

# The effects of proper Pre-Processings in Biomedical Image Segmentation

June 18, 2024

*Examples Studies & Demo (CLAHE VS HistNorm)*

## 1 Proposal

Machine- and Deep Learning... - .. has an easier task when extracting most relevant information  
- .. works similar to the human sense (here: vision)

=> What we see better, mostly also AI sees better!

## 2 Example Brain MRI Segmentation

### 2.1 Different MRI Modi

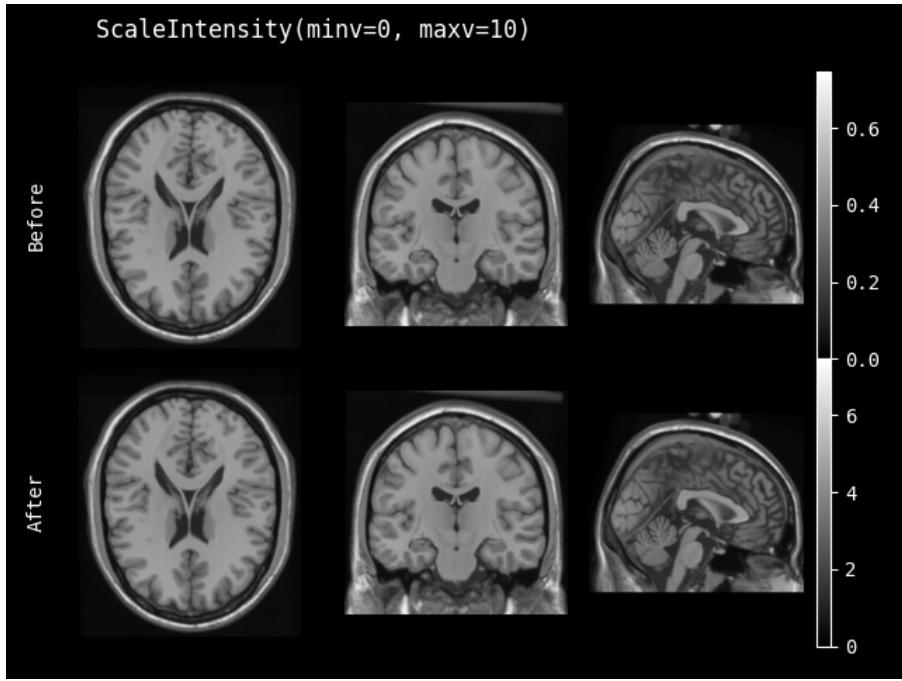
- **T1**.. relaxation time for the magnetic fields
- **T2**.. relaxation time for the spins
- **w**.. weighted -> removes responses potentially coming from fat tissue
- **gt**.. ground truth
- **FLAIR**.. Fluid-attenuated inversion recovery -> suppresses fluids

### 2.2 Standard Pre-Processings

All the usual stuff!

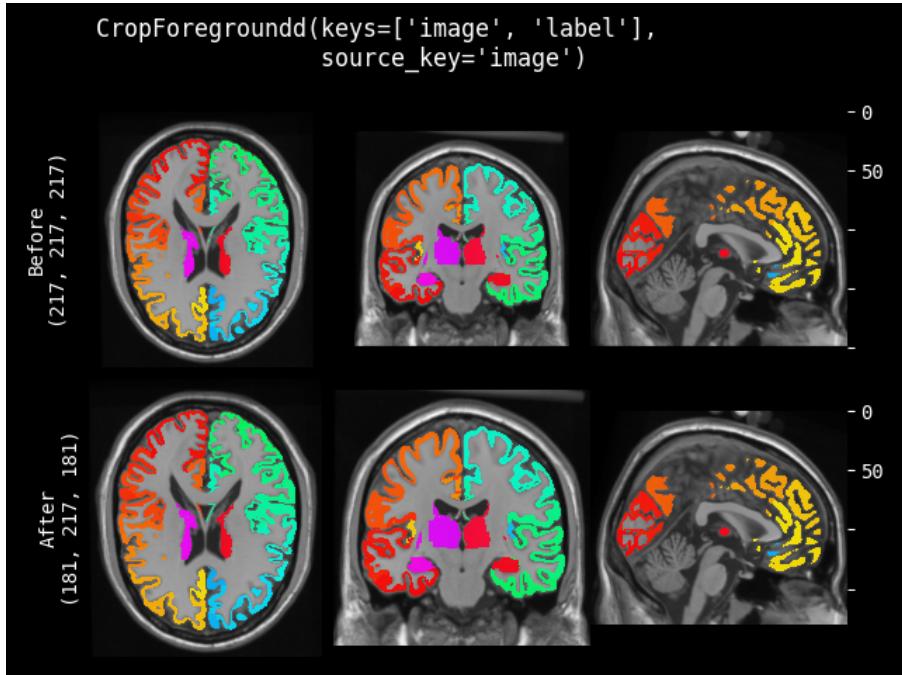
[s1] MONAI Docs // Transforms: <https://docs.monai.io/en/stable/transforms.html>

### 2.2.1 Scale Intensity



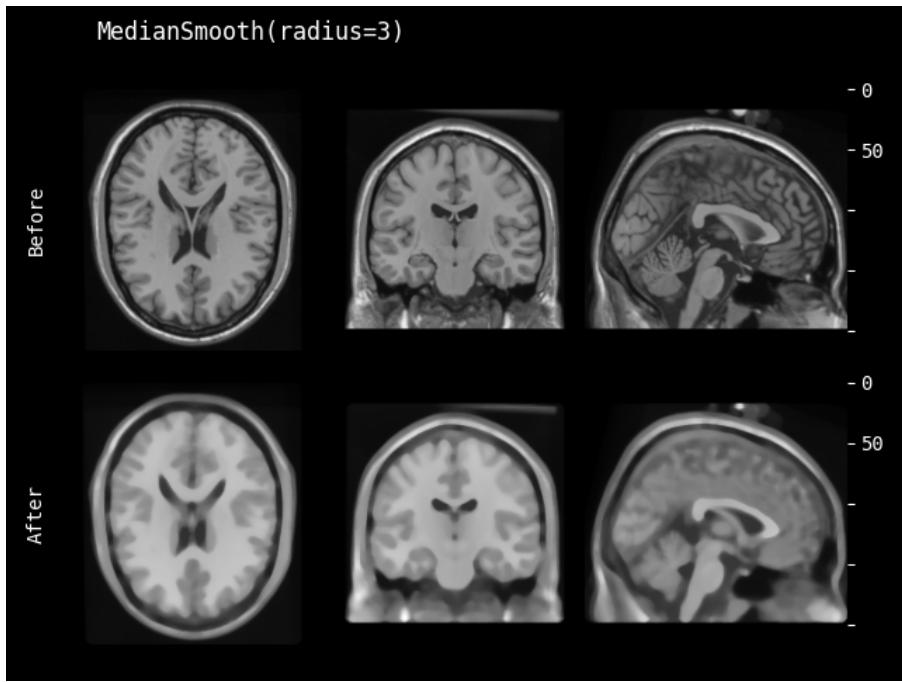
Example of adjusting the value range through ScaleIntensity preprocessing. (Source: MONAI [s1])

### 2.2.2 Crop to Foreground



Example of dynamically cropping the image around the foreground. (Source: MONAI [s1])

