
Maschinelles Lernen in der Medizinphysik

Alexander Ratke

26. März 2021

Bachelor-Programmierkurs

Tagesablauf

Grundlagen zur Bildgebung und -bearbeitung

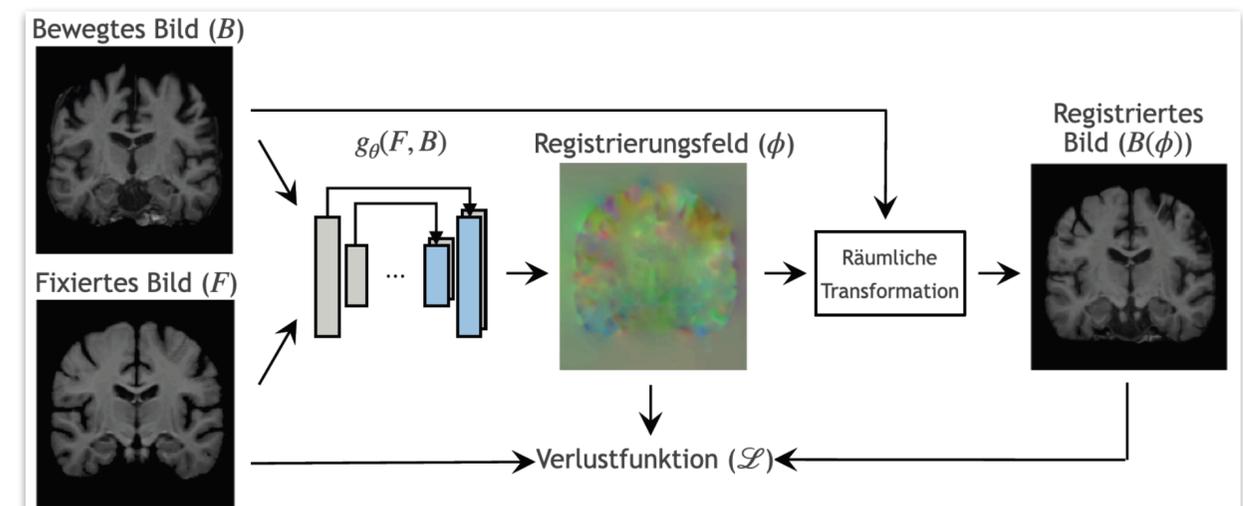
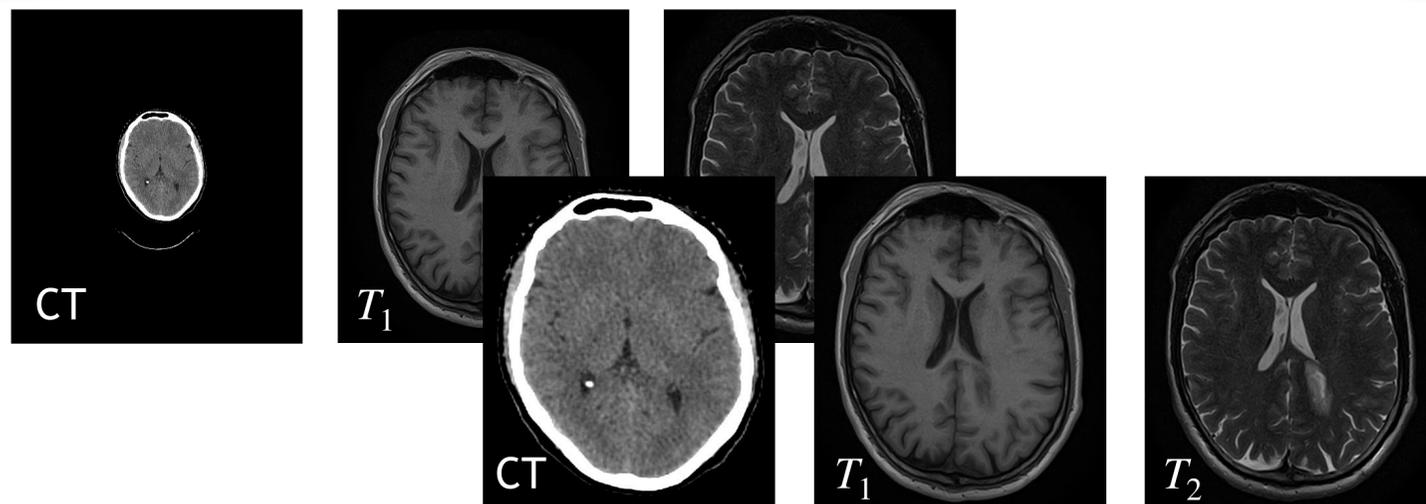
08:30 - 11:00

