2025

# Bridging the image gap

# Validation of Synthesized T1-weighted Images from FLAIR

Master’s thesis, Clinical Science and Technology, Aalborg Universitety

AALBORG
UNIVERSITET

**Projekttitel:** Validering af syntetiseret T1-vægtet billeder fra FLAIR.

**Type:** Videnskabelig artikel

**Tema:** Kandidatspeciale

**ECTS:** 30 ECTS

**Projektperiode:** februar 2025 - juni 2025

**Institution:** Aalborg Universitet, Aalborg

**Uddannelse:** Klinisk videnskab og teknologi

**Semester:** 4. semester

**Projektgruppe:** 10504

**Projektgruppens medlemmer:**

Maria Dalmose Rasmussen

**Vejleder:**

Lasse Riis Østergaard

**Bi vejleder:**

Anders Ulrich Eriksen

**Sider:** 46

**Referencesystem:** Elsevier Vancouver, Mendeley

## Resumé

**Baggrund**: Magnetisk Resonans billeddannelse (MR) anvendes hyppigt klinisk til vurdering af apopleksi, hvor Fluid Attenued Inversion Recovery (FLAIR) er en central sekvens til detektering og karakterisering af patologi. Dog er T1-vægtede sekvenser, som ofte kræves til avanceret neurobilledanalyse og forskningssoftwareværktøjer, ofte utilgængelige, hvilket udgør en betydelig begrænsning. SynthSR, et neuralt netværk der kan syntetisere manglende T1-vægtet billeder ud fra andre MR-sekvenser, tilbyder en potentiel løsning. Formålet med dette studie var at validere SynthSR's evne til at syntetisere realistiske T1-vægtet billeder ud fra FLAIR, og at vurdere ligheden af de syntetiserede T1-vægtet billeder sammenlignet med referencedata.

**Metode:** Data bestod af FLAIR og T1-vægtet MR-billeder fra 95 apopleksipatienter. FLAIR billederne fungerede som input til SynthSR for at syntetisere T1 billeder. Efter syntetisering, blev både de originale og syntetiserede T1 billeder registreret til MNI-152 skabelon og normaliseret før udregning af Mean Squared Error (MSE) og Structural Similarity Index (SSIM), samt subjektiv vurdering.

**Resultater:** 87 FLAIR og T1-vægtet billede par blev inkluderet til syntetisering. Objektiv vurdering viste en gennemsnits MSE på $0.56 \pm 0.17$ og en gennemsnits SSIM på $0.35 \pm 0.66$ når de syntetiserede T1-vægtet billeder blev sammenlignet med de originale T1-vægtet billeder. Sammenlignet med MNI-152 skabelonen viste de originale T1-vægtet billeder en gennemsnits MSE på $1.12 \pm 0.17$ og en gennemsnits SSIM på $0.32 \pm 0.06$, hvor de syntetiserede T1-vægtet billeder viste en gennemsnits MSE på $0.99 \pm 0.04$ og en gennemsnits SSIM på $0.32 \pm 0.01$.

**Konklusion:** SynthSR syntetiserede T1-vægtet billeder ud fra FLAIR der visuelt lignede de originale T1-vægtet billeder. Dog viste objektive målinger forskelle i både pixelintensitet og strukturel lighed, hvilket understreger behovet for yderligere forskning.

**Project title:** Validation of Synthesised T1-weighted Images from FLAIR

**Type:** Scientific article

**Tema:** Master’s thesis

**ECTS:** 30 ECTS

**Project period:** February 2025 - June 2025

**Institution:** Aalborg Universitety, Aalborg

**Education:** Clinical Science and Technology

**Semester:** 4. semester

**Project group: 10504**

**Project Group members:**

Maria Dalmose Rasmussen

**Supervisor:**

Lasse Riis Østergaard

**Co-supervisor:**

Anders Ulrich Eriksen

**Pages:** 46

**Reference system:** Elsevier Vancouver, Mendeley

## Abstract

**Background**: Magnetic Resonance Imaging (MRI) is widely used clinically for stroke assessment, with Fluid-Attenuated Inversion Recovery (FLAIR) being a key sequence for detecting and characterising pathology. However, T1-weighted sequences, often required for advanced neuroimaging analysis and research software tools, are frequently unavailable, posing a limitation. SynthSR, a neural network capable of synthesising missing T1-weighted images from other MRI sequences, offers a potential solution. The aim of this study was to validate SynthSR’s capability to synthesise realistic T1-weighted images from FLAIR, and to assess the similarity of the synthesised T1-weighted images compared to the ground truth.

**Method:** Data consisted of FLAIR and T1-weighted MRI images from 95 stroke patients. FLAIR images served as input for SynthSR to synthesise T1-weighted images. After synthesis, both the original and synthesised T1-weighted images were registered to the MNI-152 template and normalised before computing Mean Squared Error (MSE) and Structural Similarity Index (SSIM), alongside subjective validation.

**Results:** 87 FLAIR and T1-weighted image pairs were included for synthesis. Objective assessment revealed a mean MSE of $0.56 \pm 0.17$ and mean SSIM of $0.35 \pm 0.66$ when synthesised T1-weighted images were compared against original T1-weighted images. Against the MNI-152 template, original T1-weighted images exhibited a mean MSE of $1.12 \pm 0.17$ and mean SSIM of $0.32 \pm 0.06$, whereas synthesised T1-weighted images showed a mean MSE of $0.99 \pm 0.04$ and mean SSIM of $0.32 \pm 0.01$.

**Conclusion:** SynthSR synthesised T1-weighted images from FLAIR that were visually similar to the original T1-weighted images. However, objective measurements revealed differences in both pixel intensity and structural similarity, emphasising the need for further research.

## Preface

This master's thesis was conducted by one student from the master's in clinical science and technology at Aalborg University from 1$^{st}$ of February to 2$^{st}$ of June 2025.

The master's thesis is primarily intended for lecturers, supervisors, researchers, and fellow students at the Faculty of Health Sciences, Aalborg University. It can be accessed via Aalborg University's project library, AAU Studenterprojekter. Furthermore, it is intended to specialists who works with neuroimaging, especially MRI, or image synthesis.

The master's thesis's objective was to assess synthesised T1-weighted images and their potential utility in neuroscience research. This was facilitated by image processing and synthesis of data derived from a Brazilian study by Lemos et al (1).

### Scientific paper

The master's thesis was authored as a scientific paper, which was structured according to the IMRAD format.

### Worksheets

Supplementary worksheets are appended to the scientific article to meet the current learning objectives from the semester description.

The author of this master's thesis would like to extend a sincere thank you to my supervisors, Lasse Riis Østergaard and Anders Ulrich Eriksen, for their efficient and prompt sparring, as well as their excellent guidance and constructive feedback. A special thank you goes to them for their patience and advice throughout the process of learning programming and image processing, and for always having their door open.

The Vancouver citation system has been used in this study.

## Table of content

*Original article*

# Validation of Synthesised T1-weighted Images from FLAIR

**Maria Dalmose Rasmussen, MSc** [1]

## Abstract

**Background**: Magnetic Resonance Imaging (MRI) is widely used clinically for stroke assessment, with Fluid-Attenuated Inversion Recovery (FLAIR) being a key sequence for detecting and characterising pathology. However, T1-weighted sequences, often required for advanced neuroimaging analysis and research software tools, are frequently unavailable, posing a limitation. SynthSR, a neural network capable of synthesising missing T1-weighted images from other MRI sequences, offers a potential solution. The aim of this study was to validate SynthSR's capability to synthesise realistic T1-weighted images from FLAIR, and to assess the similarity of the synthesised T1-weighted images compared to the ground truth.

**Method:** Data consisted of FLAIR and T1-weighted MRI images from 95 stroke patients. FLAIR images served as input for SynthSR to synthesise T1-weighted images. After synthesis, both the original and synthesised T1-weighted images were registered to the MNI-152 template and normalised before computing Mean Squared Error (MSE) and Structural Similarity Index (SSIM), alongside subjective validation.

**Results:** 87 FLAIR and T1-weighted image pairs were included for synthesis. Objective assessment revealed a mean MSE of $0.56 \pm 0.17$ and mean SSIM of $0.35 \pm 0.66$ when synthesised T1-weighted images were compared against original T1-weighted images. Against the MNI-152 template, original T1-weighted images exhibited a mean MSE of $1.12 \pm 0.17$ and mean SSIM of $0.32 \pm 0.06$, whereas synthesised T1-weighted images showed a mean MSE of $0.99 \pm 0.04$ and mean SSIM of $0.32 \pm 0.01$.

**Conclusion:** SynthSR synthesised T1-weighted images from FLAIR that were visually similar to the original T1-weighted images. However, objective measurements revealed differences in both pixel intensity and structural similarity, emphasising the need for further research.

**Keywords:** Neuroimaging, deep learning, image synthesis, magnetic resonance imaging, mean squared error, structural similarity index, medical imaging, stroke, FLAIR, T1-weighted imaging, image quality assessment, SynthSR.

## Introduction

Stroke is one of the leading causes of morbidity and mortality worldwide (2). Magnetic Resonance Imaging (MRI) is one of the leading imaging procedures for diagnosis, characterization and follow-up for stroke, and are used for both clinical and research purposes. In clinical practice, Fluid-Attenuated Inversion Recovery (FLAIR) is one of the several MRI sequences used for stroke diagnosis due to its ability to visualize pathological changes such as white matter abnormalities and hyperintense

[1] Department of Health Science and Technology, Aalborg University, Aalborg, Denmark

lesions. Another MRI sequence, T1-weigthed imaging, are especially used in neuroscience because of its capability to demonstrate anatomy (3–5). Therefore, neuro studies use this sequence as a reference for registration, segmentation and atlas based analyse (3–5). Furthermore, most of the software for doing those analysis, requires high resolution T1-weigthed images (6,7). Given that T1-weighted images are not included in all MRI scans, including MRI scans for stroke (8–10), this is a limitation for neuroscience research.

As the majority of MRI scans are conducted for clinical purposes, this is reflected in the available datasets, and sequences with T1-weighting is missing in most of the clinical scans (11,12). This is a challenge in neuroscience research, because to execute analyses such as segmentation and registration specific demands are necessary.

To solve these challenges with missing MRI images, synthetic MRI images could be a resolution for use in neuroscience. Numerous studies have investigated several methods for synthesising MRI images using different models of neural networks (6,13,14). Recently, SynthSR have demonstrated promising results in synthesising neurological imaging compared to previous approaches. SynthSR is a neural network implemented in FreeSurfer, that synthesise clinical brain scans into high resolution T1-weighted images (6). The result can then be analysed with any established tool for registration or segmentation. SynthSR is based on two Convolutional Neural Network (CNN) each with its own U-net and combine a domain randomization approach with a generative brain MRI model. This makes it possible to handle scans in any contrast or resolution. Furthermore, is uses an auxiliary segmentation task to produce realistic images and replace lesions with healthy tissue (6).

Although T1-weigthed MRI images is among the sequences that are most often missing, but also most wanted for analyses, there is limited research in synthesising this sequence.

Therefore, the purpose of this study is to validate SynthSR capability to synthesise T1-weighted MRI images from FLAIR. Furthermore, to validate the similarity of the synthesised T1-weighted images compared with original T1-weighted images as the ground truth.

## Method

To validate SynthSR capability to synthesise T1-weighted images and assess the quality of those images, several processes were performed. These processes are shown in Figure 1.

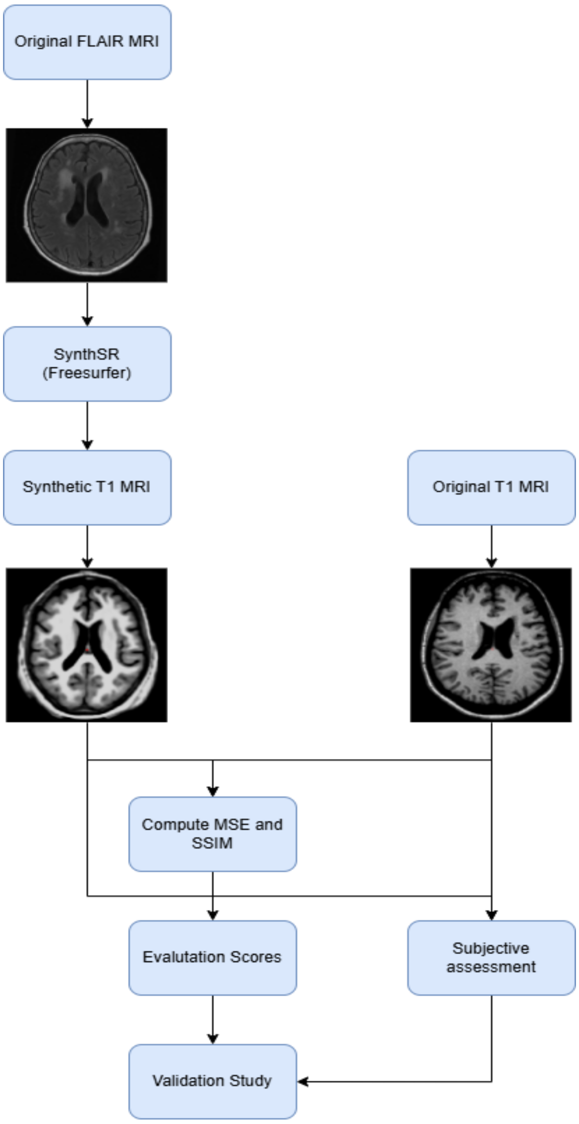


*Figure 1: Flowchart illustrating the methodology for evaluating synthesised T1-weighted image quality using SynthSR.*

In general, all procedures were performed using *Python 3.10.12* on *Ubuntu 22.04*. For image synthesis, *Freesurfer 8.0.0* was utilised to run *SynthSR 2*.

