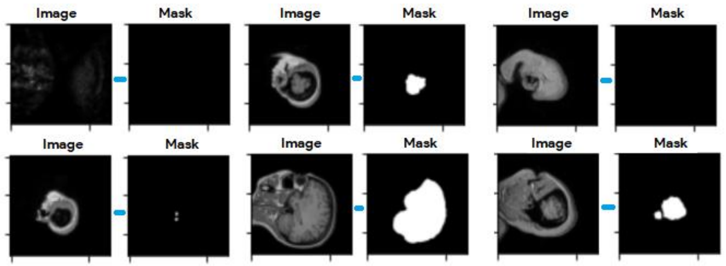


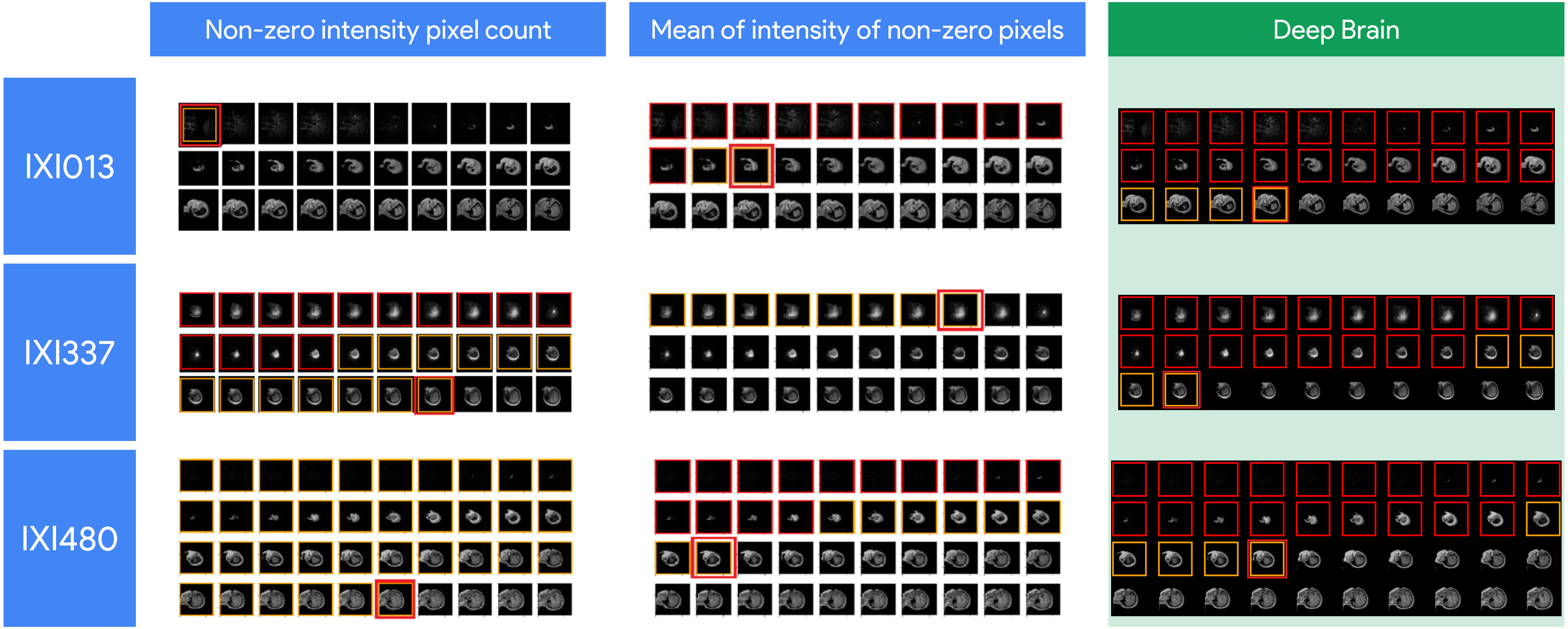
- Not as good as expected
- Weak against noisy shiny points and strange structures

Development: Relevant slices selection II

Our approach: DeepBrain [1] → Pretrained CNN for brain segmentation

- Return mask of probabilities of each pixel belonging to brain
- Low-Threshold: 4.5% of brain pixels
- Final relevant slices: 59278