- Check von Hardware und Software
- Einführung in die Bildgebung
- Arbeiten mit Aufnahmen im DICOM-Format
 - ➔ 1. Aufgabe
- Einführung in die Bildbearbeitung
 - ➔ 2. Aufgabe

Grundlagen zur Bildregistrierung mit ML

13:00 - 16:00

- Einrichtung und Einweisung in Git
- Installation notwendiger Python-Pakete
- Einführung in die Bildregistrierung
 - ➔ 3. Aufgabe
 - ➔ 4. Aufgabe

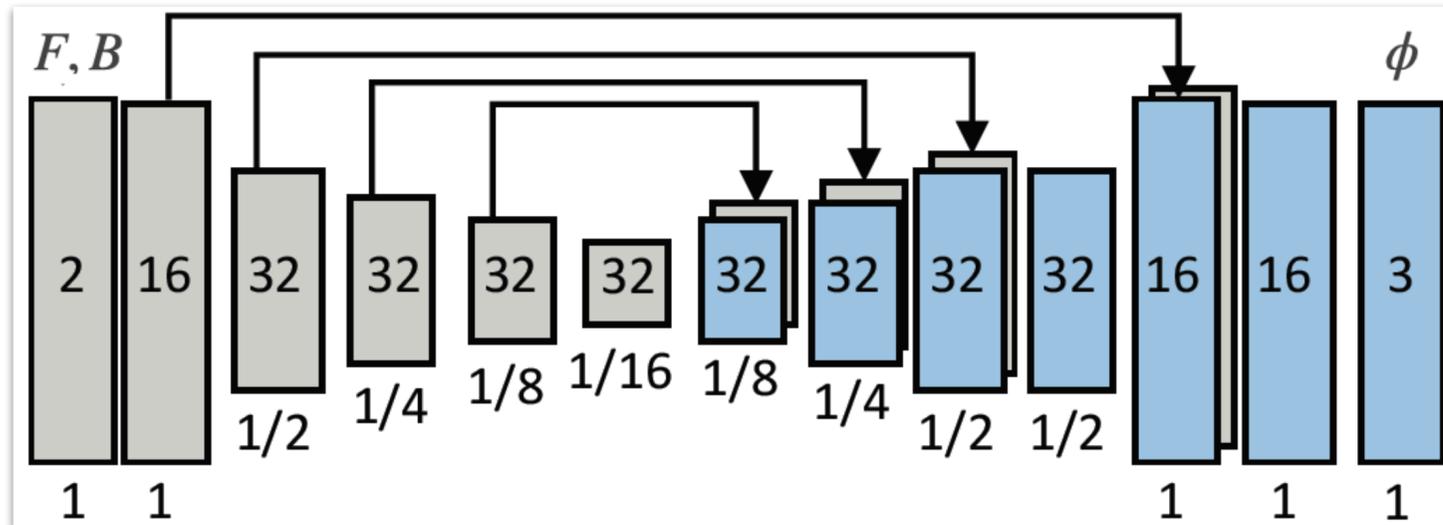


Teil 2

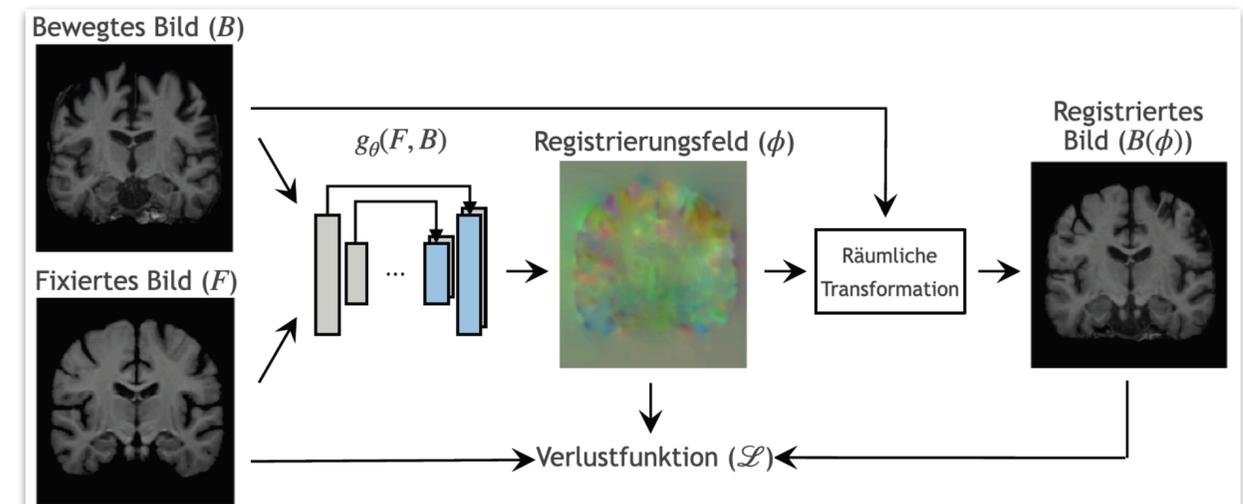
Grundlagen zur Bildregistrierung mit ML

13:00 - 16:00

- Einrichtung und Einweisung in Git
- Installation notwendiger Python-Pakete
- Einführung in die Bildregistrierung
 - ➔ 3. Aufgabe
 - ➔ 4. Aufgabe



$$NCC(R, I) = \frac{\left(\sum_p (I(p) - \bar{I}(p)) (R(p) - \bar{R}(p)) \right)^2}{\sum_p (I(p) - \bar{I}(p))^2 \sum_p (R(p) - \bar{R}(p))^2}$$



Git

- E5 Git-Repository für die Fusionierte Bildgebung ist [hier](#) zu finden

- Skripte zur Vorbereitung, Registrierung und Fusion
- Einrichtung des Repositories im home-Ordner mit *Clone with SSH* via

```
git clone ssh://git@git.e5.physik.tu-dortmund.de:10022/aratke/medical-image-fusion.git
```