Relevant Python libraries and modules used for individual processes included: *SimpleITK 2.4.1, NumPy 2.2.4, Pandas 2.2.3, Nibabel 5.3.2, Scikit-image (skimage) 0.25.2, Scikit-learn (sklearn) 1.6.1* and *Advanced Normalization Tools (ANTs) 0.3.26.3.*

## Data

The data used in this study originated from the study by Lemos et al (1). The data were collected in 2018-2019, with stroke patients recruited from a pain centre at Hospital das Clínicas da Universidade de São Paulo. The inclusion criteria for patients were adults with stroke (ischaemic or haemorrhagic) and documented MRI lesions. The exclusion criteria for the study were missing or unsuitable MRI images, as well as the absence of T1-weighted, FLAIR, or diffusion-weighted images (1).

## Image Synthesis

Image synthesis in this study was performed using SynthSR within Freesurfer (6). 3D FLAIR MRI images were used for synthesis and thus served as input to SynthSR.

As the input images originated from both 1.5 T and 3 T MRI scanners, the synthesis was performed with the low-field flag enabled. Furthermore, the newest version of SynthSR v2 was used, so the v1 flag was not applied. The command utilised 32 CPU threads, which was optional but made the process faster.

## Image registration

Prior to measuring the anatomical metrics and comparing the synthesised T1-weighted image with the original T1-weighted image, all images required registration and alignment.

Image registration involved establishing a geometric transformation to align the images. In this way, two different MRI images could be compared, and an overlay between the images could be visualised and measured. For registration, the standard MNI-152 brain space was used. This model was an average of 152 T1-weighted MRI scans, which were transformed to form a symmetric model and create a common spatial framework where different brains could be compared (15,16).

For this study, multiple registrations were performed. FLAIR images were registered to the synthesised T1-weighted images when the synthesis was carried out. Furthermore, the original T1-weighted images and the synthesised T1-weighted images were registered to the MNI-152 space.

The *ANTs* library was utilised for registration procedures (17–19).

Registration allowed for the assessment of how overall registration performance differed when a synthesised T1-weighted image was used instead of the original T1-weighted image. Furthermore, registration enabled the visualisation and evaluation of the spatial similarity between the synthesised and original T1-weighted images. Likewise, the SynthSR accuracy and quality in generating synthetic T1-weighted images from FLAIR images could be validated by studying how well the images spatially matched.

Henceforth, the MNI-152 space will be referred to as “the template image” in this study.

## Intensity Normalisation

After registration, the images were normalised to ensure consistent intensity values for the subsequent measurements. This normalisation was performed in Python using Z-score normalisation, calculated as follows:

$$\mathrm{Z} = \frac{x - \mu}{\sigma}$$

This normalisation transformed the data to have a mean around 0 and a standard deviation around 1.

For this normalisation, the *NumPy, Nibabel* and *Scikit-learn (sklearn)* libraries were used (20–24).

The normalisation was applied individually to both the original and synthesised T1-weighted images, as well as the template image.

These normalised images were then used for objective image quality measurement and are henceforth referred to as “synthesised T1-weighted images,” “original T1-weighted images,” and “template images” respectively.

## Anatomical metrics

In this study, objective image quality measurements were performed using Mean Squared Error (MSE) and the Structural Similarity Index (SSIM). Each metric was measured between the original T1-weighted image and the synthesised T1-weighted image. Additionally, measurements were taken between the original T1-weighted image and the template it was registered to, and finally, between the synthesised T1-weighted image and the template.

### Mean Squared Error (MSE)

MSE calculates the average of the squared differences between pixels in two images, providing a numerical value for the total error or difference between them. It is given by the following formula:

$$MSE = \frac{1}{M * N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(i,j) - g(i,j)]^2$$

Here, $g$ represents the synthesised image and $f$ represents the ground truth image. $M$ and $N$ are the dimensions of the image, and $(i,j)$ refer to the pixel values within the images.

Higher MSE values indicate larger differences in pixel intensity between the images. Therefore, the aim is to achieve an MSE value close to zero (25,26).

The libraries used for calculating MSE in Python were *SimpleITK, NumPy* and *Pandas* (27–29)

### Structural Similarity Index (SSIM)

The SSIM index measures the structural similarity between two images - the synthesised and the original MRI image - by combining three components:

$$SSIM = \left(\frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1}\right) * \left(\frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2}\right) * \left(\frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3}\right)$$

The SSIM index typically ranges between zero and one. If the SSIM was equal to one, the images *f* and *g* were identical.

Kowalik-Urbaniak et al. showed that SSIM had the best performance and was therefore the measure that most closely aligned with radiologists’ assessments (25,26).

The libraries used for calculation SSIM in Python were *Nibabel, NumPy, Pandas* and *Scikit-image (skimage)* (20,21,23,29,30)

## Statistical Analysis

Descriptive statistics were performed using *SPSS 29.0.0.0*. The descriptive statistics included mean, median, minimum, maximum, standard deviation, variance and percentiles for both MSE values and SSIM values. Furthermore, Boxplots were made for visual representation of the descriptive statistics for MSE values and SSIM values.

## Subjective Image Quality Assessment

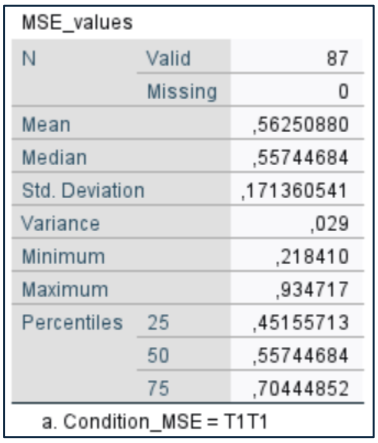
A subjective image quality assessment was conducted by the author of this study, in addition to the objective measurements. This was performed by visually comparing the synthesised and original images.

## Results

MRI data from 95 patients were initially included in this study. These data consisted of T1-weighted and FLAIR sequences from all 95 patients. However, after a subjective review of the data, MRI data from 8 patients were excluded due to poor image quality and artefacts. Therefore, MRI data from 87 patients were finally included.

### Comparison of Original T1-weighted Images with Synthesised T1-weighted Images

Table 1 presents descriptive statistics for the MSE values obtained from the comparison between the synthesised T1-weighted and original T1-weighted images.



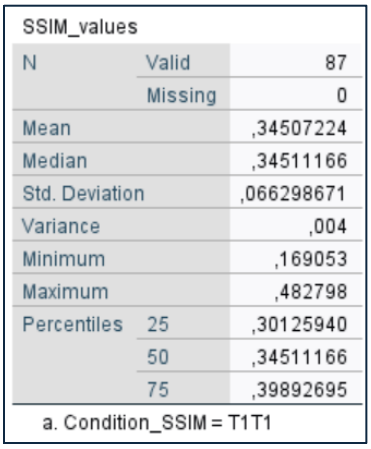
| MSE_values | | |
|---|---|---|
| N | Valid | 87 |
| | Missing | 0 |
| Mean | | ,56250880 |
| Median | | ,55744684 |
| Std. Deviation | | ,171360541 |
| Variance | | ,029 |
| Minimum | | ,218410 |
| Maximum | | ,934717 |
| Percentiles | 25 | ,45155713 |
| | 50 | ,55744684 |
| | 75 | ,70444852 |

a. Condition_MSE = T1T1

*Table 1: Descriptive statistics of MSE-values between original T1-weighted and synthesised T1-weighted images.*

For the comparison between the synthesised T1-weighted and original T1-weighted images, the mean MSE value was 0.56 ± 0.17, and the values ranged from 0.22 to 0.93 (Table 1). The descriptive statistics are visually presented in Figure 2, which shows no outliers.

The SSIM values between the original T1-weighted and synthesised T1-weighted images had a mean of 0.35 ± 0.66 with a range from 0.17 to 0.48 (Table 2). As with the MSE values, there were no outliers for SSIM values (Figure 3).



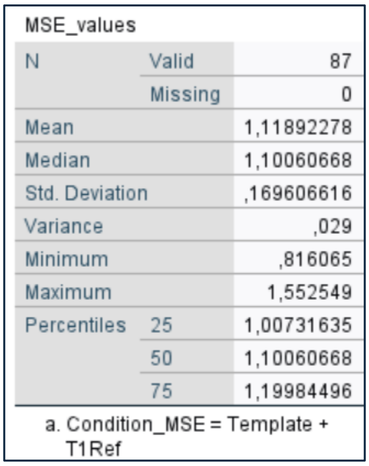
| SSIM_values | | |
|---|---|---|
| N | Valid | 87 |
| | Missing | 0 |
| Mean | | ,34507224 |
| Median | | ,34511166 |
| Std. Deviation | | ,066298671 |
| Variance | | ,004 |
| Minimum | | ,169053 |
| Maximum | | ,482798 |
| Percentiles | 25 | ,30125940 |
| | 50 | ,34511166 |
| | 75 | ,39892695 |

a. Condition_SSIM = T1T1

*Table 2: Descriptive statistics of SSIM-values between original T1-weighted and synthesized T1-weighted images*

### Comparison of Original T1-weighted Images and the Template Image

The results for the comparison between the original T1-weighted images and the template image are shown in Table 3 and Table 4.



| MSE_values | | |
|---|---|---|
| N | Valid | 87 |
| | Missing | 0 |
| Mean | | 1,11892278 |
| Median | | 1,10060668 |
| Std. Deviation | | ,169606616 |
| Variance | | ,029 |
| Minimum | | ,816065 |
| Maximum | | 1,552549 |
| Percentiles | 25 | 1,00731635 |
| | 50 | 1,10060668 |
| | 75 | 1,19984496 |

a. Condition_MSE = Template + T1Ref

*Table 3: Descriptive statistics of MSE-values between the template image and the original T1-weighted images.*

The mean MSE value was 1.12 ± 0.17, with a range from 0.82 to 1.55 (Table 3). The mean SSIM value was 0.32 ± 0.06 with a range from 0.11 to 0.41 (Table 4).