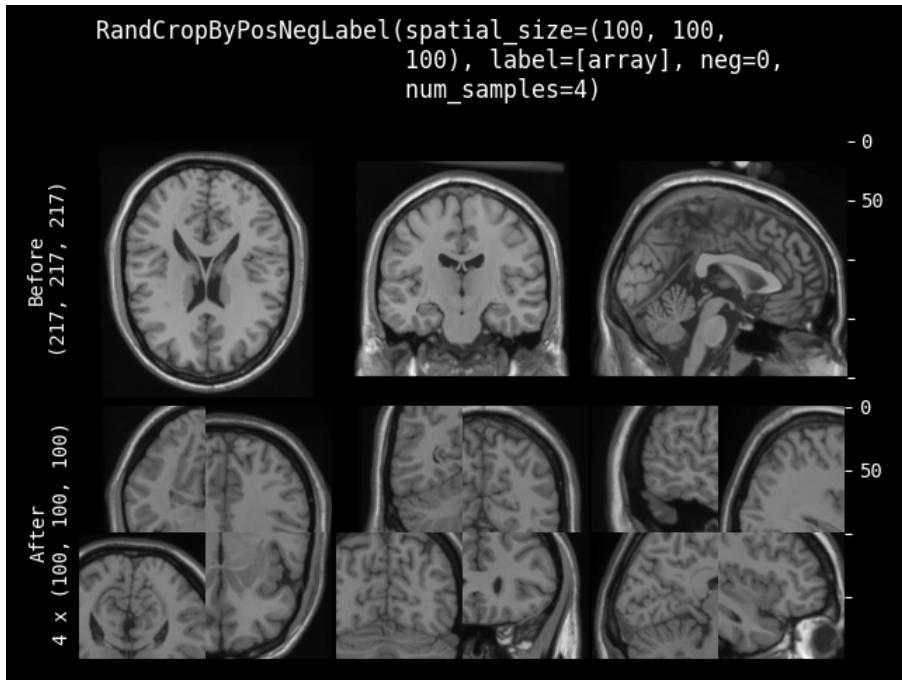
### 2.2.3 Median Smooth



Example of smoothing out the scan by taking the median of a  $7 \times 7 \times 7$  block around each voxel. (Source: MONAI [s1])

## 2.3 Augmentation Types for Training Data

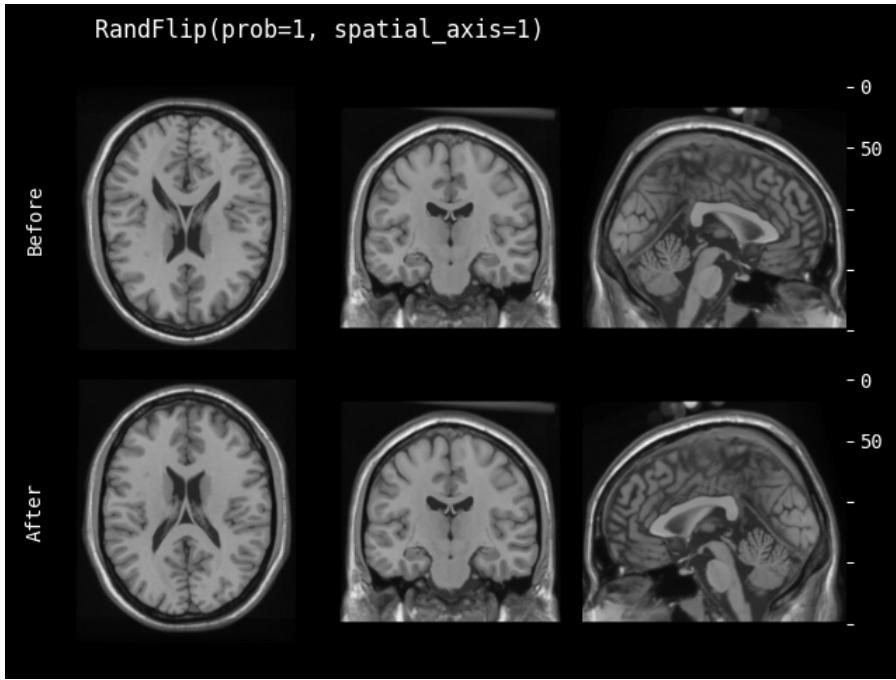
### 2.3.1 Random Cropping by Label



Example of randomly cropping the image around a specific positive and negative label ratio in fixed

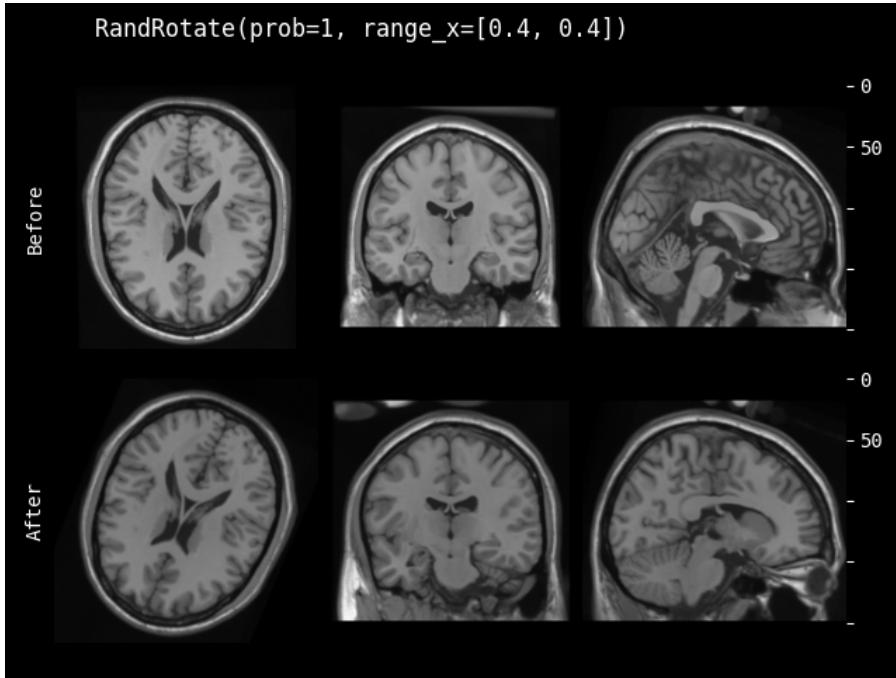
blocks. (Source: MONAI [s1])

### 2.3.2 Random Flip



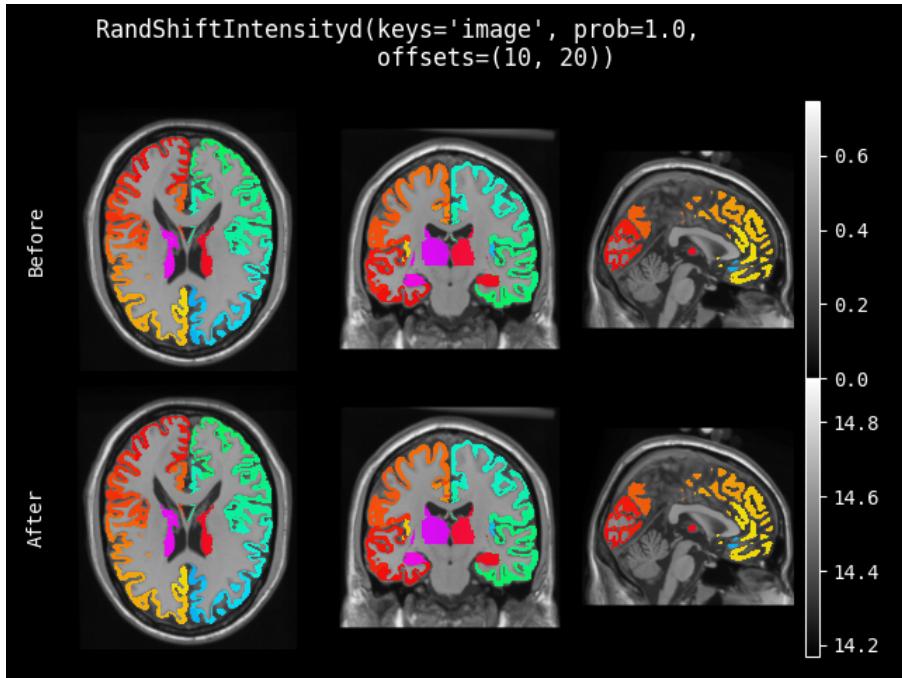
Example of flipping MRI scans around axis 1 with a probability 1.0. (Source: MONAI [s1])