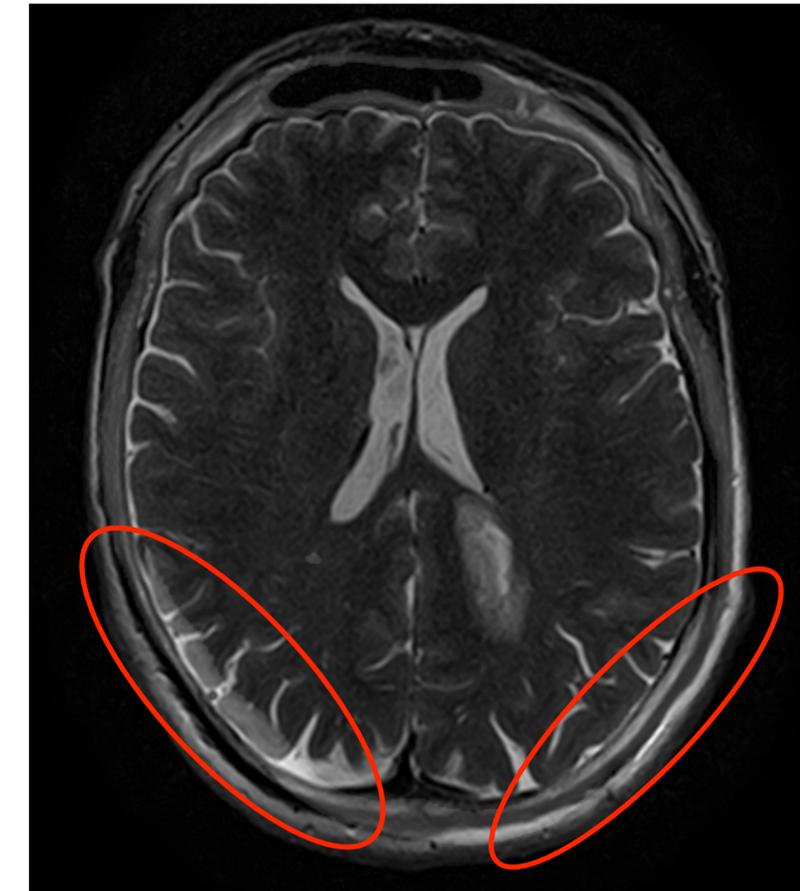
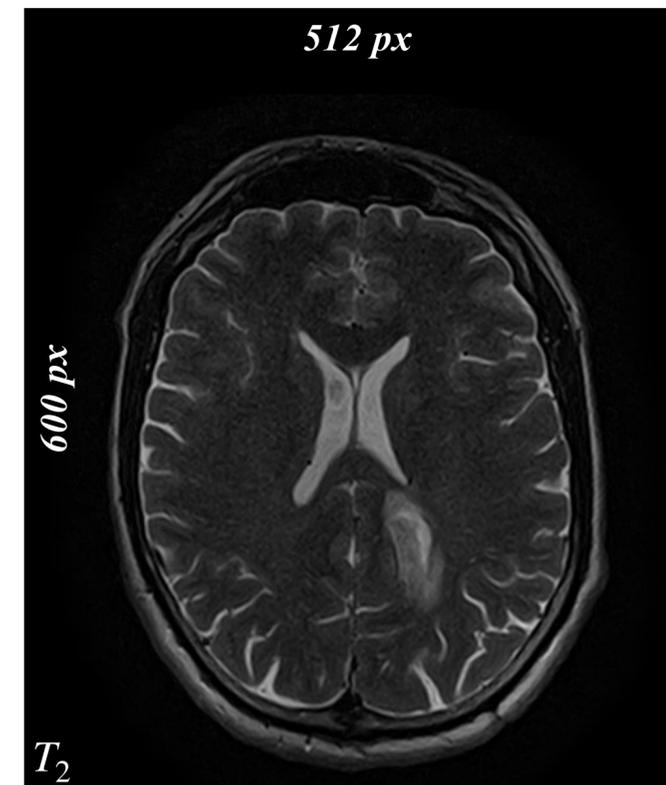
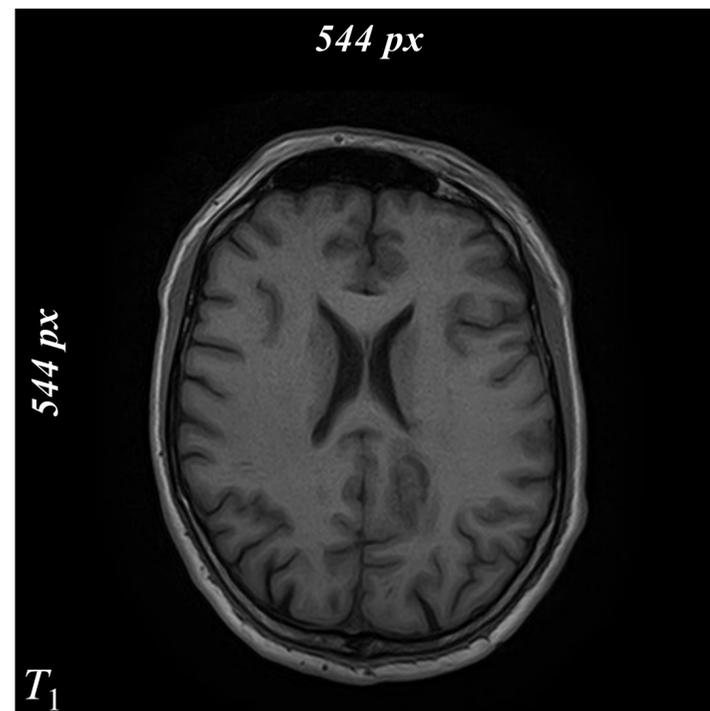