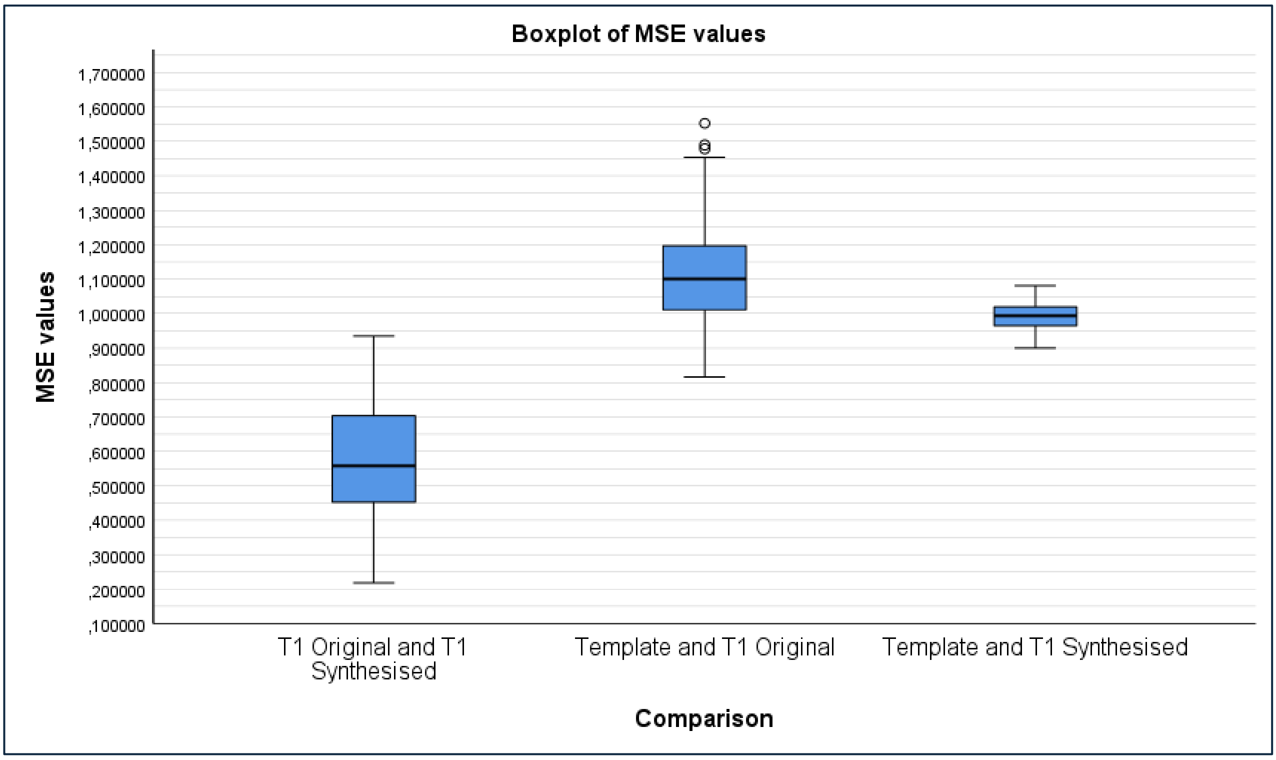


*Figure 2: Boxplot of MSE values for the three comparisons: Original T1-weighted images compared to synthesised T1-weighted images, original T1-weighted images compared to template image, and synthesised T1-weighted images compared to template image.*

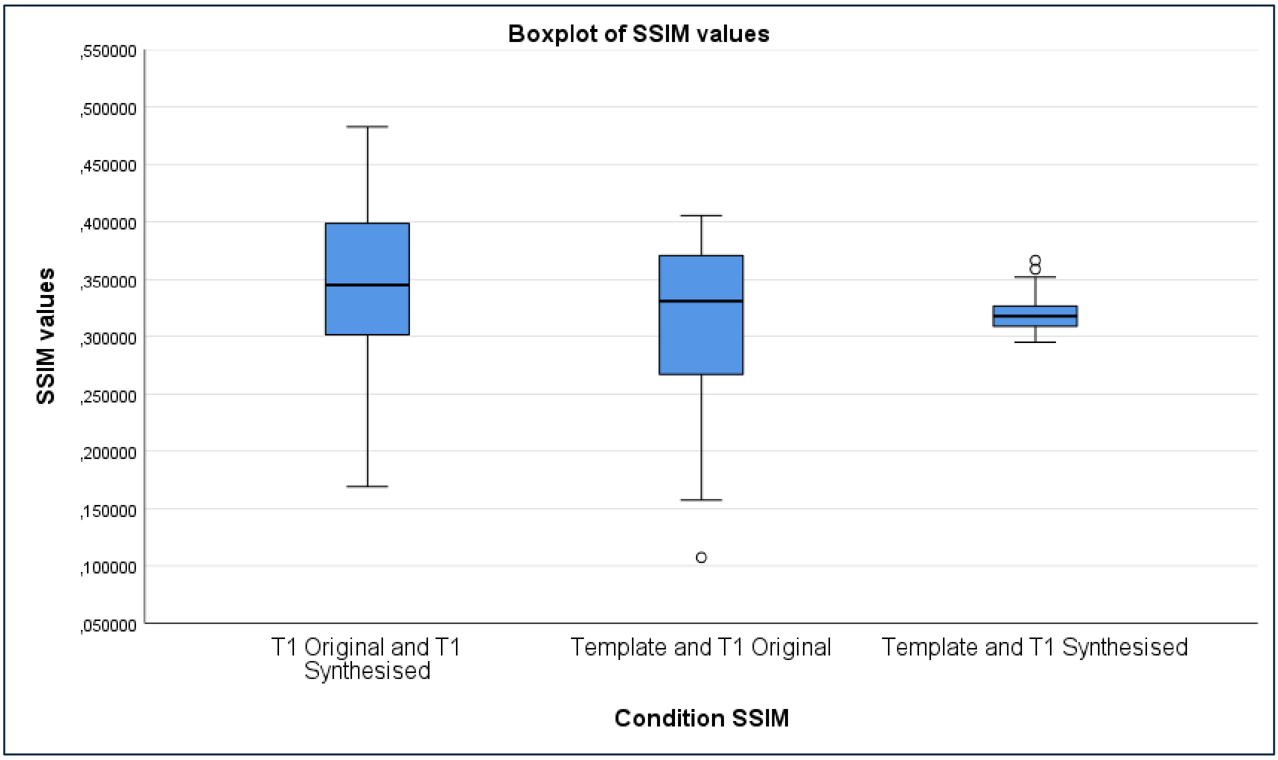
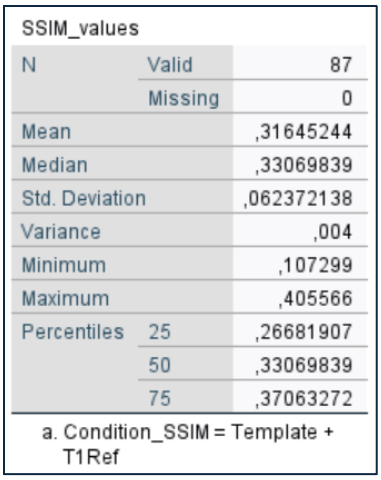


*Figure 3: Boxplot of SSIM values for the three comparisons: Original T1-weighted images compared to synthesised T1-weighted images, original T1-weighted images compared to template image, and synthesised T1-weighted images compared to template image.*



| SSIM_values | | |
|---|---|---|
| N | Valid | 87 |
| | Missing | 0 |
| Mean | | ,31645244 |
| Median | | ,33069839 |
| Std. Deviation | | ,062372138 |
| Variance | | ,004 |
| Minimum | | ,107299 |
| Maximum | | ,405566 |
| Percentiles | 25 | ,26681907 |
| | 50 | ,33069839 |
| | 75 | ,37063272 |

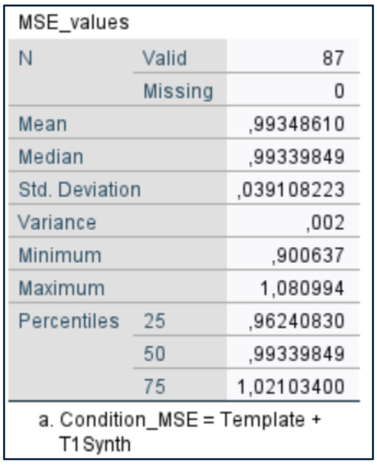
a. Condition_SSIM = Template + T1Ref

*Table 4: Descriptive statistics of SSIM-values between the template image and the original T1-weighted images*

The boxplot in Figure 2 and Figure 3 visually presents the results, and outliers were observed for both MSE and SSIM values. For the MSE values, three outliers had a higher value than the upper IQR, which was around 1.25 (Figure 2). For the SSIM values, one outlier fell below the lower bound of the IQR, which was around 0.15 (Figure 3).

## Comparison of Synthesised T1-weighted Images and the Template Image

The results of MSE measurement between the synthesised T1-weighted images and the template image are shown in Figure 5. The mean MSE value was 0.99 ± 0.04 with a range from 0.90 to 1.08 (Table 5).

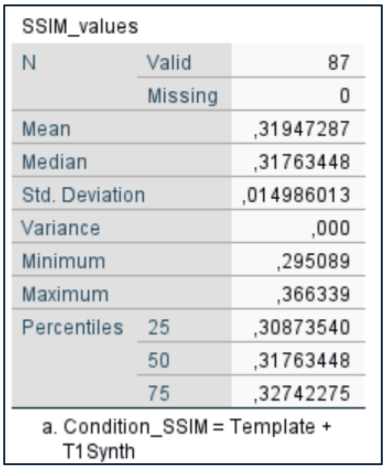


| MSE_values | | |
|---|---|---|
| N | Valid | 87 |
| | Missing | 0 |
| Mean | | ,99348610 |
| Median | | ,99339849 |
| Std. Deviation | | ,039108223 |
| Variance | | ,002 |
| Minimum | | ,900637 |
| Maximum | | 1,080994 |
| Percentiles | 25 | ,96240830 |
| | 50 | ,99339849 |
| | 75 | 1,02103400 |

a. Condition_MSE = Template + T1Synth

*Table 5: Descriptive statistics of MSE-values between the template image and the synthesised T1-weighted images.*

For the SSIM values, the results are shown in Figure 6. The mean SSIM value was 0.32 ± 0.01 with a range from 0.30 to 0.37 (Table 6).



| SSIM_values | | |
|---|---|---|
| N | Valid | 87 |
| | Missing | 0 |
| Mean | | ,31947287 |
| Median | | ,31763448 |
| Std. Deviation | | ,014986013 |
| Variance | | ,000 |
| Minimum | | ,295089 |
| Maximum | | ,366339 |
| Percentiles | 25 | ,30873540 |
| | 50 | ,31763448 |
| | 75 | ,32742275 |

a. Condition_SSIM = Template + T1Synth

*Table 6: Descriptive statistics of SSIM-values between the template image and the synthesised T1-weighted images.*

Figure 2 and Figure 3 displays the boxplot for the MSE and SSIM values between the template and the synthesised T1-weighted images. This revealed no outliers for the MSE values but showed two outliers for the SSIM values. These outliers were just above the upper bound of the IQR, which was around 0.34 (Figure 3).

## Subjective Results

Figure 4 displayed four image pairs, each consisting of an original T1-weighted image and its corresponding synthesised T1-weighted image after registration and normalisation. The first image pair exhibited the highest MSE value (0.93) and the lowest SSIM value (0.17) (Figure 4a). The second image pair had the lowest MSE value (0.22) (Figure 4b), while the third image pair showed the highest SSIM value (0.48) (Figure 4c). The final image pair presented with a moderate MSE value (0.56) and a moderate SSIM value (0.36) reflecting the study's overall findings (Figure 4d).

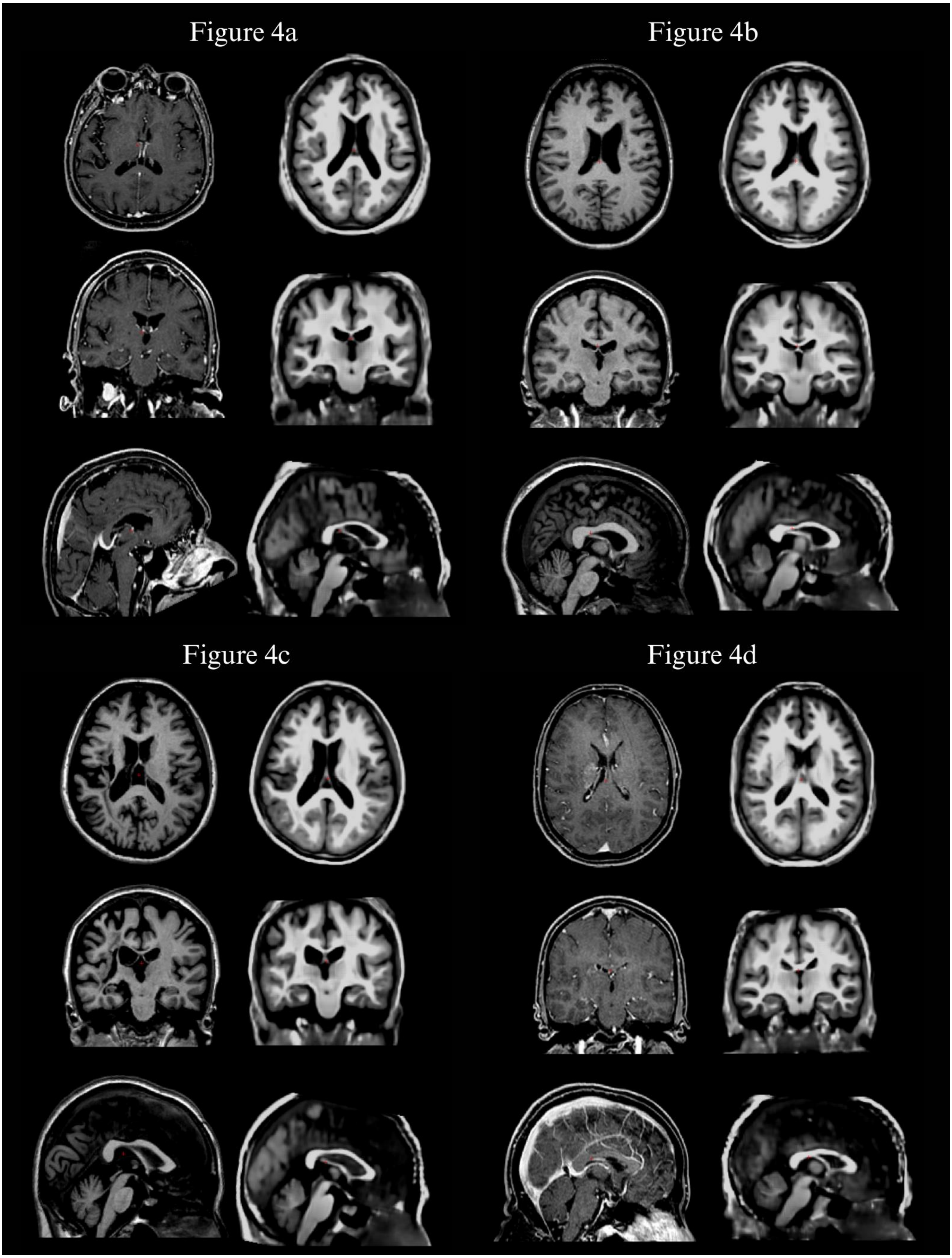


*Figure 4: Four image pairs with the original T1-weighted images to the left and the synthesized T1-weighted images to the right, for visual comparison. 4a) Shows the image pair with the highest MSE-value and lowest SSIM-value. 4b) Shows the image pair with the lowest MSE-value. 4c) Shows the image pair with the highest SSIM-value. 4d) Shows the image pair with a moderate MSE-value and moderate SSIM value.*