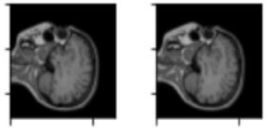




[1] Deep Brain, Ivan Itzcovich, 2018, https://github.com/iitzco/deepbrain

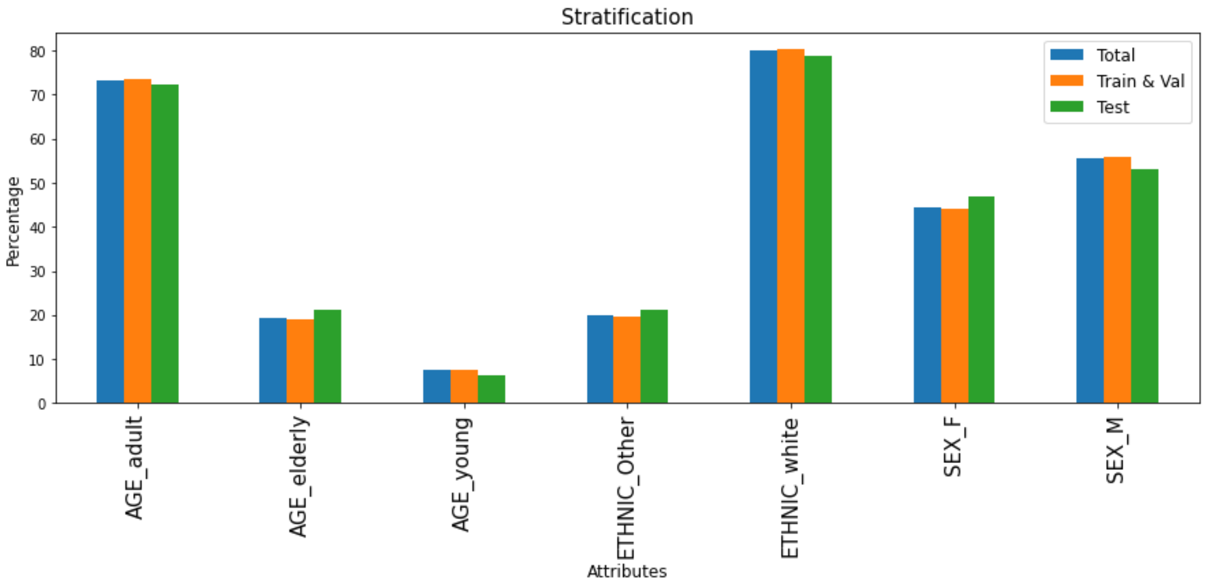
Development: Split data

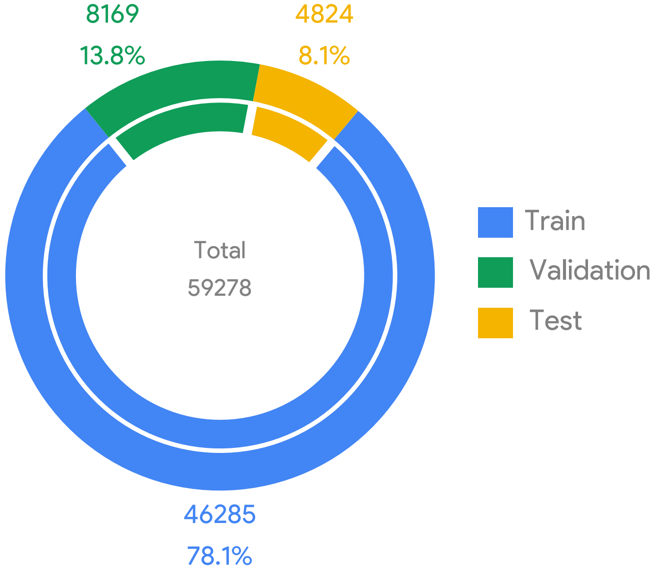
- i.i.d: Data must be independent and identically distributed between Train, Validation and Test sets
- Independence: avoid big correlations
  - Close images from the same volume → Almost equal → Should not be in different splits
  - Solution: Split straight from volumes



Slices 67 and 68

- Identically distributed: Stratify volumes by demographic features
  - Stratify by AGE Group, SEX and ETHINIC





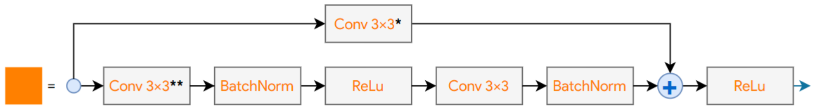
Development: Architectures I

Design Guidelines:

- Make Shallow models → IoT, mobile and remote places
- Compare
  - Skip-Connection archt. VS Residuals archt. VS its combination
  - MSE VS DSSIM loss functions
  - L2 Regularization VS No regularization
- Do not use neither max pooling nor classical upsampling [1]
  - Discards relevant features
- Use stride=2 in convolutions and transpose convolutions

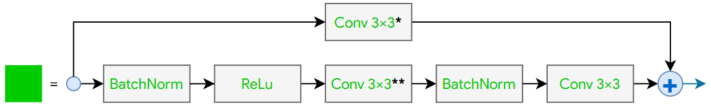
Building Blocks:

ORIGINAL RESIDUAL ENCODER BUILDING BLOCK



** Only added if necessary to match filters and dimensions*
*** stride=2 if downsampling*

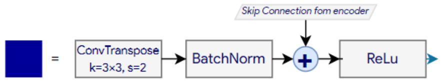
FULL PRE-ACTIVATION RESIDUAL ENCODER BUILDING BLOCK



** Only added if necessary to match filters and dimensions*
*** stride=2 if downsampling*

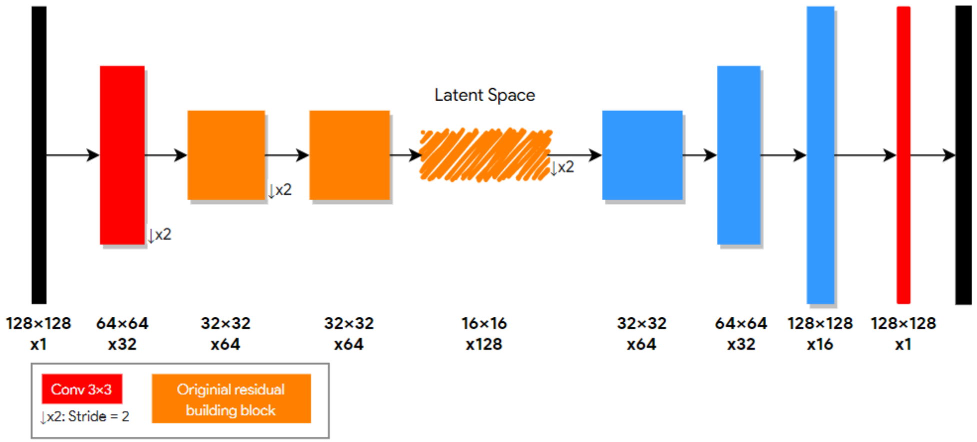
UPSAMPLING DECODER BUILDING BLOCK

UPSAMPLING DECODER BUILDING BLOCK IN FCN+SKIP CONNECTIONS

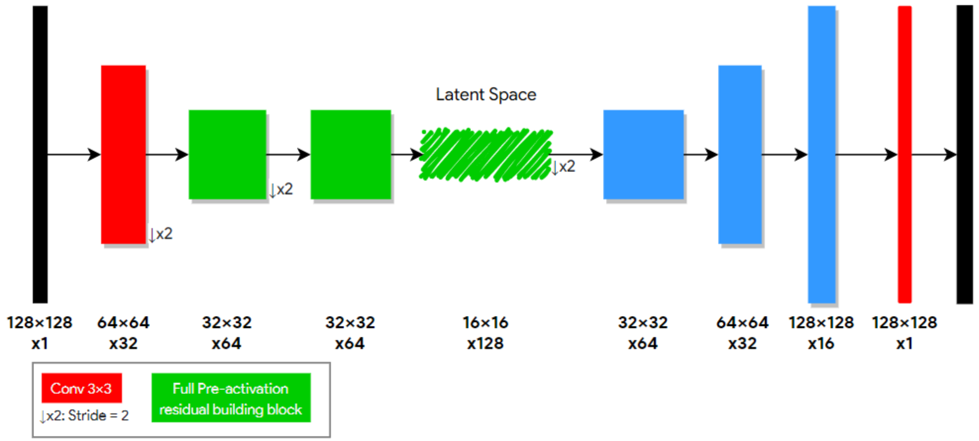


Architectures

SHALLOW RESIDUAL AUTOENCODER (ORIGINAL BLOCK)