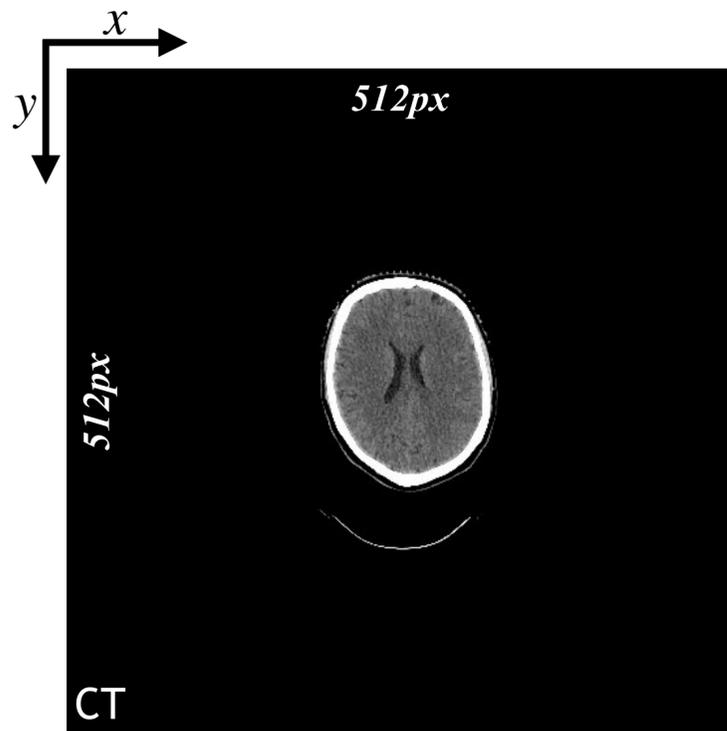
- Notwendige python-Pakete installieren durch *medical-image-fusion/registration/setup.py*