## Discussion

This study investigated the similarity of synthesised T1-weighted images compared to original T1-weighted images. The results showed that the synthesised T1-weighted images and original T1-weighted images had a mean MSE value of $0.56 \pm 0.17$, which indicates an average moderate symmetry between the pixel intensity in the images. This result means that the pixel values of the synthesised images show numerical deviations from the original images. Although there were a few images where the pixel intensity was very similar between the synthesised and original images, with an MSE value around 0.2. These images would be more pixel-wise identical. However, it should be noted that MSE is sensitive to intensity shifts and noise whilst ignoring structural similarity (31). This means that a low MSE does not necessarily guarantee a visually identical image, if the structural integrity is compromised. Hence SSIM was also measured between the synthesised and original images.

The average SSIM value between the synthesised images and the original images was $0.35 \pm 0.07$ which means the images have a low structural similarity and will be different in their fundamental structure. Similarly, the visual image quality appears poor, as illustrated in Figure 4, due to the substantial difference between the original and synthesised T1-weighted images. The image pair exhibiting the highest structural similarity, as indicated by its SSIM score, achieved an SSIM value of 0.48. Even this best-case value, however, suggests that SynthSR struggled to achieve high structural fidelity within the dataset, even under optimal conditions. While not perfect, a visual inspection of these images reveals a generally comparable overall structure (Figure 4). This contrasts sharply with image pairs demonstrating the lowest structural similarity, where an SSIM value of 0.17 resulted in images that appeared almost entirely dissimilar upon visual inspection (Figure 4). From this perspective, the image pair with the highest SSIM-value, despite its low score, visually represents an improvement when compared to those with the lowest SSIM values. Crucially, the combination of high MSE and low SSIM values highlights that while MSE quantifies numerical pixel-level deviations, SSIM provides critical insight into the preservation of underlying anatomical structures, which are vital for human visual perception (31). Together, these metrics unequivocally indicate that SynthSR is not performing optimally in terms of reconstructing the original MRI image. This is because the results show that the synthesised T1-weighted images exhibit considerable deviations from the original images, both numerically and structurally.

Beyond the comparison between the original T1-weighted and the synthesised T1-weighted images, an additional analysis was conducted to assess the similarity between the template image and the original T1-weighted images. The mean MSE value for this comparison was $1.12 \pm 0.17$. This result indicates a substantial average pixel-to-pixel deviation, demonstrating that the pixel values of the original images numerically differ noticeably from those of the template image. This finding holds true for the majority of the images, as evidenced by a minimum observed MSE value of 0.82, which itself suggests a low similarity and large differences in pixel intensity even for the most similar pairs within this specific comparison.

For the same comparison, the mean SSIM value was $0.32 \pm 0.06$. As this result is low, the template image and the original T1-weighted images have poor structural similarity, and image information such as tissue interfaces, anatomical structures, and fine details are likely not well preserved.

Additionally, the template image was compared to the synthesised T1-weighted images. The mean MSE value for this comparison was $0.99 \pm 0.04$, indicating a high numerical deviation. The minimum MSE value was 0.90, which suggests that all the synthesised T1-weighted images differ from the template image and have varying intensity scales.

Furthermore, the mean SSIM value for this comparison was $0.32 \pm 0.01$, with the highest value being 0.37. This result indicates that the synthesised images and the template are substantially different and have a low structural similarity.

An analysis of the MSE values revealed distinct differences when comparing images to the template. Specifically, the comparison between the template and the original T1-weighted images yielded a mean MSE of 1.12. In contrast, the synthesised T1-weighted images, when compared to the same template, demonstrated a slightly lower average MSE of 0.99, suggesting a marginally improved pixel-wise similarity in this comparison. This could suggest that the synthesised T1-weighted images can be registered to the template with comparable or even improved accuracy relative to the original T1-weighted images. Nevertheless, the observed MSE values persist at a relatively high level, implying substantial disparities in pixel intensity between the template and the synthesised T1-weighted images.

Based on the obtained MSE and SSIM values, the results of this study indicated both numerical differences in pixel intensity and low structural similarity when comparing the synthesised T1-weighted images with the original T1-weighted images, as well as with the template image. Refinements in the methodology might yield better results, such as how the normalisation was performed.

Osman et al. developed a CNN-based U-net model designed for synthesising T1-weighted images from FLAIR, among other modalities (13). A critical step in their methodology involved pre-synthesis normalisation, where data underwent both Z-score normalisation and subsequent scaling to the [0,1] intensity range. They compared the synthesised T1-weighted images with original T1-weighted images by calculating MSE and SSIM, among other metrics. The average MSE and SSIM values of $(0.0009, 0.95)$ (13) were substantially better than those of this study $(0.56, 0.45)$. In a separate study by Sharma et al., a Generative Adversarial Network (GAN) model was proposed for the synthesis of missing MRI sequences. Consistent with the approach of Osman et al., image normalisation was performed prior to the synthesis of the MRI images. Although they did not synthesise T1-weighted images from FLAIR, but rather from T2-weighted images, they obtained considerably improved SSIM values of 0.93 (11), compared to the SSIM of this study 0.45.

Considering the distinct synthesis models employed, which may explain discrepancies in results, it would be valuable to investigate the effects of pre-synthesis normalisation within the SynthSR framework, potentially involving data scaling to the [0,1] intensity range.

Iglesias et al. developed SynthSR with the aim of transforming clinical brain scans, regardless of their MRI contrasts, orientation, and resolution, into high-resolution T1-weighted scans (6). The tool was extensively tested and validated on more than 10,000 brain scans. This validation included segmenting and registering the synthesised images, which showed a high correlation with actual high-resolution scans. Furthermore, the synthesised T1-weighted images from SynthSR proved to be compatible with existing neuroimaging tools (6).

Although this study did not perform direct quality analysis of the synthesised T1-weighted images through downstream methods like segmentation, their suitability for registration was assessed. Both the synthesised T1-weighted images and the original T1-weighted images were registered to a common template. The synthesised T1-weighted images exhibited a lower MSE-value $(0.99, 1.12)$ and an identical SSIM-value $(0.32, 0.32)$ compared to the original T1-weighted images. This marginally improved pixel-wise similarity and comparable structural alignment is a key finding for registration purposes. It suggests that while the synthesis might introduce some pixel-level deviations from the original, it could simultaneously produce images with properties, such as reduced noise or a more consistent intensity scale, that are advantageous for standard template registration. This finding aligns with SynthSR's demonstrated utility and compatibility for robust registration in neuroimaging pipelines, indicating the synthesised T1-weighted images are equally effective for this purpose as the original T1-weighted images (6).

## Limitations

This study has some limitations that could affect the results. First, the data used for synthesising came from different MRI scanners, and it was unknown if the protocols had been consistent. Images from different scanners or with different protocols can have varying image properties, which could impact the registration, normalisation or synthesis. The synthesis in this study is particularly sensitive to whether the input images originate from a low-field or high-field scanner. Given that the scanner origin was unknown for the images, all were processed with the low-field flag. Therefore, the synthesis results in this study could potentially vary if this information had been known and applied. Another limitation is the absence of downstream evaluation, such as using the synthesised images for segmentation and comparing the results to segmentations derived from the original images. Until this is done, it remains unknown if the synthesised T1-weighted images can be used for further neuro-analysis.

## Conclusion

This study validated SynthSR capability to synthesise T1-weighted images from FLAIR. Furthermore, the synthesised T1-weighted images were validated by comparing them with original T1-weighted images as the ground truth. SynthSR synthesised T1-weighted images from FLAIR, that visually resembled the original T1-weighted images. However, objective image quality measurements did show differences in both pixel intensity and the structural similarity between the synthesised T1-weighted images and the original T1-weighted images. This indicates a need for further research to either optimise the synthesised images or to determine, if they are sufficient for neuroimaging-analysis.

## Conflicts of interest

The author does not have any conflicts of interest to declare.

## Fundings

This project did not receive any kind of funding

## Acknowledgement

The author would like to thank Lasse Riis Østergaard and Anders Ulrich Eriksen for providing supervising to the project.

## References

1. Delboni Lemos M, Faillenot I, Tavares Lucato L, Jacobsen Teixeira M, Mendonça Barbosa L, Joaquim Lopes Alho E, et al. Dissecting neuropathic from poststroke pain: The white matter within. Pain. 2022 Apr 1;163(4):765–78.

2. World Health Organization. The top 10 causes of death [Internet]. 2024 [cited 2025 Mar 12]. Available from: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

3. Kazam JJ, Tsiouris AJ. Brain Magnetic Resonance Imaging for Traumatic Brain Injury: Why, When, and How? Top Magn

Reson Imaging [Internet]. 2015 Oct;24(5):225–39. Available from: www.topicsinmri.com

4. Symms M, Jäger HR, Schmierer K, Yousry TA. A review of structural magnetic resonance neuroimaging. Vol. 75, Journal of Neurology, Neurosurgery and Psychiatry. BMJ Publishing Group; 2004. p. 1235–44.

5. Westbrook C, Talbot J. MRI in Practice. 5th ed. Wiley blackwell; 2018.

6. Iglesias JE, Billot B, Balbastre Y, Magdamo C, Arnold SE, Das S, et al. SynthSR: A public AI tool to turn heterogeneous clinical brain scans into high-resolution T1-weighted images for 3D morphometry. Sci Adv [Internet]. 2023 Feb 1;9(5). Available from: https://www.science.org

7. McNaughton J, Holdsworth S, Chong B, Fernandez J, Shim V, Wang A. Synthetic MRI Generation from CT Scans for Stroke Patients. BioMedInformatics. 2023 Sep 1;3(3):791–816.

8. Vert C, Parra-Fariñas C, Rovira À. MR imaging in hyperacute ischemic stroke. Vol. 96, European Journal of Radiology. Elsevier Ireland Ltd; 2017. p. 125–32.

9. Larsen E. MR - 500A Cerebrum Stroke. 2023 Mar.

10. Yavarian Y. MR – Cerebrum – Stroke FAST Track [Internet]. 2024 Jun. Available from: https://pri.rn.dk/Sider/32693.aspx

11. Sharma A, Hamarneh G. Missing MRI Pulse Sequence Synthesis using Multi-Modal Generative Adversarial Network. IEEE Trans Med Imaging [Internet]. 2020 Apr 27;39(4):1170–83. Available from: http://arxiv.org/abs/1904.12200

12. Iglesias JE, Billot B, Balbastre Y, Tabari A, Conklin J, Gilberto González R, et al. Joint super-resolution and synthesis of 1 mm isotropic MP-RAGE volumes from clinical MRI exams with scans of different orientation, resolution and contrast. Neuroimage. 2021 Aug 15;237.

13. Osman AFI, Tamam NM. Deep learning-based convolutional neural network for intramodality brain MRI synthesis. J Appl Clin Med Phys. 2022 Apr 1;23(4).

14. Conte GM, Weston AD, Vogelsang DC, Philbrick KA, Cai JC, Barbera M, et al. Generative adversarial networks to synthesize missing T1 and FLAIR MRI sequences for use in a multisequence brain tumor segmentation model. Radiology. 2021;299(2):313–23.