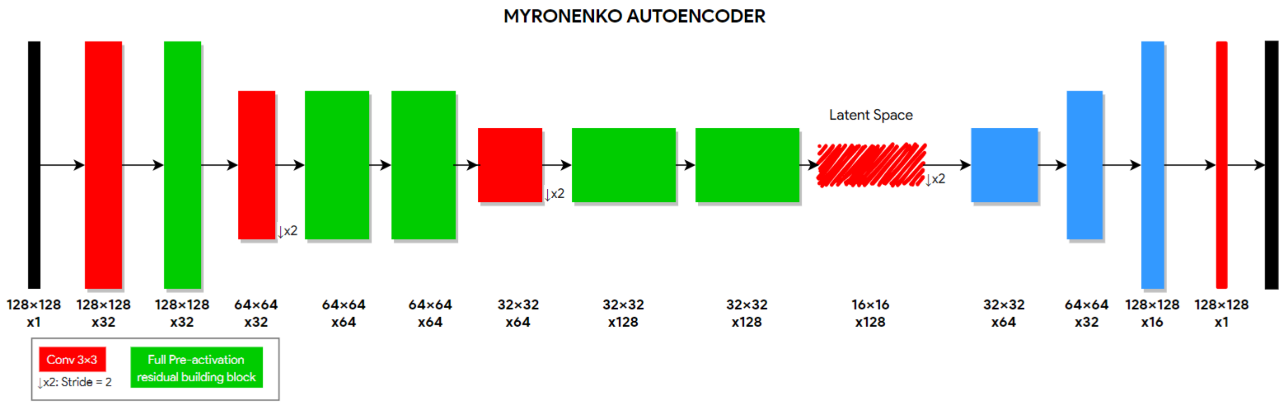


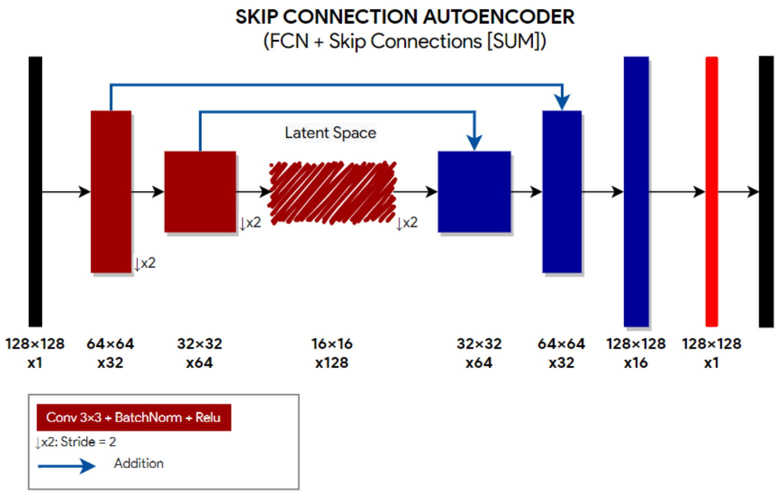
SHALLOW RESIDUAL AUTOENCODER (FULL PRE-ACTIVATION BLOCK)

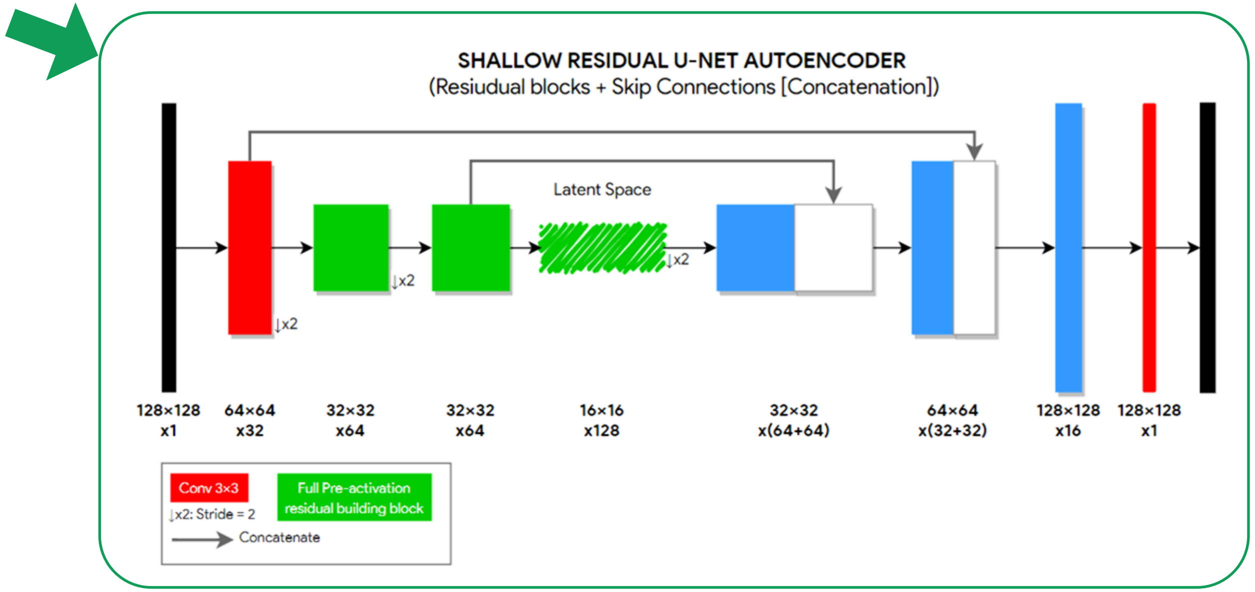


[1] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using convolutional autoencoders with symmetric skip connections. 2016.