### 2.3.3 Random Rotate



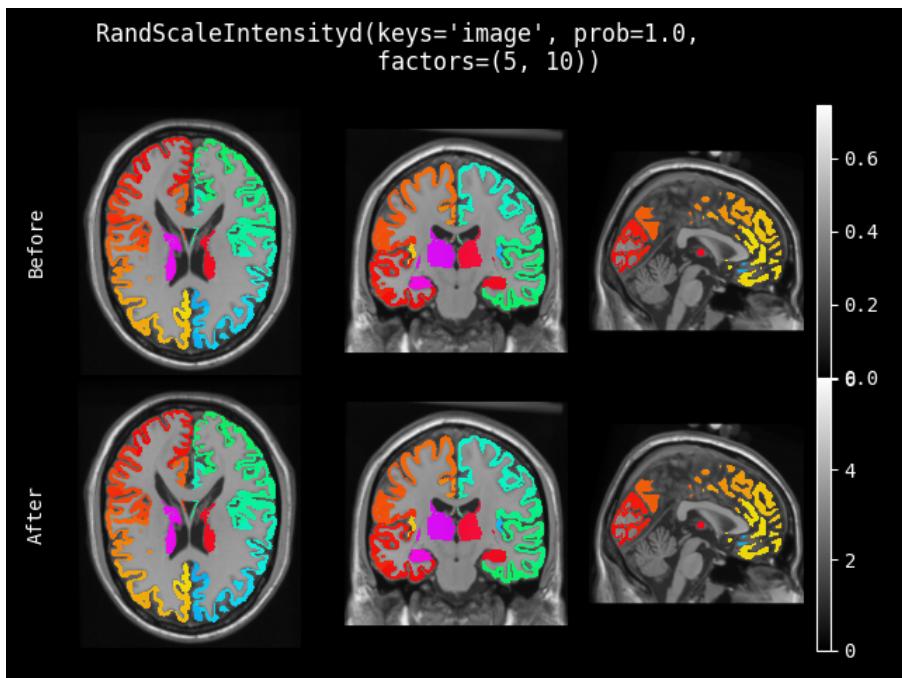
Example of rotating the image by 0.4 rads in the first two axes with a probability 1.0. (Source: MONAI [s1])

### 2.3.4 Random Intensity Shifting



*Example of shifting the intensity range between the offsets with a probability 1.0. (Source: MONAI [s1])*

### 2.3.5 Random Intensity Scaling



*Example of randomly scaling the intensity range between the factors with a probability 1.0. (Source: MONAI [s1])*

### 3 Papers

“Tchebichef Transform Domain-based Deep Learning Architecture for Image Super-resolution” [1]  
 “Enhanced pre-processing for deep learning in MRI whole brain segmentation using orthogonal moments” [2] “Sailfish Optimizer Based CLAHE with U-NET for MRI Brain Tumour Segmentation” [4]

#### 3.1 Tchebichef Transform Domain-based Deep Learning Architecture for Image Super-resolution [1]

**Problem:** Upscaling Images (also medical scans) **Dataset (medical):** COVID-19 [5] -> X-Ray & CT of human cests -> labels: Oedema, COVID-19, no Oedema **Prep.** **Method:** Tchebichef Transform **Architecture:** Residual CNN

##### 3.1.1 Tchebichef Transform [1]

$P \dots$  order,  $Q \dots$  repetition

Transform:  $T = PGQ^T$

Inverase:  $G = P^T T Q$

$$\mathbf{P} = \begin{bmatrix} \tilde{t}_0(0) & \dots & \tilde{t}_0(N-1) \\ \vdots & \ddots & \vdots \\ \tilde{t}_p(0) & \dots & \tilde{t}_p(N-1) \end{bmatrix}$$

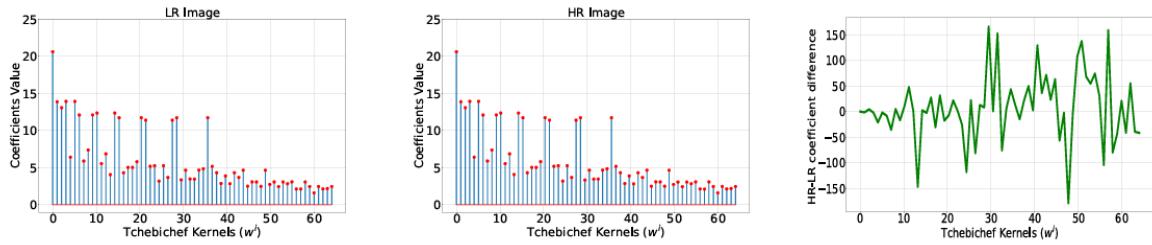
$$\mathbf{Q} = \begin{bmatrix} \tilde{t}_0(0) & \dots & \tilde{t}_0(N-1) \\ \vdots & \ddots & \vdots \\ \tilde{t}_q(0) & \dots & \tilde{t}_q(N-1) \end{bmatrix}$$

$$\tilde{t}_n(x) = \alpha_1(2x+1-N)\tilde{t}_{n-1}(x) + \alpha_2\tilde{t}_{n-2}(x)$$

where

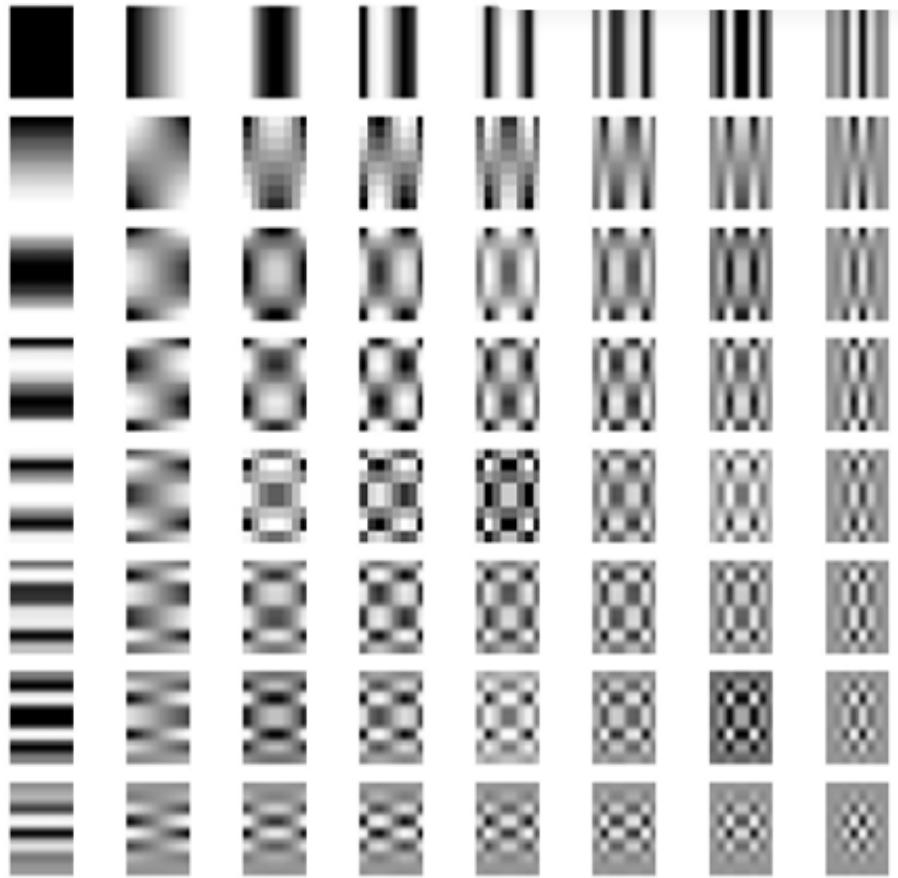
$$\alpha_1 = \frac{1}{n} \sqrt{\frac{4n^2 - 1}{N^2 - n^2}}$$

$$\alpha_2 = \frac{1-n}{n} \sqrt{\frac{2n+1}{2n-3}} \sqrt{\frac{N^2 - (n-1)^2}{N^2 - n^2}}$$