15. Mazziotta JC, Toga AW, Evans A, Fox P, Lancaster J. A probabilistic atlas of the human brain: Theory and rationale for its development. Neuroimage [Internet]. 1995 [cited 2025 Jun 1];2(2):89–101. Available from: https://pubmed.ncbi.nlm.nih.gov/9343592/

16. Giff A, Noren G, Magnotti J, Lopes AC, Batistuzzo MC, Hoexter M, et al. Spatial normalization discrepancies between native and MNI152 brain template scans in gamma ventral capsulotomy patients. Psychiatry Res Neuroimaging [Internet]. 2023 Mar 1 [cited 2025 Jun 1];329:111595. Available from: https://www.sciencedirect.com/science/article/pii/S0925492723000057

17. Registration — ANTsPy dev (latest) documentation [Internet]. [cited 2025 Jun 1]. Available from: https://antspy.readthedocs.io/en/latest/registration.html

18. Avants BB, Tustison N, Johnson H. Advanced Normalization Tools (ANTS) Release 2.x [Internet]. 2014. Available from: https://brianavants.wordpress.com/2012/04/13/updated-ants-compile-instructions-april-12-2012/

19. Source code for ants.registration.registration [Internet]. [cited 2025 Jun 1]. Available from: https://antspy.readthedocs.io/en/latest/_modules/ants/registration/registration.html#registration

20. NiBabel developers. Streamlines [Internet]. 2006 [cited 2025 Jun 1]. Available from: https://nipy.org/nibabel/reference/nibabel.streamlines.html

21. NiBabel developers. dataobj_images [Internet]. 2006 [cited 2025 Jun 1]. Available from: https://nipy.org/nibabel/reference/nibabel.dataobj_images.html#nibabel.dataobj_images.DataobjImage.get_fdata

22. NiBabel developers. nifti1 [Internet]. 2006 [cited 2025 Jun 1]. Available from: https://nipy.org/nibabel/reference/nibabel.nifti1.html#nibabel.nifti1.Nifti1Image

23. NumPy Developers. NumPy reference — NumPy v2.2 Manual [Internet]. 2024 [cited 2025 Jun 1]. Available from: https://numpy.org/doc/stable/reference/index.html#reference

24. scikit-learn developers. StandardScaler — scikit-learn 1.6.1 documentation [Internet]. 2025 [cited 2025 Jun 1]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

25. Kowalik-Urbaniak IA, Castelli J, Hemmati N, Koff D, Smolarski-Koff N, Vrscay ER, et al. Modelling of subjective radiological assessments with objective image quality measures of brain and body CT images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) [Internet]. 2015 [cited 2025 Mar 19];9164:3–13. Available from: https://link.springer.com/chapter/10.1007/978-3-319-20801-5_1

26. Kowalik-Urbaniak I, Brunet D, Wang J, Koff D, Smolarski-Koff N, Vrscay ER, et al. The quest for "diagnostically lossless" medical image compression: a comparative study of objective quality metrics for compressed medical images. In: Medical Imaging 2014: Image Perception, Observer Performance, and Technology Assessment. SPIE; 2014. p. 903717.

27. Yaniv Z, Lowekamp BC, Johnson HJ, Beare R. SimpleITK Image-Analysis Notebooks: a Collaborative Environment for Education and Reproducible Research. J Digit Imaging. 2018 Jun 1;31(3):290–303.

28. NumPy community. NumPy Reference Release 2.2.0 Written by the NumPy community. 2025.

29. McKinney W. User Guide — pandas 2.2.3 documentation [Internet]. 2024 [cited 2025 Jun 1]. Available from: https://pandas.pydata.org/docs/user_guide/index.html#user-guide

30. The scikit-image team. skimage.metrics — skimage 0.24.0 documentation [Internet]. 2024 [cited 2025 Jun 1]. Available from: https://scikit-image.org/docs/0.24.x/api/skimage.metric

s.html#skimage.metrics.structural_similarity

31. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing. 2004 Apr;13(4):600–12.

## Worksheets

To enhance your understanding of the presented material, these supplementary worksheets fill gaps left by the scientific paper. They align with current learning objectives in this semester by offering detailed process descriptions and results not included in the original publication.

## Table of contents for worksheets

## Worksheets 1: Background information

In this worksheet, further background information for the scientific problem is giving.

### Stroke

Annually, 12.000 Danes are hospitalized with a stroke in Denmark, and it stands as the fourth most frequent cause of death (32). Treatment for stroke involves either thrombolysis or thrombectomy, which must begin as soon as possible (32).

Thrombolysis medically dissolves the thrombus by administering 0.9 mg pr kg of alteplase intravenously to the patient (32,33). However, this treatment is only effective if administered within three to four a half hours of symptom onset (32–34). Thrombectomy, conversely, is a minor surgical intervention that removes the thrombus and must be initiated within six hours of symptom onset (32). Earlier treatment, regardless of type, leads to fewer patients requiring post-treatment assistance (34,35).

To initiate the treatments as fast as possible, stroke must be detected by either CT- or MR-scanning (36). MRI offers superior sensitivity for demonstrating ischemia and can identify it more quickly than CT. Additionally, MRI yields insights into the infarct's size and mechanism, information not provided by CT (8,36).

Vert et al. argue for which sequences should be included in an MRI scan to detect information about the presence and location of an intravascular thrombus that requires treatment, the size of irreversibly infarcted tissue, and the presence of hypo perfused tissue. These sequences are T2*, MRA, DWI, T2-FLAIR and PWI (8).

At Aalborg University hospital, MRI can also be used for stroke, where the following sequences are included; DWI, T2 Flair fs, T2*, 3D ASL without contrast, TOF and T2 FSE (9,10).

### MRI sequences

#### T2*

The T2*-sequence is a gradient-echo sequence, utilizing a gradient pulse rather than an radiofrequency pulse (RF pulse). It offers a short scanning time as both repetition time (TR) and echo time (TE) are briefer due to faster magnetization and dephasing compensation (37).

Vert et al. contend that T2*-weighted sequences ought to be part of the MRI protocol for stroke. They are particularly sensitive for detecting both acute and chronic intracerebral haemorrhage, as they exploit deoxyhaemoglobin's paramagnetic effect, which creates hypointensity in blood-filled areas. Furthermore, they can identify microbleeds and hypointense blood vessels in ischaemic areas, serving as diagnostic markers for stroke (8).

### Diffusion Weighted Imaging (DWI)

Diffusion Weighted imaging (DWI) is a fast MRI sequence that uses pulses sensitive to the diffusion of water molecules. This makes the sequence particularly useful for detecting ischaemic damage in the brain within minutes of a thrombus (8,37). Furthermore, DWI can identify subclinical satellite ischaemic lesions, which can provide information about the stroke's mechanism (8). Fiebach et al. likewise conclude that DWI has become the most sensitive (88 % to 100 %) and specific (95 % to 100 %) imaging technique for detecting acute infarction, even very early after symptom onset, when comparing MRI-DWI and CT (38).

### Flair

FLAIR is used to visualize pathological changes in the brain, as it suppresses the signal from free fluid, such as cerebrospinal fluid (CSF), making it easier to detect lesions, particularly in white matter. To suppress CSF, a long inversion time (TI) is required, as the CSF must reach zero magnetisation, along with a long repetition time (TR) to preserve the T2 contrast (37). This sequence is useful for detecting ischaemic lesions, precisely because the signal from CSF is suppressed, and for visualising vascular hyperintensities, which are used as an indirect indicator for collateral assessment (8).

Furthermore, FLAIR, along with DWI, is used to estimate the age of the infarct, which is important in relation to further treatment (8).

### Time of Flight (TOF)

This sequence is used to visualize blood vessels. This works by applying an inversion pulse in the area where the blood is to be "labelled". This pulse alters the magnetism in the blood's protons, thereby giving the blood a stronger signal compared to the surrounding tissue. After the scan, the moving blood can be separated from the stationary surrounding tissue, based on the different signals detected (37).

Vert et al. contend that this sequence is useful in an MRI stroke protocol, as it is particularly valuable for detecting vascular occlusion and/or stenosis in patients with acute ischaemic stroke. The detection of this is crucial for subsequent clinical decisions (8).

### Perfusion Weighted Imaging (PWI)

This sequence is used to identify areas in the brain with reduced blood flow due to arterial blockages, and to distinguish between the infarct core and the ischaemic penumbra. For this sequence, a gadolinium-based contrast agent is injected, which then causes signal changes, thereby providing information about blood flow and blood volume in the brain (8).

Vert et al. further estimate that this sequence can be advantageously used together with DWI to differentiate the infarct core from the penumbra (8).

### 3D Arterial Spin Labelling (ASL) without contrast

This sequence is used to measure cerebral blood flow (CBF). This is possible by applying an RF pulse to the blood in the neck or carotid artery, thereby "labelling" the blood and altering its magnetic properties. The signal from the labelled blood cells will consequently have different signals than the unlabelled blood cells when they reach the area where the signal is measured. The change between the labelled and unlabelled blood cells can be used to calculate CBF and can provide information about how quickly blood is moving through the brain's tissue (37).

### T2 Fast Spin Echo (FSE)

This sequence uses multiple 180° RF pulses rather than just one, which makes the sequence fast. It also utilizes T2-relaxation where fluid will exhibit a high signal, as protons in tissue with high water content lose their signal more slowly before the next RF pulse is transmitted (37).

Therefore, this sequence is good for visualizing tissues with high water content, such as stroke and oedema (37).

### T1-weighted

T1-weighted images are primarily used to visualise anatomy and pathology. They are often used for mapping or creating atlases, as their contrast primarily depends on the differences in the T1 recovery times between fat and water. This is achieved by using a short TR and a short TE, which results in fat appearing bright (hyperintense) and water appearing dark (hypointense) on the image (37).

## Worksheet 2: Conduction and documentation of the systematic literature review

Initially initial literature searches were conducted to gather information, identify knowledge gaps and to formulate the aim of this study. Furthermore, the initial literature searches served to find keywords, synonymous and thesauri applicable to the systematic literature search.

The systematic literature search was conducted on background of the initial literature search. They included scientific articles from the systematic literature search was used in the discussion of this study.

The systematic literature search was conducted through search blocks (Table A). Each block was compiled by the Boolean operator AND, and the search terms were compiled by the Boolean operator OR. The search terms consisted of both keywords, thesauri and synonymous, and truncation and phrasing were applied in the searches.

| Block 1 | Block 2 | Block 3 | Block 4 |
|---|---|---|---|
| Brain /exp 4 synonyms<br>Stroke :all<br>Ischemia /exp 21 synonyms | Nuclear Magnetic Resonance /exp 5 synonyms<br>Nuclear Magnetic Resonance imaging /exp 8 synonyms<br>Diagnostic imaging /exp 4 synonyms<br>Neuroimaging /exp 3 synonyms<br>MRI :all | Deep Learning /exp 4 synonyms<br>Neural Network /exp<br>Machine Learning /exp 3 synonyms<br>Artificial intelligence /exp 2 synonyms<br>Convolutional neural network /exp 13 synonyms<br>Generative adversarial network /exp 2 synonyms | Synthesizing :all<br>SynthSR :all<br>Synthesis /exp<br>Synthetic :all<br>Modality translation :all<br>Image-to-image translation /exp<br>Cross-modality synthesis :all |

*Table A: Search block for the systematic literature search*

The systematic literature searches were conducted in Embase and IEEE Xplore. Embase was selected because it represents one of the largest bibliographic databases in the fields of medicine and health sciences and is commonly employed by the author (39). IEEE Xplore was selected because it provides access to more than three million documents in electronic and computer science (40).