Development: Architectures II







Architectures summary:

- Shallow residual autoencoder:
  - Original building block
  - Full-pre-activation building block
- Skip connection convolutional autoencoder
- Myronenko Autoencoder [1]
- Residual U-NET: Proposed one
  - Combination of U-NET-based and residual building blocks
  - V-Net-based in 2D

[1] Andriy Myronenko. 3d MRI brain tumor segmentation using autoencoder regularization. 2018.

Development: Environment and data generator

Environment

On Device

- HP Omen i7-9H 2.6G
- 16GB RAM
- NVIDIA GeForce GTX 1660 Ti: 6GB of GDDR6 RVAM

Coding Environment

- Python 3.7.9
- Tensorflow-gpu 2.3.1
- TF Bulit-in Keras 2.4.0
- CUDA 10.1
- CuDNN 7.6.5
- Miniconda with Jupyter notebooks
- Visual Studio Code
- Github and Zenhub

Why on premise?

↓

Address the software speed optimization problem

Data Loader/Generator

Customized Tensorflow Data Loader

- Tf.data
- 4 times faster than classical Keras method *ImageDataGenerator.flow_from_directory*

Data preprocessing and augmentation ON THE FLY

- While the model is training

Optimized through PARALELLISM and PREFETCHING data in memory

- Read images, preprocess, augment and prefetch in a parallel way

Everything made with Native Tensorflow functions

- Improves performance
    - Faster load, preprocess, augment, prefetch and parallelize
- *Tf.Tensors*
- *Tf.Dataset* (prefetch, batch map...)
- *Tf.math*
- *Tf.image*
- *Tensorflow-addons* (data aug)
- ...

MRI Preprocessing

Downsampling

- 128x128

Normalization

- Range [0-1]
- Activation function of last layer

↓

Sigmoid

Data Augmentation

Dropout pixels

- Random level [0 - 5] %

Gaussian Noise

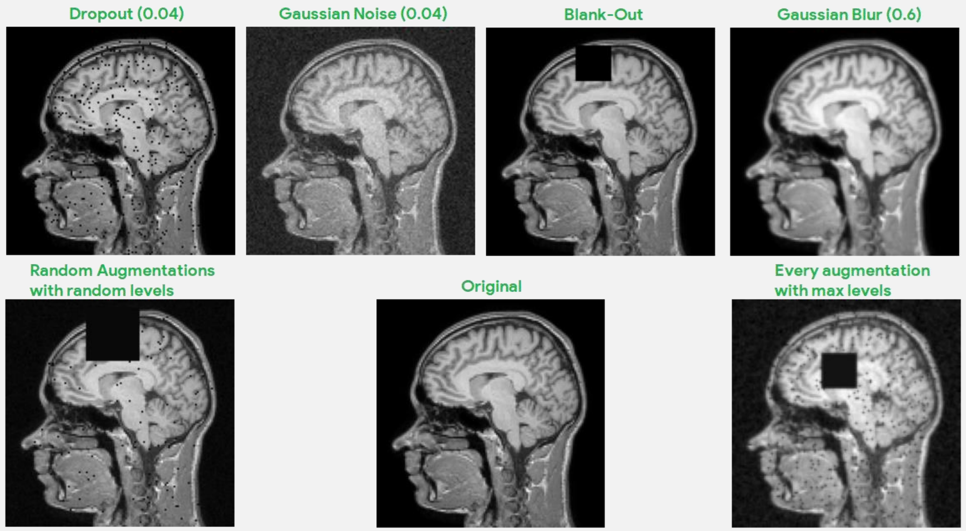
- Random level [0 - 0,04] STD

Gaussian Blur

- σ = 0.6, p = 0.1

Blank-out Region

- Random position
- Random size
- p = 0.1

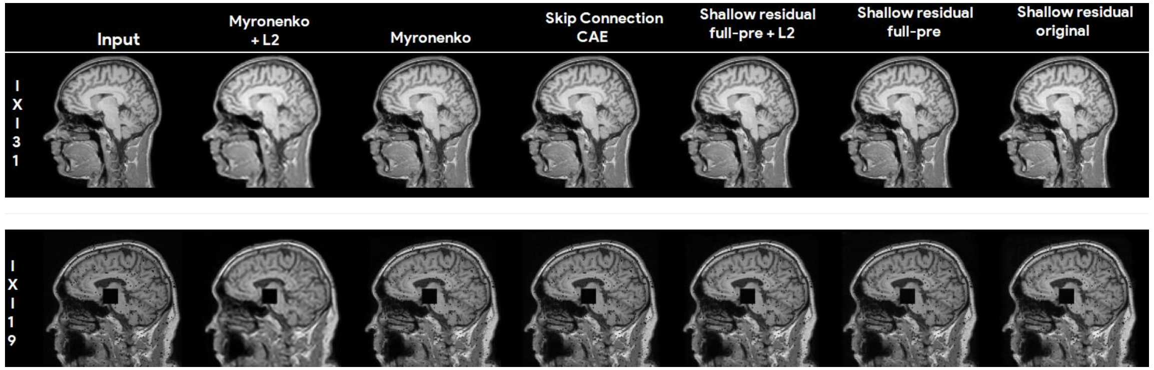


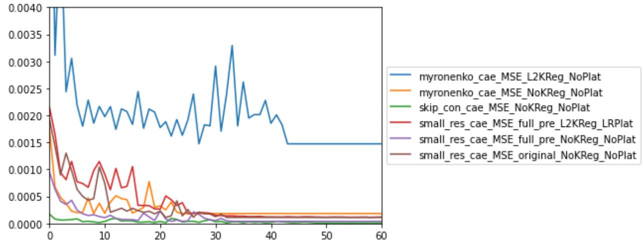
Development: Experiments Without Data Augmentation