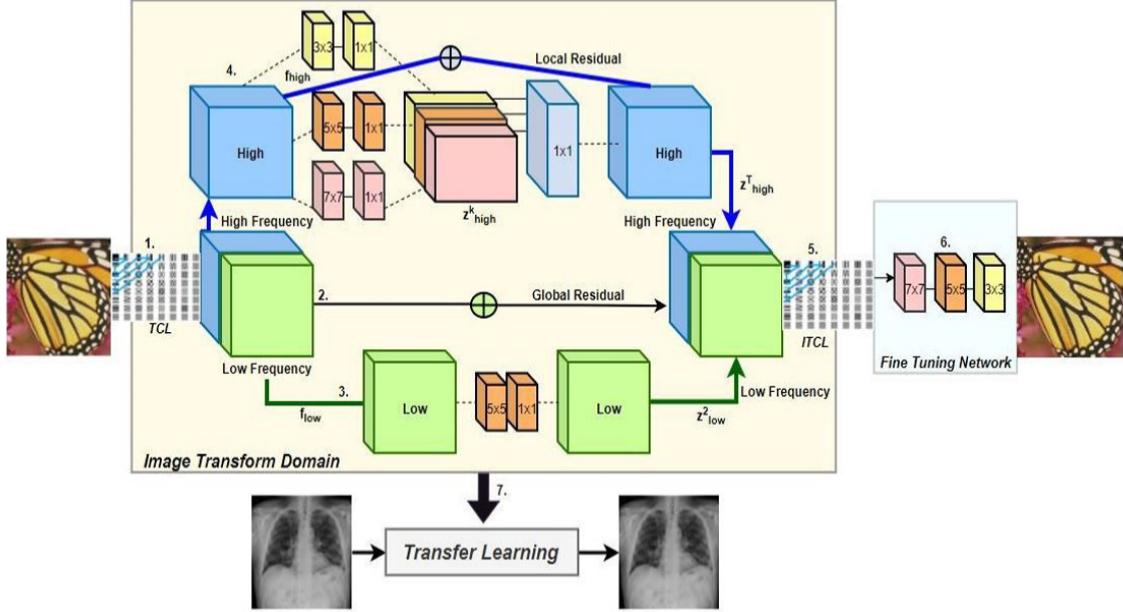
*Comparison of the corresponding Tchebichef coefficients in high- (HR) and low resolution (LR). [1]*

-> Still some difference at certain points!



*Basis functions of the Tchebichef transform. One can see here pretty well, how the frequencies increase from the upper left to the lower right. [1]*

### 3.1.2 NN-Architecture [1]



*Residual U-Net architecture with normal and inverse Tchebichef transform and high- and low frequency pathways. The high frequency parts need a much more complex CNN structure!*

### 3.1.3 Results [1]

TABLE III  
PSNR/SSIM COMPARISON ON COVID-19 DATASET

Dataset	Scale	Bicubic	SRCCN	TTDSR (Proposed)
COVID-19	2x	41.32/0.9419	41.76/0.9493	<b>43.43/0.9806</b>
	3x	40.19/0.9248	40.25/0.9281	<b>40.79/0.9681</b>
	4x	39.09/0.9059	38.96/0.9077	<b>39.83/0.9351</b>

*Results on COVID-19 breast X-rays. TTDSR is the proposed method.*

-> Good improvement over other methods!

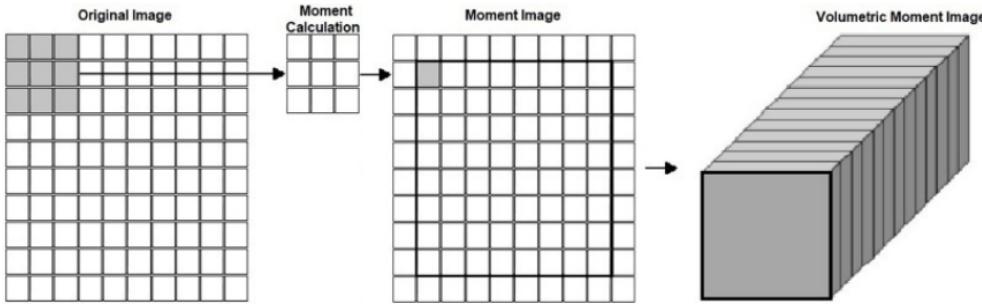
**PSNR:** Pixel Signal to Noise Ratio (JPEG: 54 dB)

$$\begin{aligned}
 PSNR &= 10 \cdot \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right) \\
 &= 20 \cdot \log_{10} \left( \frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right) \\
 &= 20 \cdot \log_{10} (\text{MAX}_I) - 10 \cdot \log_{10} (\text{MSE}).
 \end{aligned}$$

## 3.2 Enhanced pre-processing for deep learning in MRI whole brain segmentation using orthogonal moments [2]

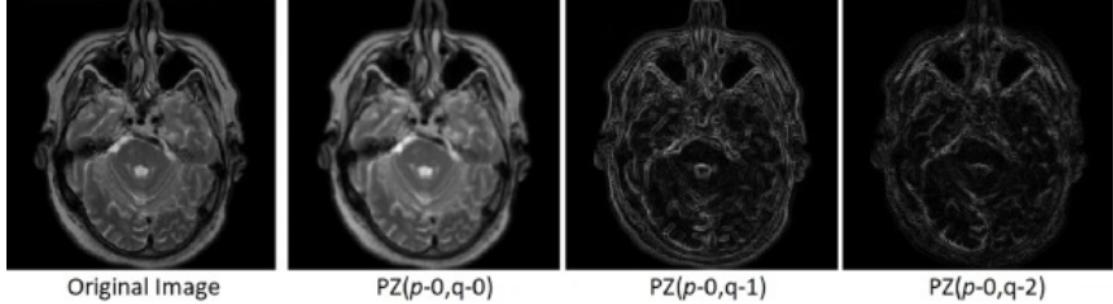
**Problem:** Brain MRI Segmentation Datasets: *Neurofeedback Skull-stripped* (NFBS) [6] -> T1 MRI of human brain ( $N=125$ ,  $D=512 \times 512 \times 192 \Rightarrow 256 \times 256 \times 96$ ) -> labels: Brain, Skull *Open access series of imaging studies* (OASIS) [7] -> T1-weighted MRI age 18-96 ( $N=416$ ,  $D=256 \times 256 \times 176$ ) -> normal aging VS Alzheimer's disease *The Cancer Imaging Archive* (TCIA) [8] -> T1, T2, and FLAIR MRI ( $N=62$ ,  $D=512 \times 512 \times 256 \Rightarrow 256 \times 256 \times 2$ ) -> Different tumor stages **Prep.** **Method:** 3x3 1st order 3 rep. Legendre- (L), Tchebichef- (T) and Pseudo-Zernike (PZ) Moments **Architecture:** U-Net

### 3.2.1 Moment Kernel [2]



The kernels of order  $p = 0$  and repetition  $q = 0$  of each moment respectively are calculated for 3x3 sliding windows around the center. [2]

-> Only slices are processed!



0th order Pseudo-Zernike moment with 2 repetitions. [2]

-> Only small contours are preserved!