- Ausführen mit: `pip install --user voxelmorph`
 - *Package for DICOM medical file reading and writing*: `pydicom`
 - *Machine Learning Framework*: `tensorflow==2.2`
 - *Neural Network API*: `keras==2.3.1`
 - *Neural Networks Toolbox for Medical Imaging*: `neurite`

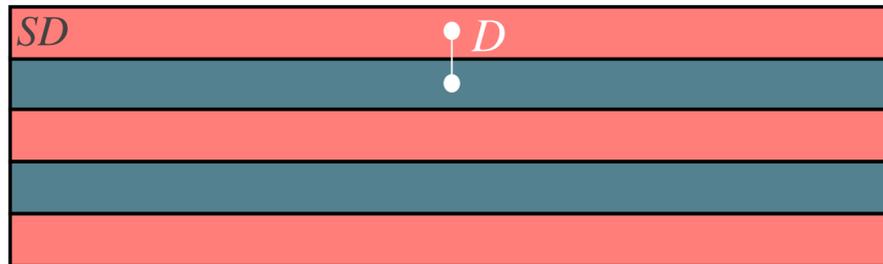
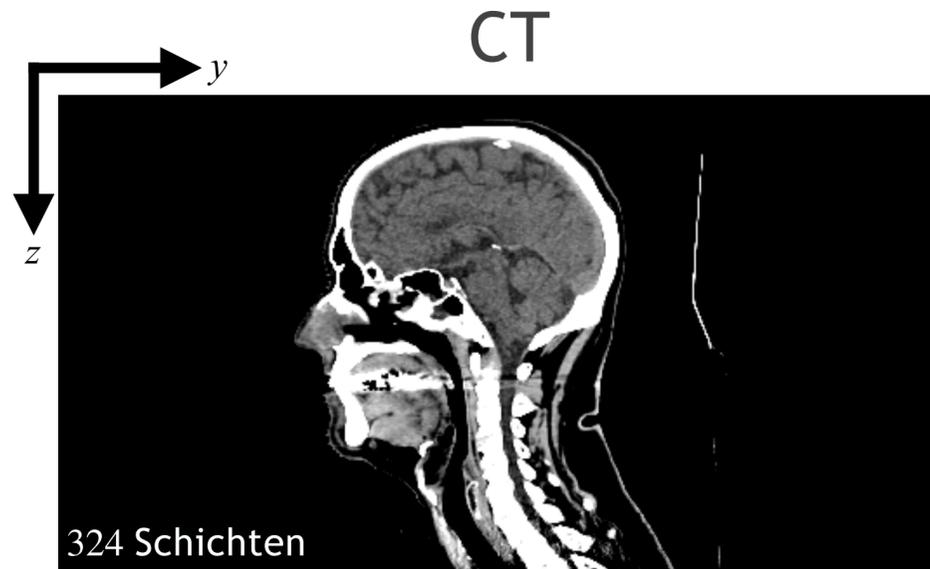
- Überprüfung, ob alle Skripte und Pakete vorliegen!