Train Parameters	
Batch	
Size	32
Epochs	
Max	100
RMSProp Optimizer	
LR	1e-3
epsilon	1e-7
Early Stopping	
Patience	20
Min delta	*MSE* 2e-7
	DSSIM 5e-5
Reducer LR on Plateau	
Factor	0.2
Patience	4
L2 Regularizer (if used)	
Value	1e-5

** Experiments without data augmentation has been done for watching how models overfit and for checking the code of scripts and Python Classes.*

Model	loss	L2	Val loss	MSE	DSSIM	PSNR
Skip connection CAE	MSE	No	1.10e-5	1.07e-05	1.04e-03	49.8
Shallow RES full-pre	MSE	No	3.92e-5	3.82e-05	2.64e-03	44.4
Shallow RES full-pre	MSE	Yes	1.10e-4	8.56e-05	4.90e-03	41.0
Shallow RES orignial	MSE	No	1.11e-4	1.09e-04	1.40e-02	39.7
Myronenko CAE	MSE	No	1.83e-4	1.81e-04	5.59e-03	37.7
Myronenko CAE	MSE	Yes	1.47e-3	1.27e-03	4.40e-02	29.3





- Models trained
  - Skip connection CAE
  - Shallow RES full-pre
  - Shallow RES full-pre +L2
  - Shallow RES original
  - Myronenko CAE
  - Myronenko CAE + L2
- Loss Functions
  - MSE Pixel-wise: $\frac{1}{N}\sum_{i=n}^{n}\left(Y_i - \hat{Y}_i\right)^2$
- Test Metrics
  - MSE
  - DSSIM: $\frac{1 - SSIM(x,y)}{2}$

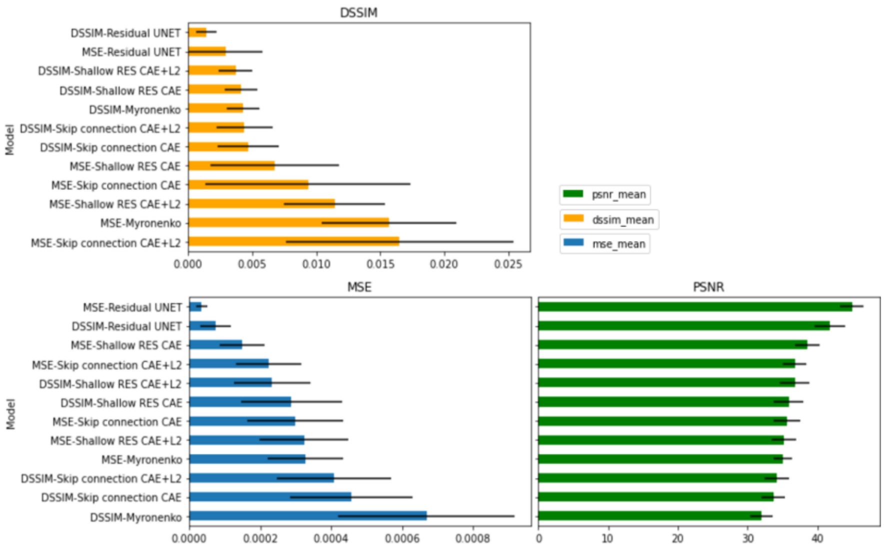
$$^{*}SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

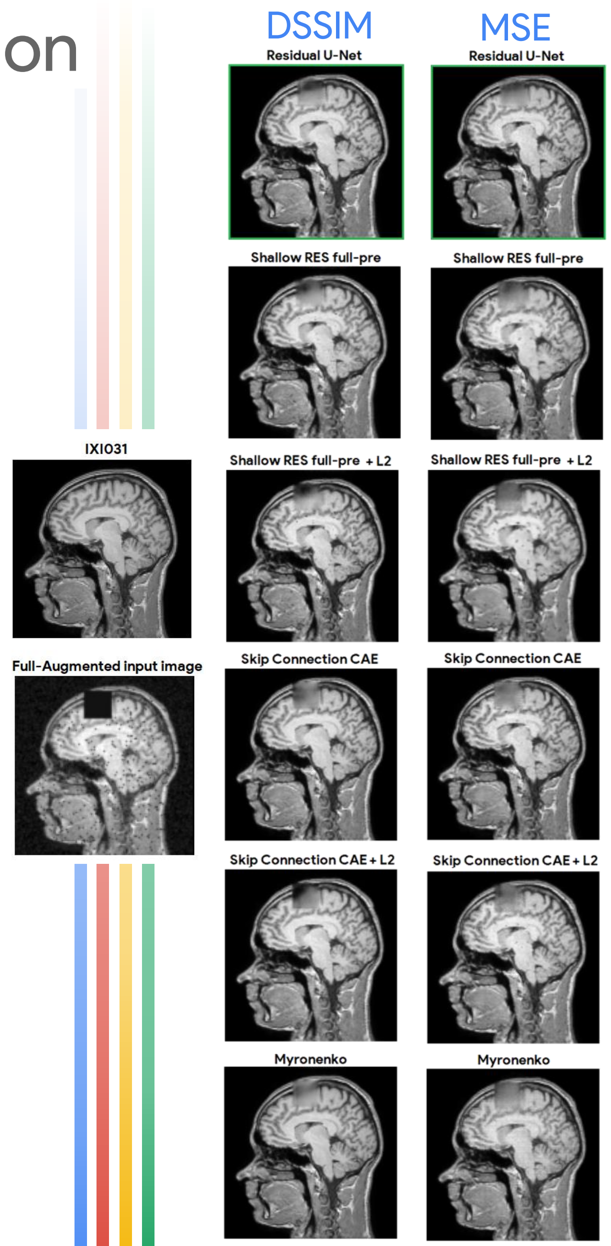
- PSNR: $10 \text{ x } \log_{10} \frac{\text{peakval}^2}{\text{MSE(x, y)}}$

Development: Experiments With Data Augmentation

- Models trained
  - **Residual U-NET** (Proposed)
  - Shallow RES full-pre
  - Shallow RES full-pre +L2
  - Skip connection CAE
  - Skip connection CAE + L2
  - Myronenko CAE
- Loss Functions
  - MSE
  - DSSIM
  - Every architecture with each loss: 12 final models
- Test Metrics
  - MSE
  - DSSIM
  - PSNR
- Train Parameters
  - Same