### 3.2.2 Pseudo Zernike Moment [2]

$$V_{pq}(x, y) = R_{pq}e^{jq\theta} \text{ with } \theta \text{ rad. pos. } R_{pq}(r) = (k_1 r + k_2)R_{(p-1)q}(r) + k_2 R_{(p-2)q}(r) \text{ with } p = q + 2, q + 3, \dots, p_{max}$$

$$k_1 = \frac{2p(2p+1)}{(p+q+1)(p-q)}; k_2 = -2p + \frac{(p+q)(p-q-1)}{(2p-1)}k_1; k_3 = (2p-1)(p-1) - 0.5(p+q-1)(p-q-2)k_1 + 2(p-1)k_2$$

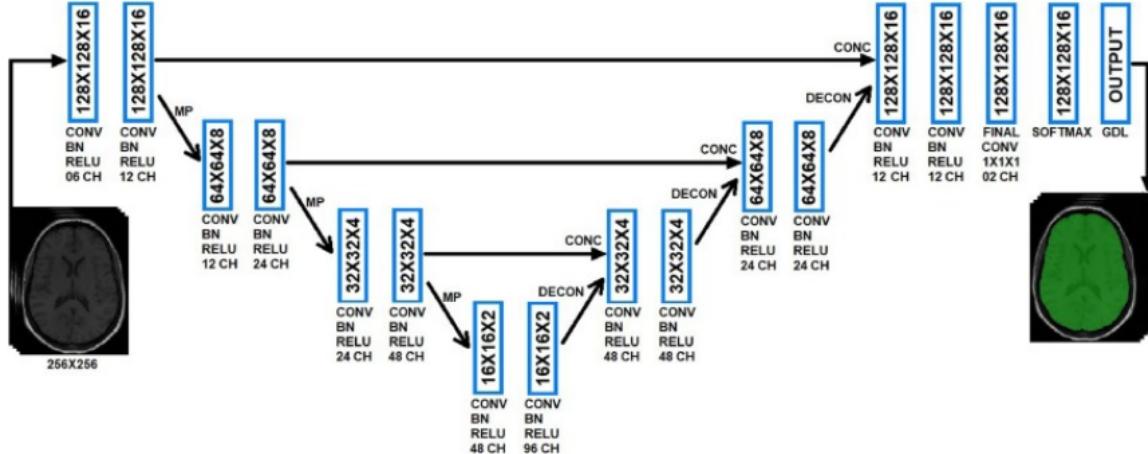
$$\Rightarrow A_{pq} = \frac{p+1}{\pi} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I_{i,j} V_{pq}^*(x_i, y_j) \Delta x_i \Delta y_j \text{ with } I.. \text{ the image}$$

### 3.2.3 Legendre Moment ( $O(M^2N^2)$ ) [2]

$$R_p(x) = \frac{(2p-1)xR_{p-1}(x)-(p-1)R_{p-2}(x)}{p} \text{ with } 0 \leq |q| \leq p, R_0(x) = 1, P_1(x) = x \text{ and } p > 1$$

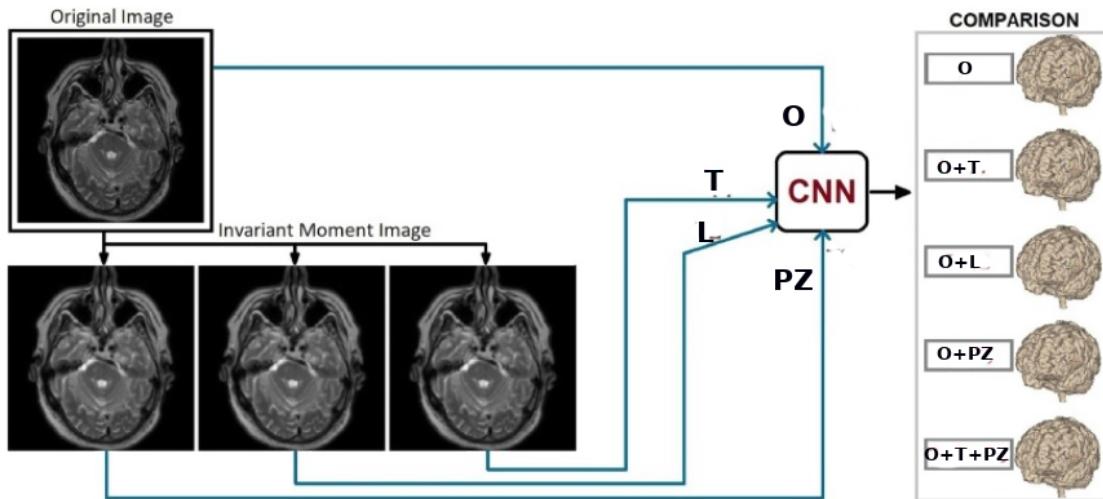
$$\rightarrow L_{pq} = \frac{(2p+1)(2q+1)}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_p(x_i)R_q(y_j)I_{i,j}$$

### 3.2.4 Architecture [2]



The U-Net setup used for the different tasks. The final layer always goes from 16 to  $k$  channels ( $k \dots$  no. classes). [2]

### 3.2.5 Setup [2]



The setups for the different data representations original ( $O$ ) and moments Pseudo Zernike ( $PZ$ ), Legendre ( $L$ ) and Tchebichef ( $T$ ) [2]

### 3.2.6 Results [2]

	<b>O</b>	<b>O+L</b>	<b>O+T</b>	<b>O+PZ</b>	<b>O+T+PZ</b>	
<b>NFBS</b>	$0.8983 \pm 0.02$	$0.7632 \pm 0.08$	$0.9212 \pm 0.02$	$0.9156 \pm 0.02$	$0.9395 \pm 0.01$	
<b>OASIS</b>	$0.9370 \pm 0.01$	$0.9066 \pm 0.02$	$0.9547 \pm 0.002$	$0.9358 \pm 0.008$	$0.9561 \pm 0.004$	
<b>TCIA</b>	$0.9044 \pm 0.01$	$0.8751 \pm 0.02$	$0.9117 \pm 0.002$	$0.9104 \pm 0.008$	$0.9149 \pm 0.004$	<i>Mean</i>

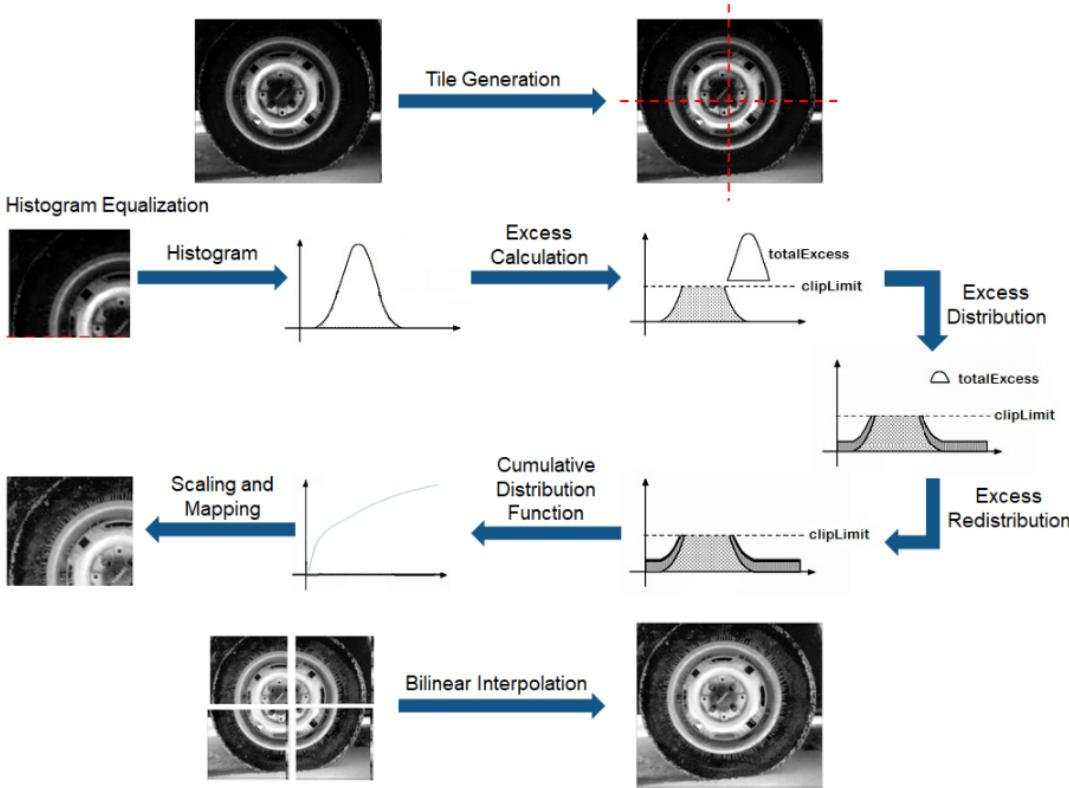
*DICE Scores of the different preprocessing methods on each dataset. [2]*