Vorbereitung der Aufnahmen

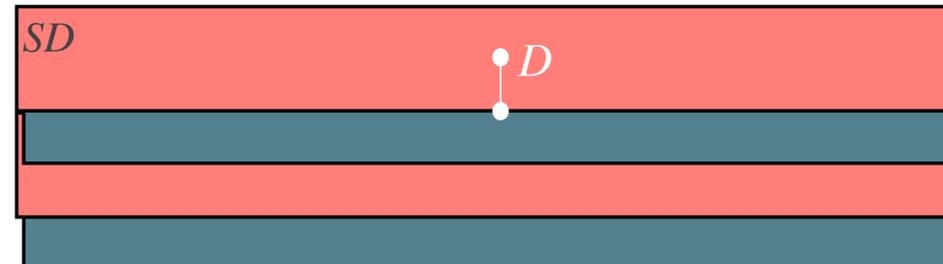
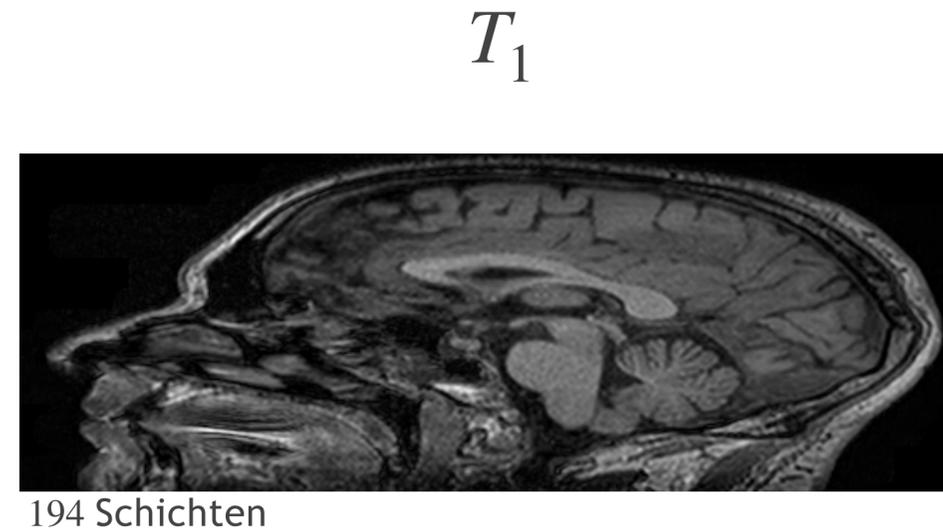
- Problematik zwischen CT und MRT:
 - Schichten passen nicht perfekt zueinander
 - Lageunsicherheiten
 - Unterschiedliche Bildeigenschaften (Schichtanzahl, -dicke und -abstand)
- Ziele:
 - Gleiche Schichtanzahl zwischen CT, T_1 und T_2
 - Gleiche Strukturen in den Schichten durch Rotation