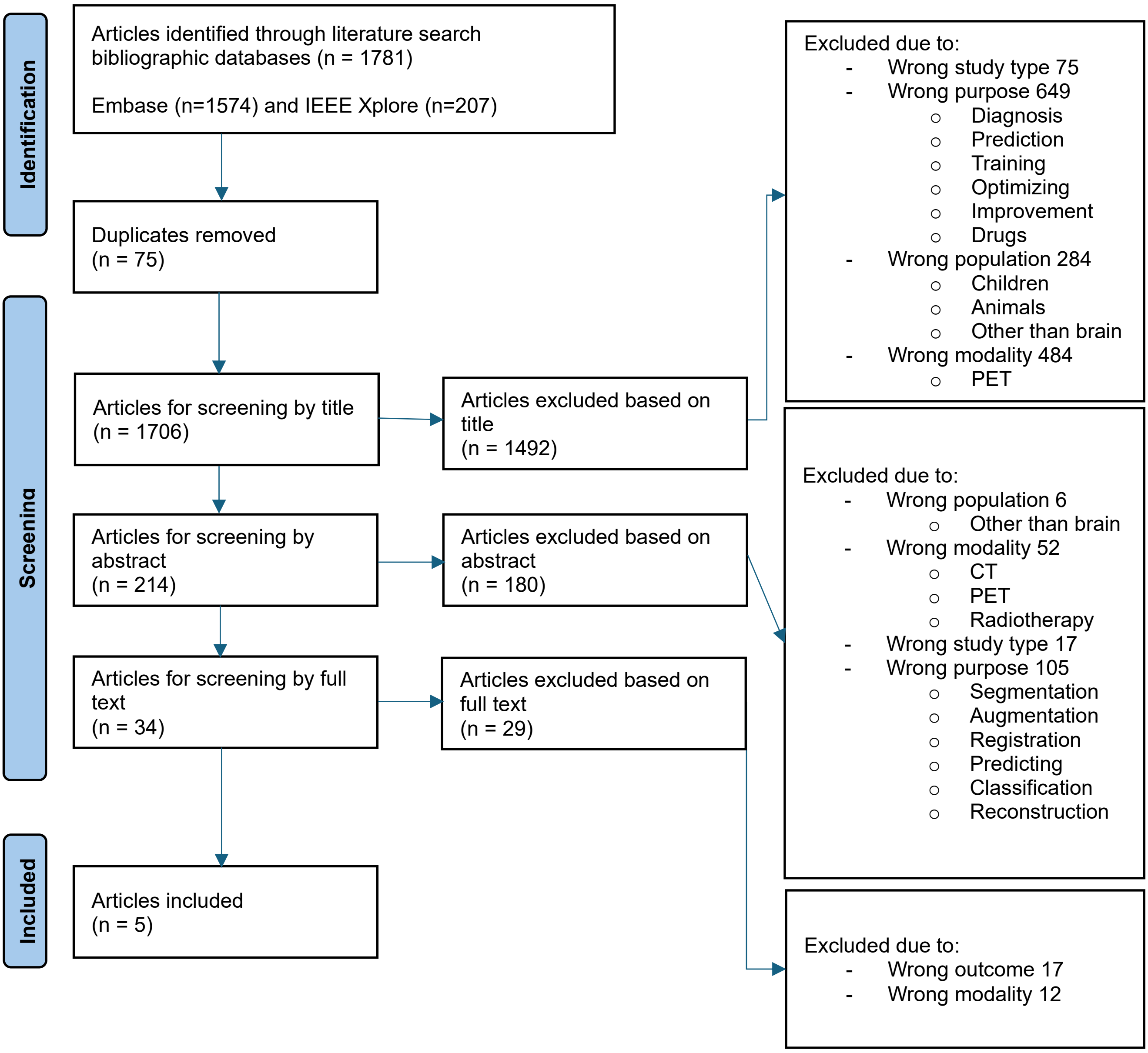


*Figure A: Flowchart for the selection process of the systematic literature searches*

The selection process was made by the author, where duplicates were removed at first, and then the articles was screened by title, followed by screening of abstract and last screening by full text (Figure A). The exclusion criterions were used for the selection of articles and was primarily wrong purposes or outcome regarding this project. This could be articles with focus on prediction of diseases or aging of tumours, optimizing of scan protocols or treatments, training of the healthcare professionals, reconstruction of image or classification of pathology. The rest of the criterions are showed both in Figure A, but also in the table with both inclusion and exclusion criterions (Figure A, Table B).

| Inclusion | Exclusion |
|---|---|
| Population:<br>- Adults 18+ years<br>- Brain images<br>- Stroke<br>Modality:<br>- MRI<br>Purpose:<br>- Synthesizing of images<br>- Handling missing MRI data<br>- Image measurements | Study type:<br>- Pilot study<br>- Protocol<br>- Expert article<br>- In vitro<br>Population:<br>- Animals<br>- Children<br>- Images of other parts than the brain<br>Modality:<br>- PET<br>- Radiotherapy<br>- CT<br>- X-ray<br>Purpose:<br>- Prediction<br>- Diagnosis<br>- Classification<br>- Training<br>- Optimizing<br>- Improvement<br>- Segmentation<br>- Augmentation<br>- Registration<br>- Reconstruction<br>- Synthesizing of chemical drugs |

*Table B: Criterions for the systematic literature search*

The criterions about modality were made, because it was essential to find the studies that already had worked with MRI images, because this is the purpose of this project. Furthermore, some of the purpose criterions was made under the selection process. Segmentation, among other things, was one of the criterions that was made under the selection process, because there were enough articles about synthesizing that fit the purpose of the project better, than articles about segmentation.

At last, five articles were chosen to use both in the introduction and in the discussion for this project (Figure A). The articles were:

- SynthSR: A public AI tool to turn heterogeneous clinical brain scans into high-resolution T1-weighted images for 3D morphometry (6)

- Missing MRI Pulse Sequence Synthesis using Multi-Modal Generative Adversarial Network (11)
- Joint super-resolution and synthesis of 1 mm isotropic MP-RAGE volumes from clinical MRI exams with scans of different orientation, resolution and contrast (12)
- Deep learning-based convolutional neural network for intramodality brain MRI synthesis (13)
- Generative adversarial networks to synthesize missing T1 and FLAIR MRI sequences for use in a multisequence brain tumour segmentation model (14)

Other articles in this project were found through snowballing. By this method, articles were found in references in articles from the systematic literature search and by looking at other articles that have cited an article from the systematic literature search.

## Worksheet 3: Understanding Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep learning neural network. It is used for various task, such as image classification and image segmentation (41). They are designed to handle data presented as multiple arrays, such as images (42) and can have hundreds of layers (41). The three main layers are convolutional layer, Pooling layer and Fully Connected layer (42). The first layer, Convolutional layer is a fundamental building block. The purpose of this layer is to apply small, learnable filters to the input data to detect specific patterns or features in the data. Each filter activates when it recognises a pattern it is trained to recognise. Neurons in this layer is only connected with a small array of the input, which effectively captures the spatial relationship in the input data. Furthermore, one filter can recognise the learned feature across the entire input, which prevent overfitting. The result of the convolutional layer is multiple feature maps (41,42).

The next layer, Pooling Layer, reduces the spatial dimensions of the feature maps from the Convolutional layers, by downsampling the maps. The Pooling layer works by considering a small, confined area of the input map and produce a singular output value for that area, leading to a reduction in the map's scale. It helps reducing the risk for overfitting because the network gets less sensitive to small variations in the input data, which makes the model robust to noise and small shifts or deformation and becomes less sensitive to the exact location of a feature within the input (41,42).

The last main layer, Fully Connected layer is often the last layer in CNN. Essentially, it takes the maps from the Pooling layer and converts their features into a final prediction or decision, specifically by mapping them to probabilities or values for classification or regression (41,42). Though, the output from the Pooling layer must be flattened into a single long vector for the Fully Connected layer can work with it. In the Fully Connected layer, all neurons are connected in the preceding layer, which enables it to learn complex, non-lineary combination of the features extracted from earlier layers (42).

## Worksheet 4: Scripts

In this worksheet, the scripts used in this master's thesis are described.

### Synthesising

To synthesize FLAIR images to T1 images, SynthSR was used as mentioned in the article for this project. Furthermore, it was performed in Python.

The commando for synthesizing with SynthSR was:

```
mri_synthsr --i <input> --o <output> --threads <n_threads> [--v1] [--lowfield] [--ct]
```

After insertion of the path for SynthSR, the input image and the output image, the commando, for this project, would be:

```
/usr/local/freesurfer/8.0.0/bin/mri_synthsr --i /home/maria/Pictures/Data/Input/001/flair.nii --o /home/maria/Pictures/Data/Output/001 --lowfield --threads 32.
```

The lowfield flag was used, because the data in this project originated from both 1.5 and 3 Tesla MRI scans. Because there was no information about which type of MRI scanner the images came from, the flag was used to ensure that the results were the best as possible for all the images.

The v1 flag was not used, because there is a newer version of SynthSR, which made better synthesized images visual. Therefore, the newest version, version 2, was used automatically when no flag was used. Furthermore, the ct flag was not used, because it was only MRI scans that were processed.

At last, the n_threads flag was used to make the process faster. The computer that was used for synthesizing had 32 CPU threads, why it was chosen.

(6,43).

### Registration

```
import os
import ants


def register_pipeline(
    flair, t1, mni, output_path=None, flair_t1_registered=True, verbose=False
):
    """
    Pipeline to register FLAIR → T1 → MNI152.
    Applies transformations to align FLAIR with MNI space using either original or registered T1.
    """
```

```
def out(prefix):
    return os.path.join(output_path, prefix) if output_path else prefix
#Step 0: Resample T1 to MNI space
t1_resampled = ants.resample_image_to_target(t1, target=mni, interp_type="bSpline")
#Step 1: Register FLAIR → T1 (if not already registered)
flair_to_t1 = None
if not flair_t1_registered:
    print("Register FLAIR to T1")
    flair_to_t1 = ants.registration(
        fixed=t1_resampled,
        moving=flair,
        type_of_transform="Affine",
        aff_metric="mattes",
        syn_metric="mattes",
        write_composite_transform=True,
        outprefix=out("flair_to_t1_"),
        verbose=verbose,
    )
#Step 2: Register T1 → MNI
print("Register T1 to MNI")
t1_to_mni = ants.registration(
    fixed=mni,
    moving=t1,
    type_of_transform="Affine",
    aff_metric="meansquares",
    syn_metric="meansquares",
    write_composite_transform=True,
        outprefix=out("t1_to_mni_"),
        verbose=verbose,
    )
    #Step 3: Apply transforms to FLAIR → MNI
    transforms = [out("t1_to_mni_Composite.h5")]
    if not flair_t1_registered:
        transforms.append(out("flair_to_t1_Composite.h5"))
    flair_mni_registered = ants.apply_transforms(
        fixed=mni,
        moving=flair,
        transformlist=transforms,
        interpolator="linear",
    )
    #Step 4: Save results
    if output_path:
        os.makedirs(output_path, exist_ok=True)
        if flair_to_t1:
            ants.image_write(
                flair_to_t1["warpedmovout"], out("flair_t1_registered.nii.gz")
            )
        ants.image_write(t1_to_mni["warpedmovout"], out("t1_mni_registered.nii.gz"))
        ants.image_write(flair_mni_registered, out("flair_mni_registered.nii.gz"))
    return flair_mni_registered, t1_to_mni["warpedmovout"]
if __name__ == "__main__":
    original_data_dir = r"C:\Users\Anders\OneDrive - Aalborg Universitet\Dokumenter\Forskningsassistent - Neurologisk afdeling\Central post-stroke-pain\Brasilien_t1_flair"
    subject_id = "001"
    mni_registered_flair_org_t1 = "mni_registered_flair_org_t1"
```

```
mni_registered_flair_synth_t1 = "mni_registered_flair_synth_t1"
subject_path = os.path.join(original_data_dir, subject_id)
print(subject_path)
os.makedirs(mni_registered_flair_org_t1, exist_ok=True)
os.makedirs(mni_registered_flair_synth_t1, exist_ok=True)
print(subject_id[-2:])
subject_filename_synthRS = f"billede_{subject_id[-2:]}.nii"
synthRS_data_dir = "SynthRS"
subject_synthRS = os.path.join(synthRS_data_dir, subject_filename_synthRS)
print(subject_synthRS)
t1_img_path = os.path.join(original_data_dir, subject_id, "t1.nii")
flair_img_path = os.path.join(original_data_dir, subject_id, "flair.nii")
template_dir = "mni_icbm152_nl_VI_nifti"
template_img_path = os.path.join(
    template_dir, "icbm_avg_152_t1_tal_nlin_symmetric_VI.nii"
)
output_base_path_org = os.path.join(mni_registered_flair_org_t1, subject_id)
output_base_path_synth = os.path.join(mni_registered_flair_synth_t1, subject_id)
os.makedirs(output_base_path_org, exist_ok=True)
os.makedirs(output_base_path_synth, exist_ok=True)
#Load images
flair = ants.image_read(flair_img_path, reorient="RAS")
t1 = ants.image_read(t1_img_path, reorient="RAS")
t1_synthRS = ants.image_read(subject_synthRS, reorient="RAS")
mni152 = ants.image_read(template_img_path, reorient="RAS")
register_pipeline(flair, t1, mni152, output_base_path_org, False)
register_pipeline(flair, t1_synthRS, mni152, output_base_path_synth, True)
```

### Explanation of the codes in the script:

ants.resample_image_to_target is a function that takes an image object and change its spatial features to match a reference image.

interp_type=”bSpline” is the interpolations method that is used to maintain the image quality when the resolution

ants.registation() performs the image registration by spatially align two images so they match

fixed=t1_resampled is the reference image. The registration would try to align the moving image to this fixed image

moving=flair is the moving image, and get transformed to fit the fixed image

type_of_transform=“Affine” is the type of transformation. It is a linear transformation that are able to translation, rotation and scaling of the images.

aff_metric“mattes” is the similarity metric that is used to assess how good the images fit under the affine part of transformation. Here the Mattes Mutual Information is used

syn_metric=”mattes” is another metric for non-linear transformations

write_composite_transform=True saves a composite transform as a file with all the calculated transformations

outprefix=out(“flair_to_t1_”) is the prefix for the output files

transforms = [out()] is a list with the paths to the transformation files that should be used

if not flair_t1_registered is a conditional statement that verifies if the FLAIR-image is registered to the T1 image.

transform.append(out()) adds the transformation of FLAIR to T1 to the transform list, if FLAIR is not registered to T1

ants.apply_transforms() performs the image transformation.