=> The O+T+PZ representation outperformed the standard by about ~4% (NFBS), ~2% (OASIS) and ~1% (TCIA)

### 3.3 Sailfish Optimizer Based CLAHE with U-NET for MRI Brain Tumour Segmentation [4]

**Problem:** Brain MRI Segmentation Dataset: Brain Tumor Segmentation (BRATS) 2019 -> T1w, T2w, and FLAIR MRI scans of (tumorous) human brains -> Targets: Gliomas segmentation necrotic/active tumour and oedema **Prep.** Method: Median Filter -> Sailfish CLAHE -> Colour Moments **Architecture:** U-Net

#### 3.3.1 Contrast Limited Histogram Equalization (CLAHE)

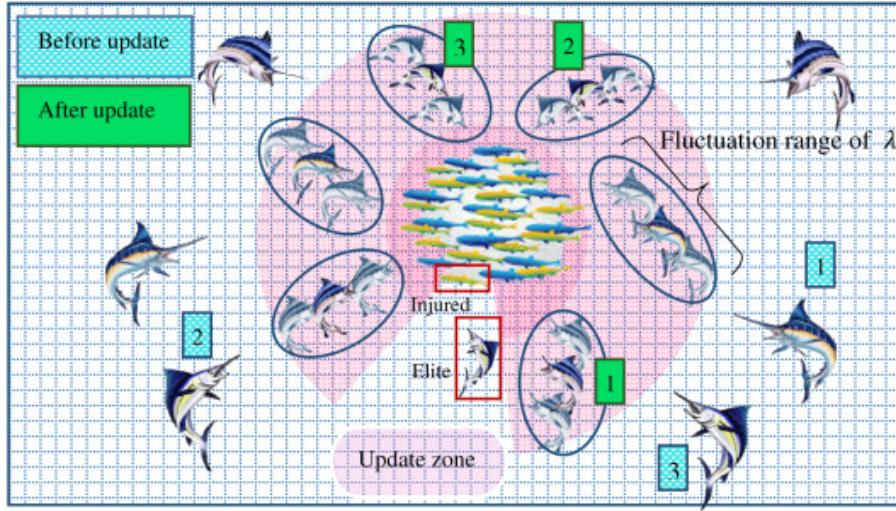


*CLAHE in a nutshell: Multiple overlapping tiles are adjusted to a specific clip limit in their histograms and elevated by the excess. Then they are recombined using bilinear interpolation. (Math-Works [s4])*

-> A clipped localized version of HistNorm

#### 3.3.2 Sailfish Optimizer [3]

Swarm based algorithm: Sailfishes hunt sardines.

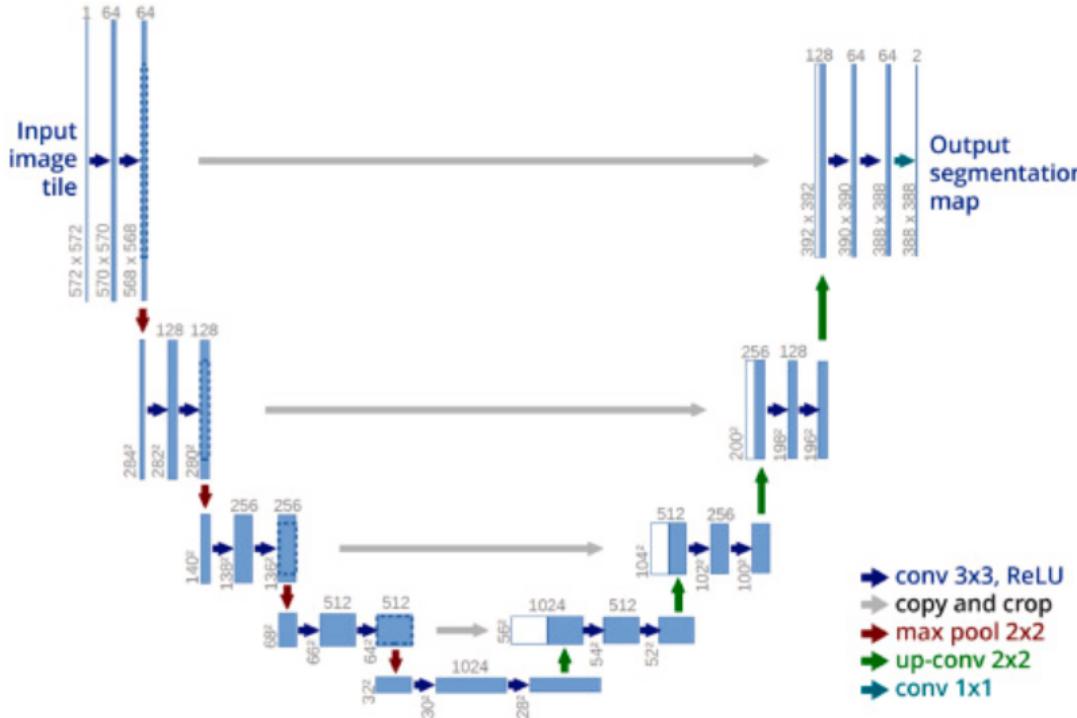


### *A pictogram of the sailfish problem [9]*

$P_{Slf}^{i+1} = P_{SlfBest} - \mu^i \times \left( rnd \times 0.5 \times (P_{SlfBest}^i + P_{SlfInjured}^i) - P_{Slf}^i \right)$  with  $P_{Slf}$ .. Sailfish position and  $rnd$ .. random number  $\in [0; 1]$

$\mu^i = 2 \times rnd \times PrD - PrD \cdot PrD..$  prey population density with  $PrD = 1 - \frac{N_{Sif}}{N_{Sif} + N_{Srd}}$  and  $N_{Sif} = NSrd \times Prcnt$