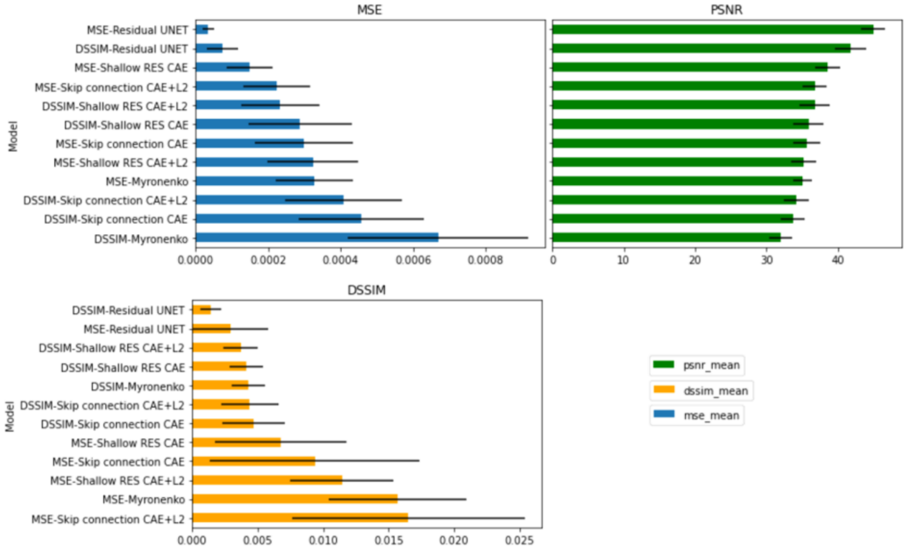
Model	loss	L2	Val loss	MSE	DSSIM	PSNR
Residual U-NET	MSE	No	**3.58e-05**	3.44e-05	**2.95e-03**	**44.9**
Shallow RES full-pre	MSE	No	1.55e-04	**1.51e-04**	6.75e-03	38.6
Skip connection CAE	MSE	Yes	2.69e-04	**2.25e-04**	1.65e-02	36.8
Skip connection CAE	MSE	No	3.10e-04	**2.99e-04**	9.36e-03	35.7
Myronenko CAE	MSE	No	3.38e-04	**3.27e-04**	1.57e-02	35.1
Shallow RES full-pre	MSE	Yes	3.72e-04	**3.24e-04**	1.14e-02	35.2
Residual U-NET	DSSIM	No	**1.50e-03**	**7.49e-05**	1.44e-03	**41.8**
Shallow RES full-pre	DSSIM	Yes	4.42e-03	2.34e-04	**3.70e-03**	36.7
Shallow RES full-pre	DSSIM	No	4.19e-03	2.88e-04	**4.14e-03**	35.9
Myronenko CAE	DSSIM	No	4.39e-03	6.69e-04	**4.31e-03**	32.1
Skip connection CAE	DSSIM	Yes	4.82e-03	4.08e-04	**4.38e-03**	34.2
Skip connection CAE	DSSIM	No	4.90e-03	4.57e-04	**4.71e-03**	33.7



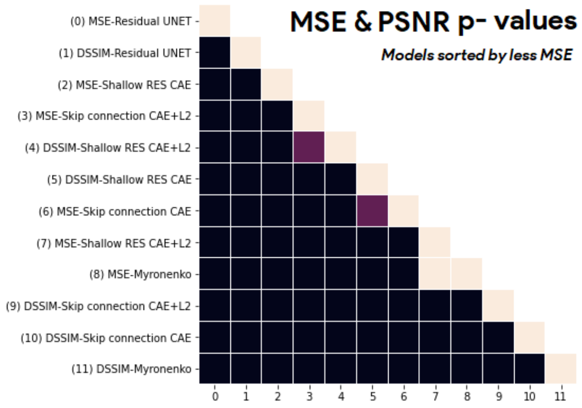


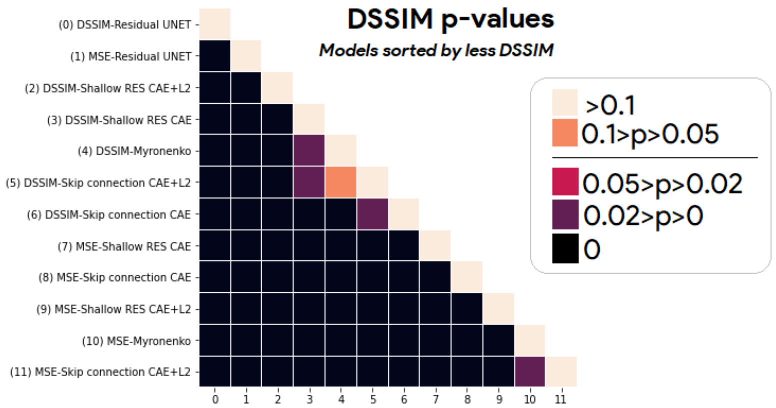
Results: Quantitative

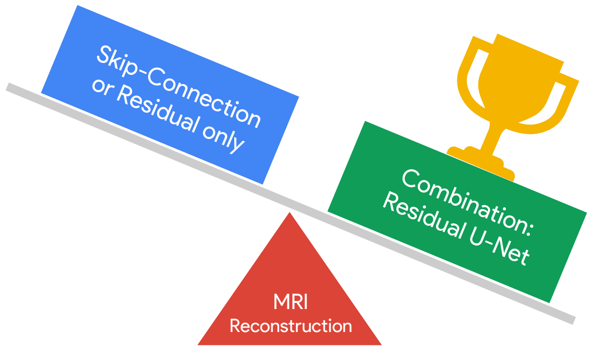
- Residual U-Net shows best results both for MSE and DSSIM Loss
- Shallow Residual full-pre follows in both
- L2 regularization improves model when DSSIM is used as loss
  - But L2 decreases performance when loss is MSE
- Results with statistical significance: Dependent t-test



$$t = \frac{MSE_1 - MSE_2}{\sqrt{\frac{S_D}{\sqrt{N}}}} \qquad S_D = \sqrt{\frac{\sum_i^{N_{test}}(MSE_{1_i} - MSE_{2_i}) - \frac{\sum_i^{N_{test}}(MSE_{1_i} - MSE_{2_i})^2}{N_{test}}}{N_{test} - 1}}$$