Interpolator=”linear” is the interpolation method that is used when pixel values are calculated in the transformed image.

ants.image_write save the transformed images.

(17).

## Normalisation

The normalisation was made in Python and was developed by the author with use of libraries for normalisation with z-score. The libraries used in this script are NumPy, nibabel and sklearn.

The script for normalisation was:

```
import numpy as np
import nibabel as nib
from sklearn.preprocessing import StandardScaler
import os


def standardise_single_image(image_path):
    """
    Loads a NifTI image, standardises its pixel values
    (Z-score normalisation) and returns the standardised image.
    """
    try:
        img = nib.load(image_path)
        image_data = img.get_fdata()
        original_shape = image_data.shape
        #Flatten the Image to a 2D Array (num_pixel, 1) for StandardScaler
        flat_image = image_data.reshape(-1, 1)
        #Initialiser og fit scaler'en på det fladgjorte billede og transformer dataen.
        scaler = StandardScaler()
        standardised_flat_image = scaler.fit_transform(flat_image)
        #Reshape de standardiserede data tilbage til den originale billedform
        standardised_image = standardised_flat_image.reshape(original_shape)
        print(f"Billede '{os.path.basename(image_path)}' er standardiseret.")
        return standardised_image, img.affine, img.header, None # Returner ingen fejl
    except FileNotFoundError:
        error_msg = f"Fejl: Filen blev ikke fundet: {image_path}"
        print(error_msg)
        return None, None, None, error_msg
    except nib.exceptions.Nifti1Error as e:
        error_msg = f"Fejl: Ikke en gyldig NIfTI-fil: {image_path} - {e}"
        print(error_msg)
        return None, None, None, error_msg
    except Exception as e:
        error_msg = f"Fejl ved behandling af {image_path}: {e}"
        print(error_msg)
        return None, None, None, error_msg


image_to_standardize_path = '/home/maria/Data/Alle_billeder/Syntetiseret/t1_mni_registered_syntetiseret_95.nii'
print(f"Behandler enkeltbillede: '{os.path.basename(image_to_standardise_path)}'")
#Standardise the image
scaled_image_data, affine, header, error = standardise_single_image(image_to_standardise_path)


if scaled_image_data is not None:
    #Save the standardised image
    try:
        output_filename = f"standardised_single_{os.path.basename(image_to_standardise_path)}"
        new_img = nib.Nifti1Image(scaled_image_data, affine, header)
        nib.save(new_img, output_filename)
        print(f"Gemt: {output_filename}")
    except Exception as e:
        print(f"Fejl ved gemning af standardiseret billede: {e}")
else:
    print(f"Standardisering mislykkedes for {os.path.basename(image_to_standardize_path)}.")
```

### Explanation of the codes in the script:

def standardize_single_image(image_path) is a definition of a function, for standardising the input image with z-score normalisation.

nib.load(image_path) loads the input image from a path

img.get_fdata() extracts the raw pixel intensity from the loaded image and returns it as a NumPy array image_data.shape saves the original dimensions of the NumPy array

image_data.reshape(-1, 1) changes the dimensions of the array. -1 is a dimension which automatically gets calculated from the total pixel in the image.

StandardScaler() standardise the data, so it has a mean at 0 and a standard deviation at 1

scaler.fit_transform performs the standardising by to steps. .fit calculates the mean and standard deviation of all pixelvalues in the image. .transform uses the z-score normalisation formula on every pixel

(20–25).

### MSE measurement

The MSE-measurement was performed in Python. The script for the measurement was developed by the author with use of libraries for calculating MSE. The libraries used in this script are SimpleITK, NumPy and Pandas.

The script for MSE-measurement was:

```
import SimpleITK as sitk

import numpy as np

import os

import pandas as pd


#Define the function: Calculates the MSE between two images

def calculate_mse(image1_path, image2_path): #The function for calculation of MSE between two images

    image1 = sitk.ReadImage(image1_path, sitk.sitkFloat32) #The first argument for the function

    image2 = sitk.ReadImage(image2_path, sitk.sitkFloat32) #The second argument for the function

    image1_array = sitk.GetArrayFromImage(image1).astype(np.float32)

    image2_array = sitk.GetArrayFromImage(image2).astype(np.float32)

    mse = np.mean((image1_array - image2_array) ** 2)

    return mse
```

### Explanation of the codes in the script:

def calculate_mse is the function. This function implements a method for calculation of MSE between two images and defines the code for achieving results.

sitk.ReadImage loads images.

sitk.sitkFloat32 specifies that SimpleITK is intended to read the image and store its pixel values as 32-bit floating-point numbers.

sitk.GetArrayFromImage converts an image object to a NumPy-array

astype() converts the data type for all elements in the NumPy-array

np.float32 is the data type notation. Every pixel value in the array saves as a 32-bit floating-point number

np.mean is a function from NumPy-library which calculates the mean of all elements in the given array. It subtracts elementwise between two images and squares each element in the array.

(27–29).

### SSIM measurement

The SSIM-measurement was performed in Python. The script for the measurement was developed by the author with use of libraries for calculating SSIM. The libraries used in this script are skimage, NiBabel, NumPy and Pandas.

The script for SSIM-measurement was:

```
from skimage.metrics import structural_similarity as ssim
import nibabel as nib
import numpy as np
import os
import pandas as pd

#Define variables that save the two paths for the images
reference_path = '/home/maria/Data/Reference/mni_registered_flair_org_t1/'
synthesized_path = '/home/maria/Data/Syntetiseret/mni_registered_flair_synth_t1/'
num_pairs = 87
#Define string variables for the prefix and suffix to the filenames
filename_prefix_reference = "t1_mni_registered_"
filename_prefix_synthesized = "flair_mni_registered_"
filename_suffix = ".nii"
#Create empty lists to store the results
pair_numbers = []
```

```
ssim_scores = []
#Start a for-loop
for i in range (1, num_pairs + 1):
    reference_filename = os.path.join(reference_path,
f"{filename_prefix_reference}{i}{filename_suffix}")
    synthesized_filename = os.path.join(synthesized_path,
f"{filename_prefix_synthesized}{i}{filename_suffix}")
    try:
    #Load the two images
        nii_imgA = nib.load(reference_filename)
        nii_imgB = nib.load(synthesized_filename)
    #Retrieve the image data as NumPy arrays
        imageA = nii_imgA.get_fdata()
        imageB = nii_imgB.get_fdata()
    #Calculate SSIM
        (score, diff) = ssim(imageA, imageB, data_range=1.0, full=True)
        diff = (diff * 255).astype("uint8")
        print(f"SSIM: {i}: {score:.4f}")
        #Add the results to the lists
        pair_numbers.append(i)
        ssim_scores.append(score)
    except FileNotFoundError:
        print("Fejl: En eller begge billedfiler blev ikke fundet. Tjek stierne.")
    except Exception as e:
        print(f"Der opstod en fejl: {e}")
#Create a panda dataframe from the lists
data = {'Image_pair': pair_numbers, 'SSIM Score': ssim_scores}
df = pd.DataFrame(data)
```

```
#Save the dataframe to an Excel-file

excel_filename = "ssim_resultater.xlsx"

df.to_excel(excel_filename, index=True)

print(f"Færdig med at beregne SSIM for de fundne billedepar.")
```

**Explanation of the codes in the script:**

num_pairs is a variable that saves an integer

range(1, num_pairs +1) is a function that generates a sequence of numbers

for i in is a for-loop that runs one time for every number. i is used to construct the unique filenames for every image pair.

os.path.join(…) is a function from os, which combines path components into a valid file path

f"{...}{i}{...} is an f-string, which builds strings in Python, and inserts the value of i between the prefix and suffix.

nib.load loads the NIfTI-image file

.get_fdata() returns the actual pixel/voxel data from the images as a NumPy-array

ssim(imageA, imageB, …) use the SSIM-function that was imported from skimage.metrics library.

data_range=1.0 tells the SSIM-function which data range the image pixels have.

full=True returns the SSIM-value (score) and a "difference"-image (diff).

diff*255 scales the values up to match a typical 8-bit image format.

.astype("uint8") converts the datatype for diff-array to uint8 which is a standard shape for grayscale images.

f"SSIM: {i}: {score:.4f}" is a f-string that gives the statistical text SSIM, the current pair number (i) and the last part of the string inserts the calculated SSIM-score with four decimals

append(i) adds the current value of i to the list

data = {'Image Pair': pair_numbers, 'SSIM Score': ssim_scores} is a Python-dictionary that assigns Image Pair to the list pair_numbers in the first column and SSIM Score to the list ssim_scores in the second column

df = pd.DataFrame(data) creates a DataFrame, which is a table from the dictionary. The variable df contains this table

df.to_excel(excel_filename, index=True) saves the DataFrame as an Excel-file. Index=True includes the DataFrame index as a separate column in the Excel-file.

(20,21,23,29,30).

## Worksheet 5: Z-score Normalisation in Image Analysis: Selection and Rationale

The normalization in the project was made by using Z-score normalization as mentioned in the article. The formula for Z-score normalization is:

$$Z = \frac{x - \mu}{\sigma}$$

Where x is the individual data point, $\mu$ is the mean of pixel intensity in the image and $\sigma$ the standard deviation.

After normalization, the images would have a mean around 0 and a standard deviation around 1, which makes it possible to compare the images intensities.

The Z-score normalization was chosen for this project, because it considers outliers and maintain the relative distribution of intensities compares to other normalization methods (24). Because the data in this project derives from different scanners the intensities can differ. Furthermore, the data was collected at different time points before contrast was giving, why the intensities could be different too. This, Z-score normalisation will account for.

Another method for normalization is to use minimum and maximum and scale the data into an interval. Minimum and maximum was not chosen in this project because it does not handle outliers as good as Z-score normalization. If there are outliers or big differences in the original intensity fields, these would possibly distort the information in the images, because the min-max normalization compromises the data in a small interval (11). The Z-score normalization maintains the statistical distribution of the intensities, which is essential when comparing images with MSE and SSIM. Furthermore, Z-score normalization is wide used in work with deep learning and image analysis (11,13).

## Worksheet 6: Visualising Image Differences: Insights from SSIM Difference Maps

When calculating SSIM, a difference image was also created for each comparison. These difference images provide a visual representation of the difference between two images, by subtracting the pixel values of one image from those of the other, pixel by pixel. The difference image displays areas where the images are similar as dark, and areas where the images have big differences as light. The darkest areas in the difference image indicates the smallest structural differences.

Figure B shows the difference image from image pair with the lowest SSIM value, and therefore most structural differences. There are some dark spots, which indicate small differences, but there are also some white or light spots, which indicate large differences. It appears noisy, and the structure in the brain is unrecognisable. The image indicates that there is a difference between the two images in both structure and intensity and not just in small details. The difference image corresponds with the SSIM value of 0,17 for this image pair.

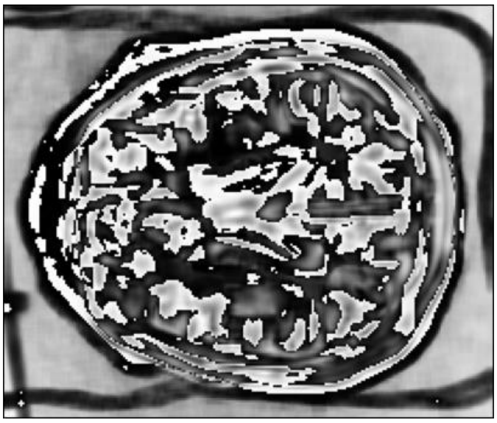


*Figure B: Difference image of the image pair with lowest SSIM value*

Figure C, conversely, shows the difference image for the image pair with the highest SSIM value and thus the images with the best structural similarity for this study. Although there are still light areas, the structure of the brain is clearer in this image compared to Figure B. This indicates that there is better structural similarity between this image pair, but because there are a lot of light areas, there are still differences. This corresponds with the SSIM value of 0.48, which indicates that there are structural differences.