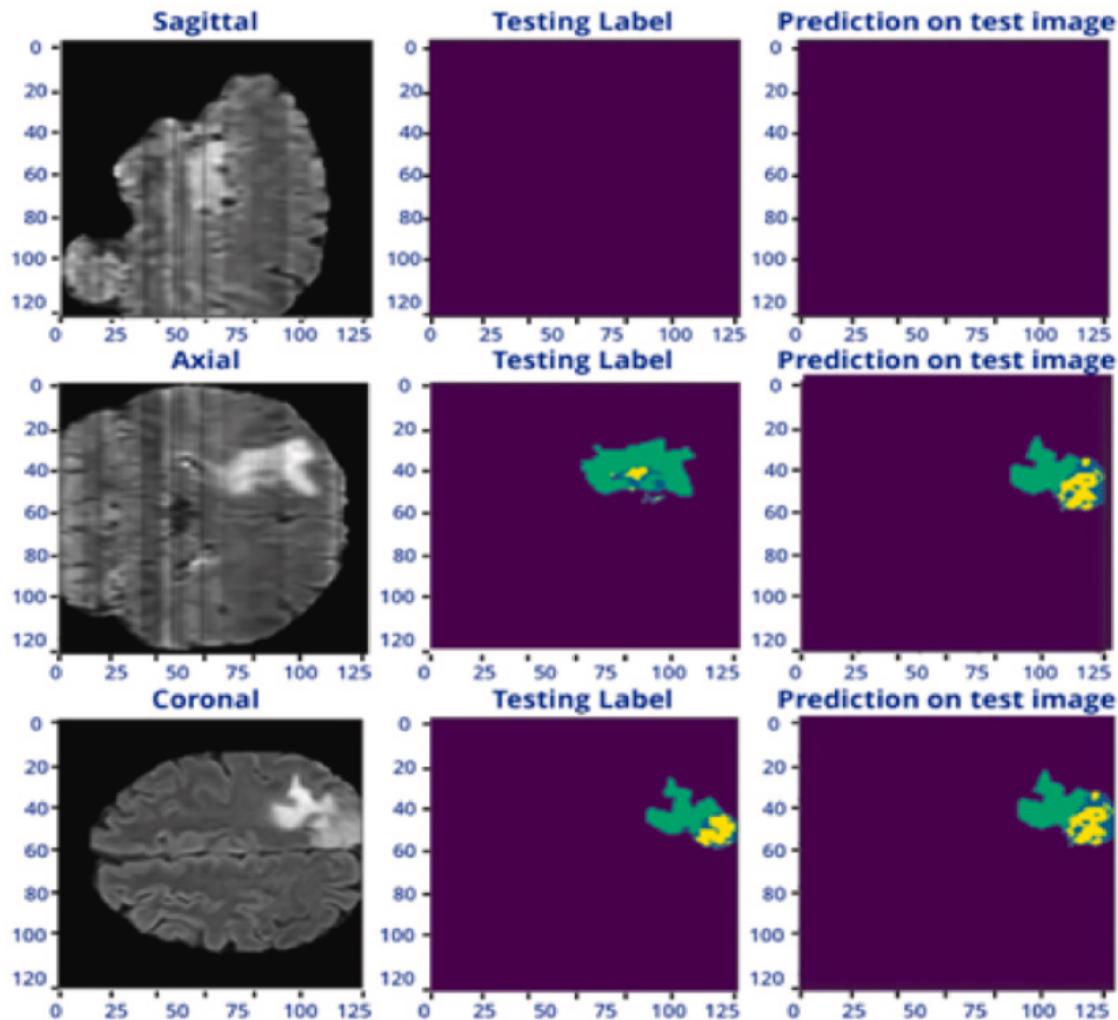
### 3.3.3 Method: UNet [3]



### 3.3.4 Results

Axial	Sagittal	Coronal
~93.5%	~93.3%	~93.5%

Resulting DICE scores for different orientations in percent [3] recalculated



Resulting segmentation beneath ground-truth and the input image [3]

- > Low quality for just 2D CLAHE
- > Probably and Master / Bachelor thesis... >\_>

## 4 Demo CLAHE VS HistNorm

### 4.1 Data

Medical Segmentation Decathlon -> Brain Tumor Segmentation



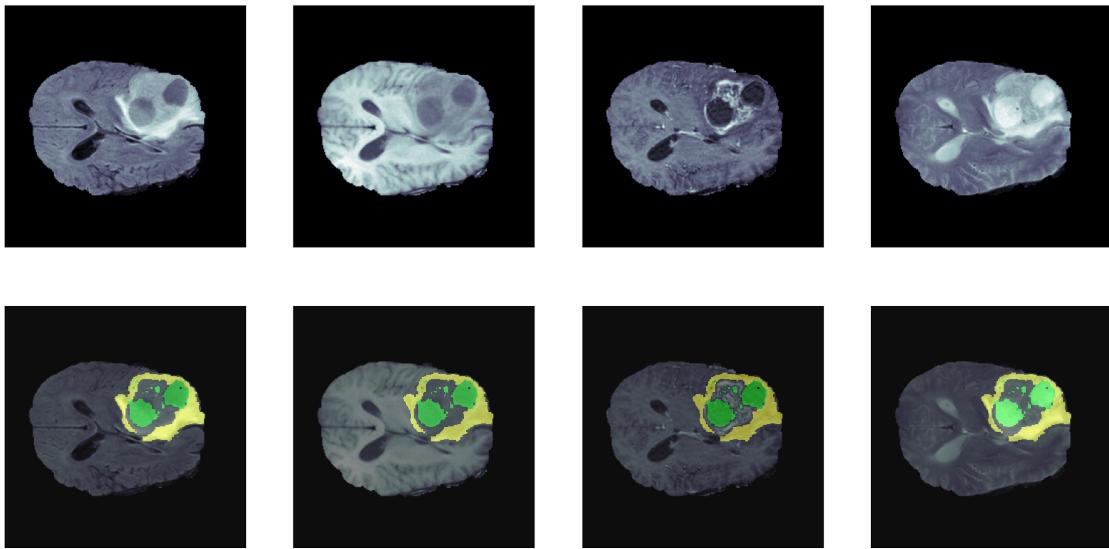
**Target:** Gliomas segmentation necrotic/active tumour and oedema **Modality:** Multimodal multisite MRI data (FLAIR, T1w, T1gd,T2w) **Size:** 750 4D volumes (484 Training + 266 Testing) **Source:** BRATS 2016 and 2017 datasets. **Challenge:** Complex and heterogeneously-located targe

Source: [s0] Medical Segmentation Decathlon - <http://medicaldecathlon.com/>

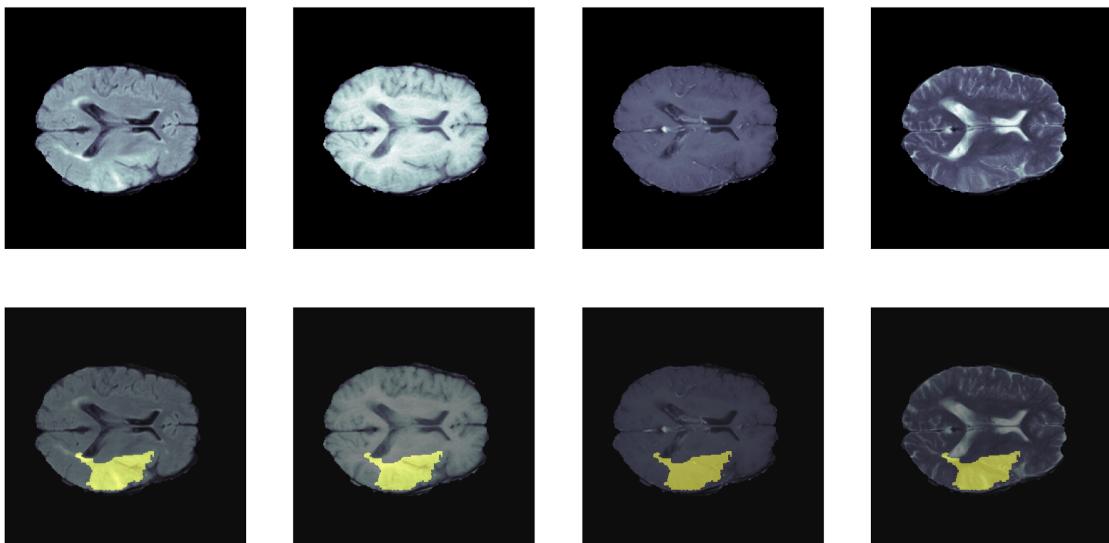
### 4.2 Preprocessing Pipeline

- **demo:** Intensity [0;1] -> Padding + Cropping into (240, 240, 240) -> CLAHE / HistNorm

```
[19]: valid = DecathlonDataset(  
    data_path,  
    task=data_task,  
    section='training',  
    transform=va_tf,  
    download=(not len(glob.glob(data_path+'/' +data_task+ '/labelsTr'))),  
    runtime_cache=True  
)
```



An example of a datapoint (top) and with its segmentation labels being overlines (bottom). The image shows the slice through the horizontal of the brain.