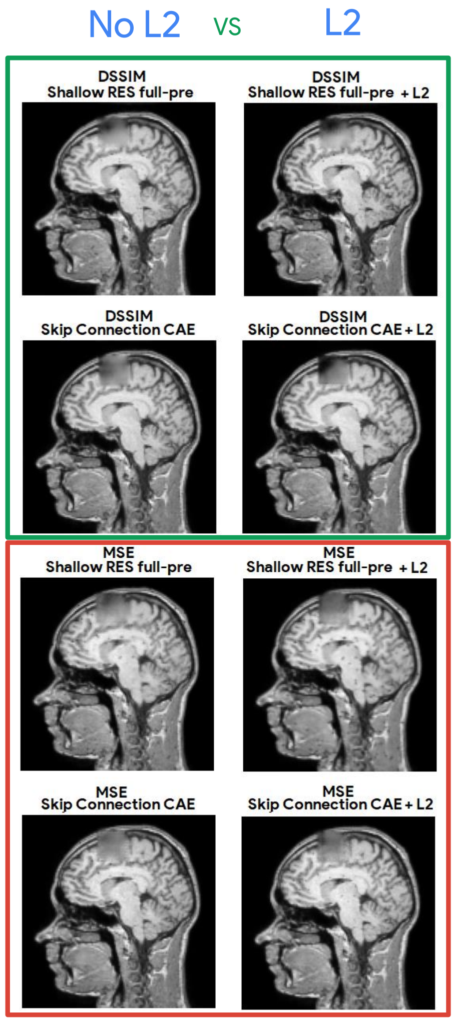




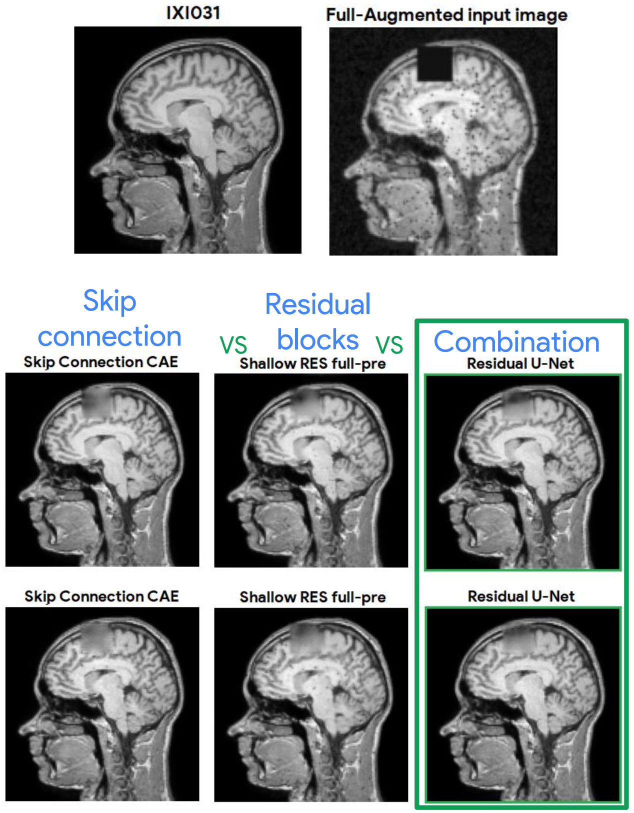


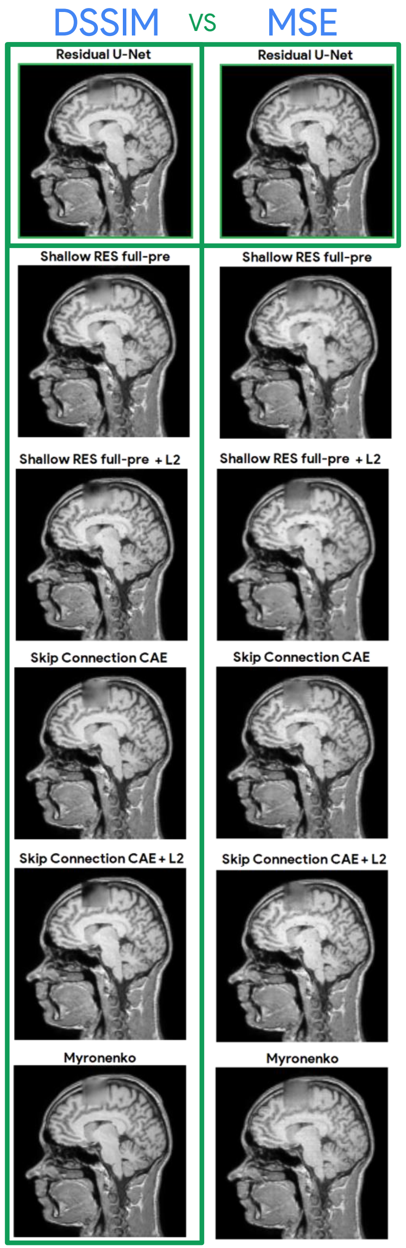
Results: Qualitative

How models reconstruct MR images?