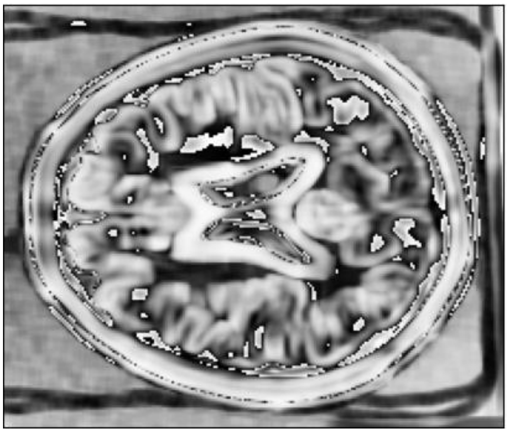


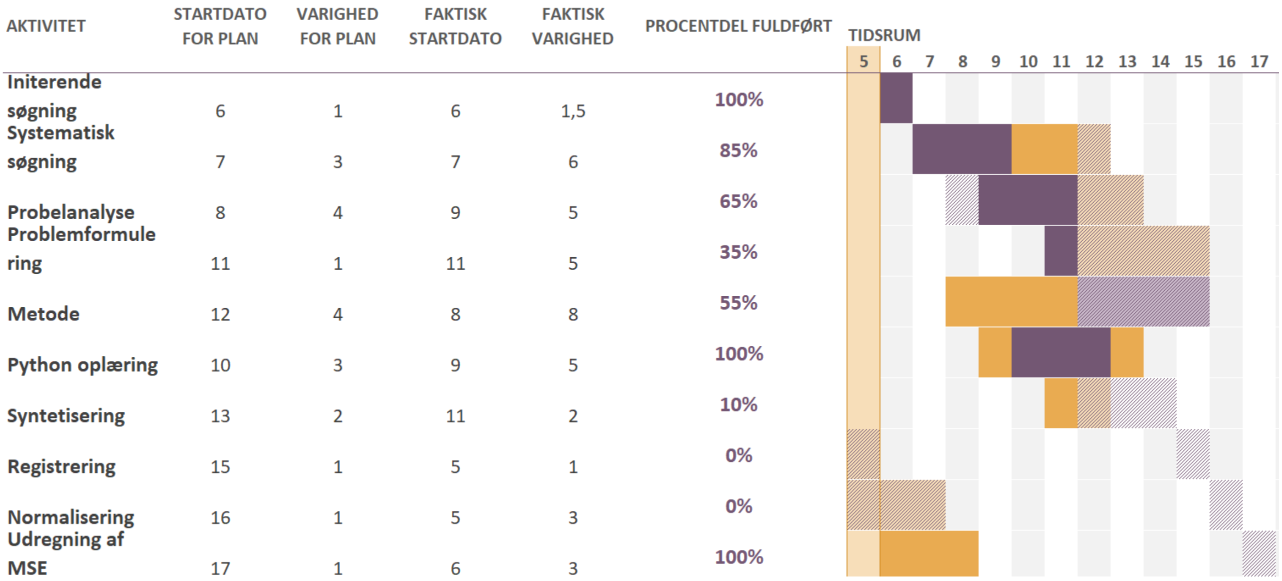
*Figure C: Difference image of the image pair with highest SSIM value*

Although the difference image in Figure C is visually better than the difference image in Figure B, the structure in the brain is still noisy, and the edges in the tissue is blurred. This emphasises the conclusion in the article, that there is a need for further analysis or optimising in the process, to conclude if the synthesised images are usable.

## Worksheet 7: Effective Time Management and Structure in master's Thesis Writing

For organising the master's thesis, a Gantt chart was made for outlining the estimated duration for each task of the study (Table C). Furthermore, it gave an overview of the different progress that should be done. It was useful for visual representation of where I was in the process and what I still needed to do. It was updated approximately once a week to check up on status.

Besides the Gantt chart, a weekly and daily plan was made in a physical calendar. Here a plan for the week was made, both with work relevant task, but also with spare time errands. Furthermore, a to-do list was made to make sure the time was well spent.



| AKTIVITET | STARTDATO FOR PLAN | VARIGHED FOR PLAN | FAKTISK STARTDATO | FAKTISK VARIGHED | PROCENTDEL FULDFØRT |
|---|---|---|---|---|---|
| Initerende søgning | 6 | 1 | 6 | 1,5 | 100% |
| Systematisk søgning | 7 | 3 | 7 | 6 | 85% |
| Probelanalyse | 8 | 4 | 9 | 5 | 65% |
| Problemformulering | 11 | 1 | 11 | 5 | 35% |
| Metode | 12 | 4 | 8 | 8 | 55% |
| Python oplæring | 10 | 3 | 9 | 5 | 100% |
| Syntetisering | 13 | 2 | 11 | 2 | 10% |
| Registrering | 15 | 1 | 5 | 1 | 0% |
| Normalisering | 16 | 1 | 5 | 3 | 0% |
| Udregning af MSE | 17 | 1 | 6 | 3 | 100% |

*Table C: Shows the first 10 activities of the Gantt diagram in the middle of the progress*

During the master's thesis, multiple supervisory sessions were held with both supervisors and the author. These were planned by the author, by booking the supervisors and both verbal and written supervision was giving through the master's thesis.

With the Gantt chart and weekly planner, the structure of the master's thesis was made. Furthermore, it helped ensure that made. Furthermore, it helped ensure that the learning objectives were met within the timeframe.

## References

1. Dansk råd for genoplivning, Sundhedsstyrelsen. Faktaark: Stroke i Danmark. 2024 Apr; Available from: http://www.nejm.org/doi/10.1056/NEJMoa1804492

2. THE NATIONAL INSTITUTE OF NEUROLOGICAL DISORDERS AND STROKE rt-PA STROKE STUDY GROUP. Tissue plasminogen activator for acute ischemic stroke. N Engl J Med [Internet]. 1995 Dec 14 [cited 2025 Mar 18];333(24):1581–8. Available from: https://pubmed.ncbi.nlm.nih.gov/7477192/

3. Emberson J, Lees KR, Lyden P, Blackwell L, Albers G, Bluhmki E, et al. Effect of treatment delay, age, and stroke severity on the effects of intravenous thrombolysis with alteplase for acute ischaemic stroke: a meta-analysis of individual patient data from randomised trials. Lancet [Internet]. 2014 Nov 29 [cited 2025 Mar 18];384(9958):1929–35. Available from: https://pubmed.ncbi.nlm.nih.gov/25106063/

4. Goyal M, Menon BK, Van Zwam WH, Dippel DWJ, Mitchell PJ, Demchuk AM, et al. Endovascular thrombectomy after large-vessel ischaemic stroke: a meta-analysis of individual patient data from five randomised trials. Lancet [Internet]. 2016 Apr 23 [cited 2025 Mar 18];387(10029):1723–31. Available from: https://pubmed.ncbi.nlm.nih.gov/26898852/

5. Jauch EC, Saver JL, Adams HP, Bruno A, Connors JJ, Demaerschalk BM, et al. Guidelines for the Early Management of Patients With Acute Ischemic Stroke. Stroke [Internet]. 2013;(44):870–947. Available from: http://stroke.ahajournals.org/lookup/suppl/

6. Vert C, Parra-Fariñas C, Rovira À. MR imaging in hyperacute ischemic stroke. Vol. 96, European Journal of Radiology. Elsevier Ireland Ltd; 2017. p. 125–32.

7. Larsen E. MR - 500A Cerebrum Stroke. 2023 Mar.

8. Yavarian Y. MR – Cerebrum – Stroke FAST Track [Internet]. 2024 Jun. Available from: https://pri.rn.dk/Sider/32693.aspx

9. Westbrook Catherine, Talbot John. MRI in practice. 5th ed. England: Wileyblackwell; 2018.

10. Fiebach JB, Schellinger PD, Jansen O, Meyer M, Wilde P, Bender J, et al. CT and diffusion-weighted MR imaging in randomized order: Diffusion-weighted imaging results in higher accuracy and lower interrater variability in the diagnosis of hyperacute ischemic stroke. Stroke. 2002 Sep;33(9):2206–10.

11. Embase | The comprehensive medical research database | Elsevier [Internet]. [cited 2025 Jun 1]. Available from: https://www.elsevier.com/products/embase

12. About IEEE Xplore [Internet]. [cited 2025 Jun 1]. Available from: https://ieeexplore.ieee.org/Xplorehelp/overview-of-ieee-xplore/about-ieee-xplore

13. Iglesias JE, Billot B, Balbastre Y, Magdamo C, Arnold SE, Das S, et al. SynthSR: A public AI tool to turn heterogeneous clinical brain scans into high-resolution T1-weighted images for 3D morphometry. Sci Adv [Internet]. 2023 Feb 1;9(5). Available from: https://www.science.org

14. Sharma A, Hamarneh G. Missing MRI Pulse Sequence Synthesis using Multi-Modal Generative Adversarial Network. IEEE Trans Med Imaging [Internet]. 2020 Apr 27;39(4):1170–83. Available from: http://arxiv.org/abs/1904.12200

15. Iglesias JE, Billot B, Balbastre Y, Tabari A, Conklin J, Gilberto González R, et al. Joint super-resolution and synthesis of 1 mm isotropic MP-RAGE volumes from clinical MRI exams with scans of different orientation, resolution and contrast. Neuroimage. 2021 Aug 15;237.

16. Osman AFI, Tamam NM. Deep learning-based convolutional neural network for intramodality brain MRI synthesis. J Appl Clin Med Phys. 2022 Apr 1;23(4).

17. Conte GM, Weston AD, Vogelsang DC, Philbrick KA, Cai JC, Barbera M, et al. Generative adversarial networks to synthesize missing T1 and FLAIR MRI sequences for use in a multisequence brain tumor segmentation model. Radiology. 2021;299(2):313–23.

18. AYENI JA. Convolutional Neural Network (CNN): The architecture and applications. Applied Journal of Physical Science [Internet]. 2022 Dec 30;4(4):42–50. Available from: https://integrityresjournals.org/journal/AJPS/article-full-text-pdf/1378CAD82

19. Bezdan T, Bačanin Džakula N. Convolutional Neural Network Layers and Architectures. In Singidunum University; 2019. p. 445–51.

20. Iglesias JE. SynthSR. 2023 [cited 2025 Jun 1]. SynthSR - Free Surfer Wiki. Available from: https://surfer.nmr.mgh.harvard.edu/fswiki/SynthSR

21. Registration — ANTsPy dev (latest) documentation [Internet]. [cited 2025 Jun 1]. Available from: https://antspy.readthedocs.io/en/latest/registration.html

22. NiBabel developers. Streamlines [Internet]. 2006 [cited 2025 Jun 1]. Available from: https://nipy.org/nibabel/reference/nibabel.streamlines.html

23. NiBabel developers. dataobj_images [Internet]. 2006 [cited 2025 Jun 1]. Available from: https://nipy.org/nibabel/reference/nibabel.dataobj_images.html#nibabel.dataobj_images.DataobjImage.get_fdata

24. NiBabel developers. nifti1 [Internet]. 2006 [cited 2025 Jun 1]. Available from: https://nipy.org/nibabel/reference/nibabel.nifti1.html#nibabel.nifti1.Nifti1Image

25. NumPy Developers. NumPy reference — NumPy v2.2 Manual [Internet]. 2024 [cited 2025 Jun 1]. Available from: https://numpy.org/doc/stable/reference/index.html#reference

26. scikit-learn developers. StandardScaler — scikit-learn 1.6.1 documentation [Internet]. 2025 [cited 2025 Jun 1]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

27. Kowalik-Urbaniak IA, Castelli J, Hemmati N, Koff D, Smolarski-Koff N, Vrscay ER, et al. Modelling of subjective radiological assessments with objective image quality measures of brain and body CT images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) [Internet]. 2015 [cited 2025 Mar 19];9164:3–13. Available from: https://link.springer.com/chapter/10.1007/978-3-319-20801-5_1

28. Yaniv Z, Lowekamp BC, Johnson HJ, Beare R. SimpleITK Image-Analysis Notebooks: a Collaborative Environment for Education and Reproducible Research. J Digit Imaging. 2018 Jun 1;31(3):290–303.

29. NumPy community. NumPy Reference Release 2.2.0 Written by the NumPy community. 2025.

30. McKinney W. User Guide — pandas 2.2.3 documentation [Internet]. 2024 [cited 2025 Jun 1]. Available from: https://pandas.pydata.org/docs/user_guide/index.html#user-guide

31. The scikit-image team. skimage.metrics — skimage 0.24.0 documentation [Internet]. 2024 [cited 2025 Jun 1]. Available from: https://scikit-image.org/docs/0.24.x/api/skimage.metrics.html#skimage.metrics.structural_similarity