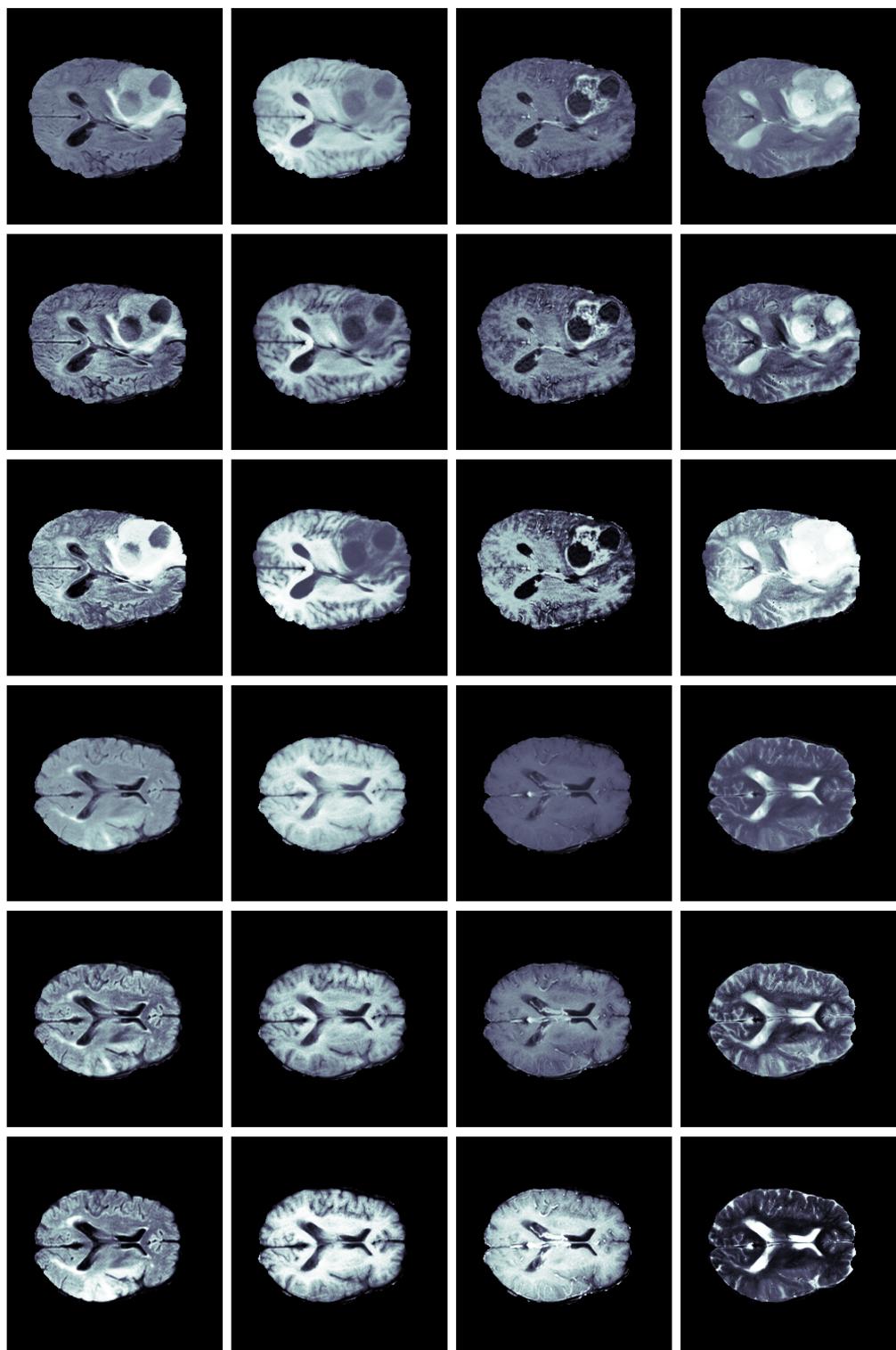
A "harder to see" example. The image shows the slice through the horizontal of the brain. We can see that sometimes seeing some oddities can be difficult!

### 4.3 CLAHE vs HistNorm

```
[26]: mclahe_tf = transforms.Compose(
    [
        transforms.LoadImaged(keys=["image", "label"]),
        transforms.EnsureChannelFirstd(keys=["image", "label"]),
        transforms.ConvertToMultiChannelBasedOnBratsClassesd(keys="label"),
        transforms.ResizeWithPadOrCropd(keys=["image", "label"], □
→spatial_size=240, mode="empty"),
        MCLAHEd(keys="image", kernel_size=[24, 24, 24]),
        transforms.ToTensord(keys=["image", "label"]),
    ]
)
hstnrm_tf = transforms.Compose(
    [
        transforms.LoadImaged(keys=["image", "label"]),
        transforms.EnsureChannelFirstd(keys=["image", "label"]),
        transforms.ConvertToMultiChannelBasedOnBratsClassesd(keys="label"),
        transforms.ResizeWithPadOrCropd(keys=["image", "label"], □
→spatial_size=240, mode="empty"),
        transforms.HistogramNormalized(keys="image"),
        transforms.ToTensord(keys=["image", "label"]),
    ]
)
```

```
[27]: mclahe_data = DecathlonDataset(
    data_path,
    task=data_task,
    section='validation',
    transform=mclahe_tf,
    download=(not len(glob.glob(data_path+'/' + data_task + '/labelsTr'))),
    runtime_cache=True
)
hstnrm_data = DecathlonDataset(
    data_path,
    task=data_task,
    section='validation',
    transform=hstnrm_tf,
    download=(not len(glob.glob(data_path+'/' + data_task + '/labelsTr'))),
    runtime_cache=True
)
```

```
[28]: mclahe_loader = DL(mclahe_data, batch_size=4)
hstnrm_loader = DL(hstnrm_data, batch_size=4)
```



**Plot** First normal-, second CLAHE and third HistNorm scans of two samples. It is clearly visible, how CLAHE adds more contrast and emphasis onto the detail, whereas the. here, HistNorm just seem to blur it out due to discretization.

## 5 Conclusion

- **Proven** effectiveness of **proper preprocessing** (studies)
- Visible quality enhancement with **CLAHE VS simple Histogramm Normalization**
- Hypothesis: **Computer Vision** is better with **visible enhancement** (as it is inspired by our vision)

## 6 Outlook

1. running more specific pre-processings
2. training U-Net, UNeTr and/or SWinUNeTr
3. model tuning & compression
4. enhancing to a more contextual dataset (e.g. whole segments of the brain)

### 6.1 Thanks for your attention!

### 6.2 References

1. Kumar & Singh 2021 - Tchebichef Transform Domain-based Deep Learning Architecture for Image Super-resolution, arXiv:2102.10640
2. Silva et al 2022 - Enhanced pre-processing for deep learning in MRI whole brain segmentation using orthogonal moments, DOI: 10.1016/j.brain.2022.100049
3. Vu et al 2018 - 3D convolutional neural network for feature extraction and classification of fMRI volumes, DOI:10.1109/PRNI.2018.8423964
4. Iakshmi & Rani 2024 - Sailfish Optimizer Based CLAHE with U-NET for MRI Brain Tumour Segmentation, DOI: 10.1016/j.measen.2024.101229
5. Cohen et al 2020 - Covid-19 image data collection. arXiv:2003.11597, 2020
6. Eskildsen et al 2012 - BEaST: Brain extraction based on nonlocal segmentation technique 10.1016/j.neuroimage.2011.09.012
7. Marcus et al 2007 - Open access series of imaging studies (OASIS): Cross-sectional MRI Data in young, middle aged, nondemented, and demented older adults10.1162/jocn.2007.19.9.1498
8. Clark et al 2013 - The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository10.1007/s10278-013-9622-7
9. Shadravan et al 2019 - The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems10.1016/j.engappai.2019.01.001

### 6.3 Sources

[s0] Medical Segmentation Decathlon - <http://medicaldecathlon.com/> [s1] MONAI Docs // Transforms: <https://docs.monai.io/en/stable/transforms.html> [s2] MATLAB Docs // Medical Image Preprocessing: <https://de.mathworks.com/help/medical-imaging/ug/overview-medical-image-preprocessing.html> [s3] MATLAB Docs // CLAHE: <https://de.mathworks.com/help/visionhdl/ug/contrast-adaptive-histogram-equalization.html>