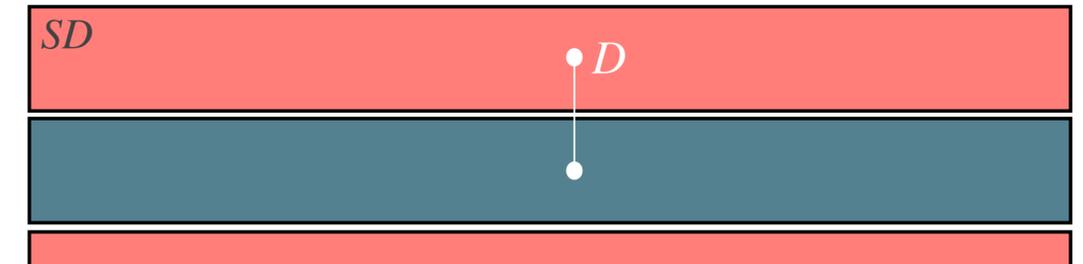
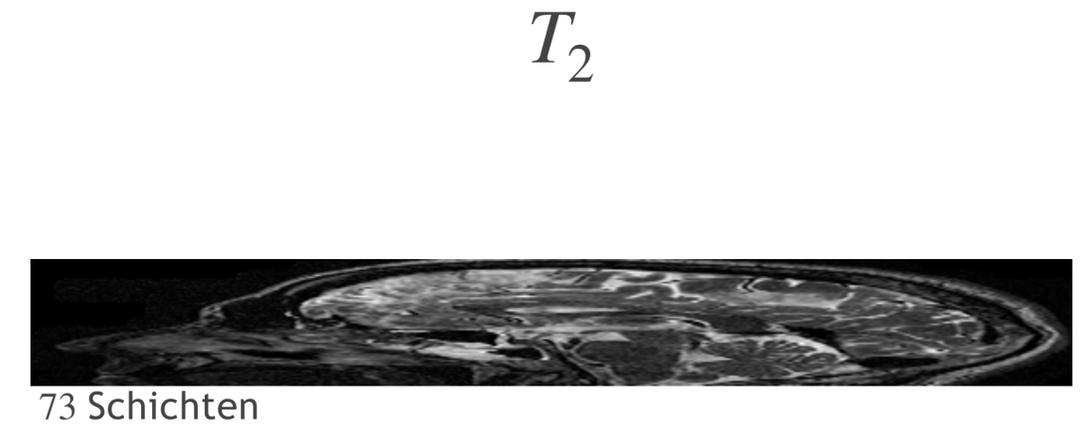
Vorbereitung der Aufnahmen