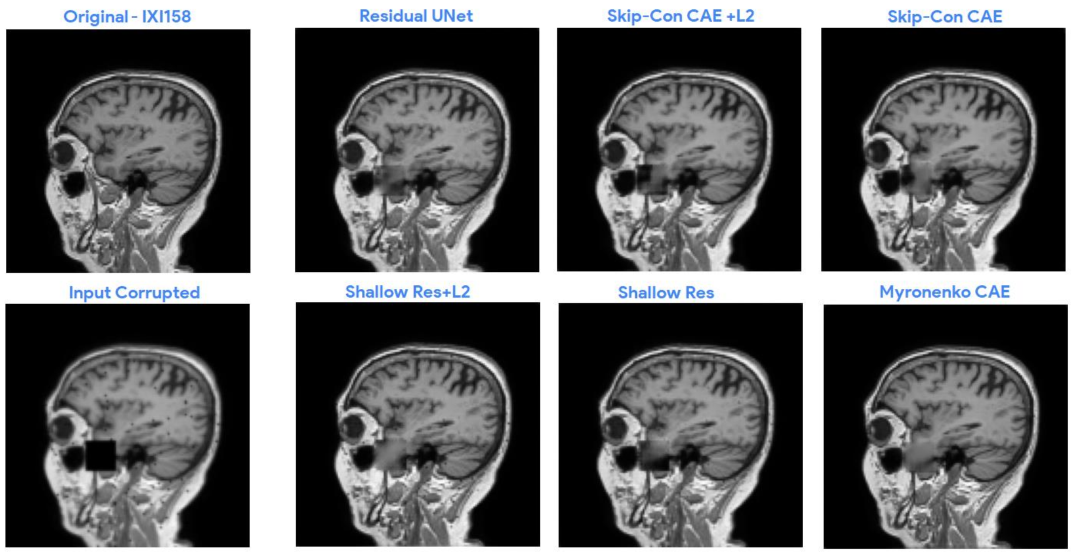
- **Archt**: Residual U-net top at reconstruction performance
- **Loss**: DSSIM models reconstruct with more structural intention
- **L2**: helps when DSSIM is used, unlike when MSE is used
- **Combination** of residual and skip-connections highly boost reconstruction performance

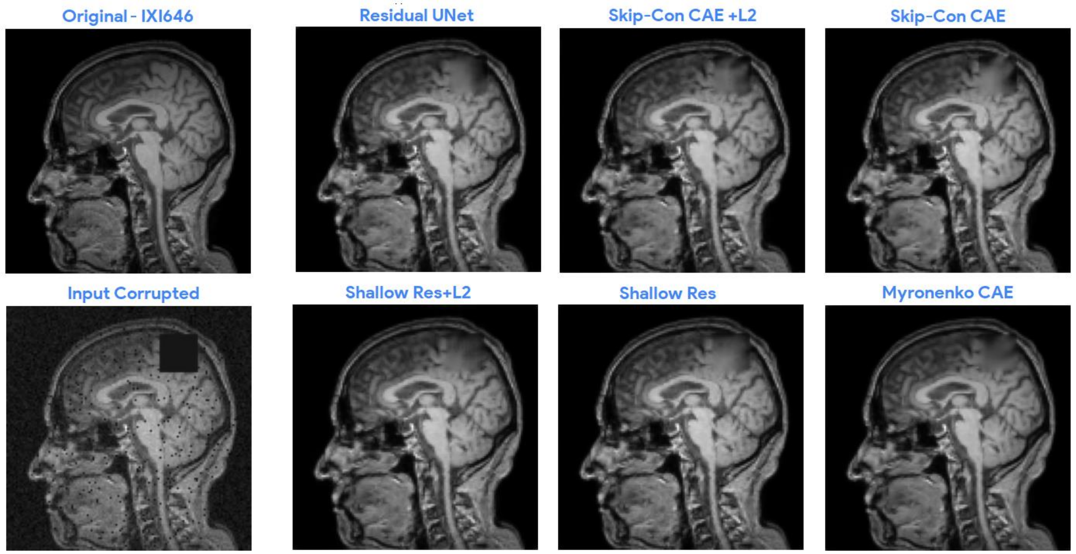




Conclusion

- Residual U-Net outperforms all methods for both loss functions Qualitatively and Quantitatively
- DSSIM loss models reconstruct better the structure and shape of the brain
- L2 regularization enhance DSSIM-loss models, but decreases MSE-loss ones
- Combination of skip connections and residual blocks outperforms each one individually
- All methods are outstanding reconstructing Gaussian noise and fixing image blur
- Dropped-out pixels are excellent reconstructed by every model but Shallow-Residual ones, which left some little noisy pixels
- Blanked-out regions are better reconstructed by DSSIM models, specifically Res-UNET and Shallow-Residual





Further work

eXplainable Artificial Intelligence (XAI)

- Deepen understanding of the performance difference
  - Visual exploration of activation of feature maps of each layer [1]
  - Visual exploration of latent space
- More reconstruction experiments
  - Reconstruct sides of the brain to see the performance improvement given by DeepBrain relevant slice selection method
  - More information → Better learning

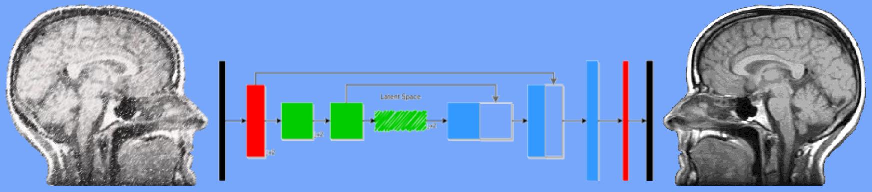
Potential trained model applications

- Unsupervised abnomaly detection. CAE as normative model. Pinaya et. al. [2]
- Improvement of brain MRI automatic analysis pipelines
  - i.e. Brain Segmentation pipelines:
    - Deep brain mask from original MRI as ground truth (GT)
    - Compare Metrics like Accuracy
      - ACU(DeepBrainMask(Corrupted), GT)
      - ACU(DeepBrainMask(Reconstruction), GT)

Train more models

- Train Residual-UNet with L2
- Different hyperparameter configurations
- Different architectures
  - Deeper models
- Transfer Learning
- Generative model approach:
  - GAN or VAE

[1] Lian-Feng Dong et. al. Learning deep representations using convolutional auto-encoders with symmetric skip connections. 2018
[2] Walter HL Pinaya et. al. Using deep autoencoders to identify abnormal brain structural patterns in neuropsychiatric disorders: A large-scale multi-sample study. 2019.



Thank you very much for your attention

Questions?

Universitat Oberta de Catalunya

Master's in Data Science
Master's Thesis
Medicine Area

January 2021

Adrián Arnaiz Rodríguez
Student

Dr. Baris Kanber
Director

aarnaizr@uoc.edu

AdrianArnaiz/Brain-MRI-Autoencoder