Schichtdicke: $SD = 1\text{mm}$
Schichtabstand: $D = 1\text{mm}$



Schichtdicke: $SD = 2\text{mm}$
Schichtabstand: $D = 1\text{mm}$

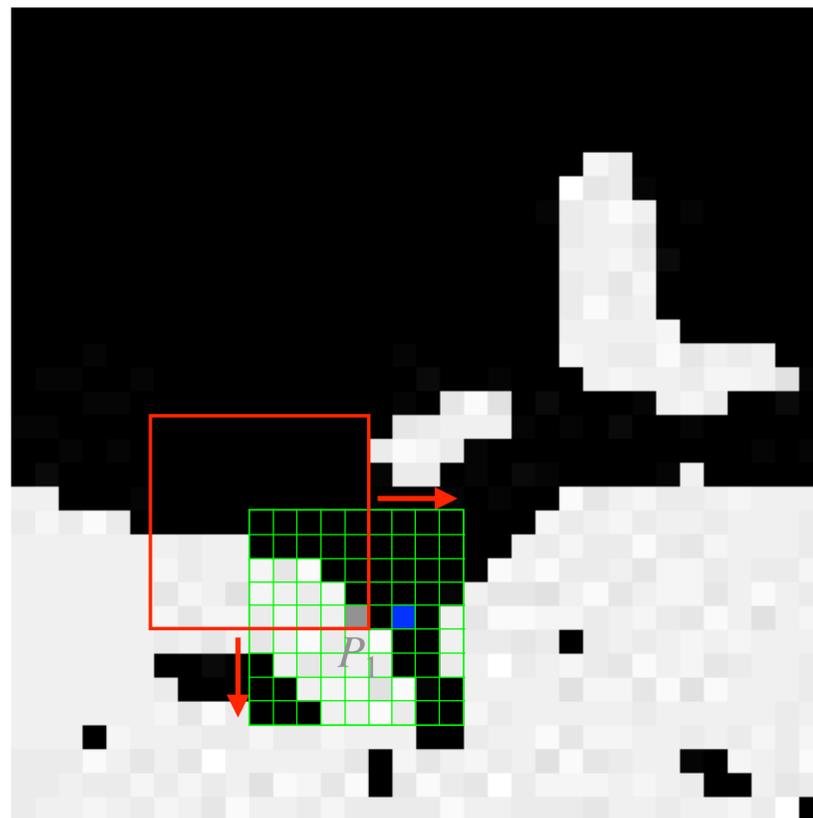
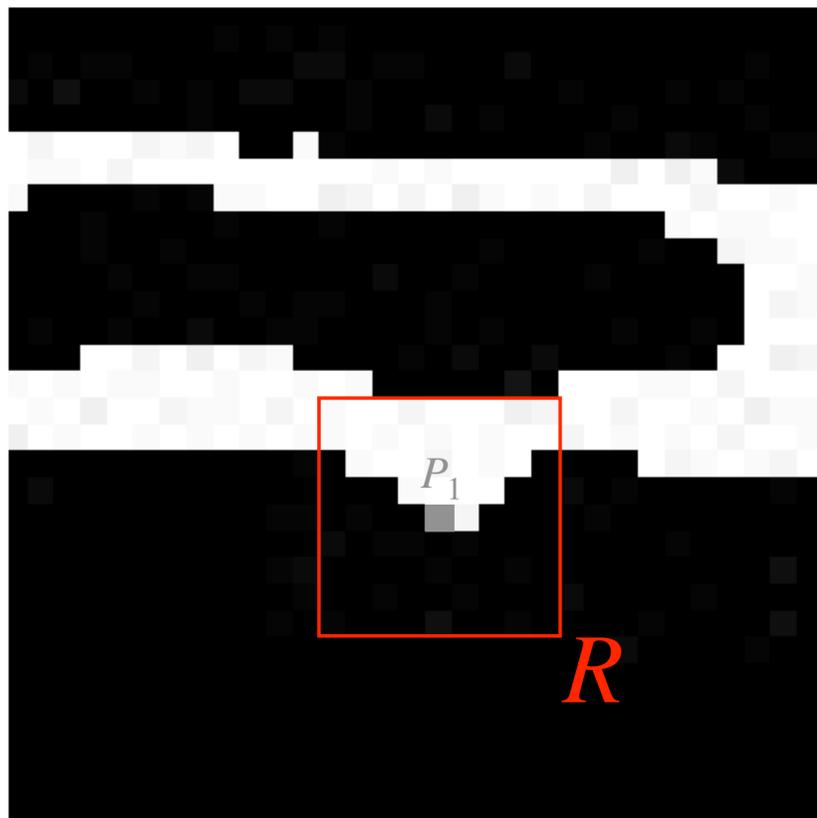


Schichtdicke: $SD = 2\text{mm}$
Schichtabstand: $D = 2,2\text{mm}$

Normierte Kreuzkorrelation

- Vergleich von Strukturen in Bildern unabhängig von Intensität
- Abtasten eines Bildes I mit einem Referenzbild R
- Wertebereich von NCC im Intervall $[0,1]$
 - 0 → Maximale Abweichung
 - 1 → Maximale Übereinstimmung

$$NCC(R, I) = \frac{\left(\sum_p (I(p) - \bar{I}(p)) (R(p) - \bar{R}(p)) \right)^2}{\sum_p (I(p) - \bar{I}(p))^2 \sum_p (R(p) - \bar{R}(p))^2}$$

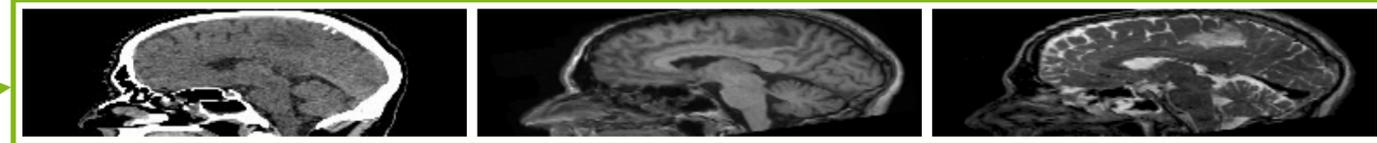
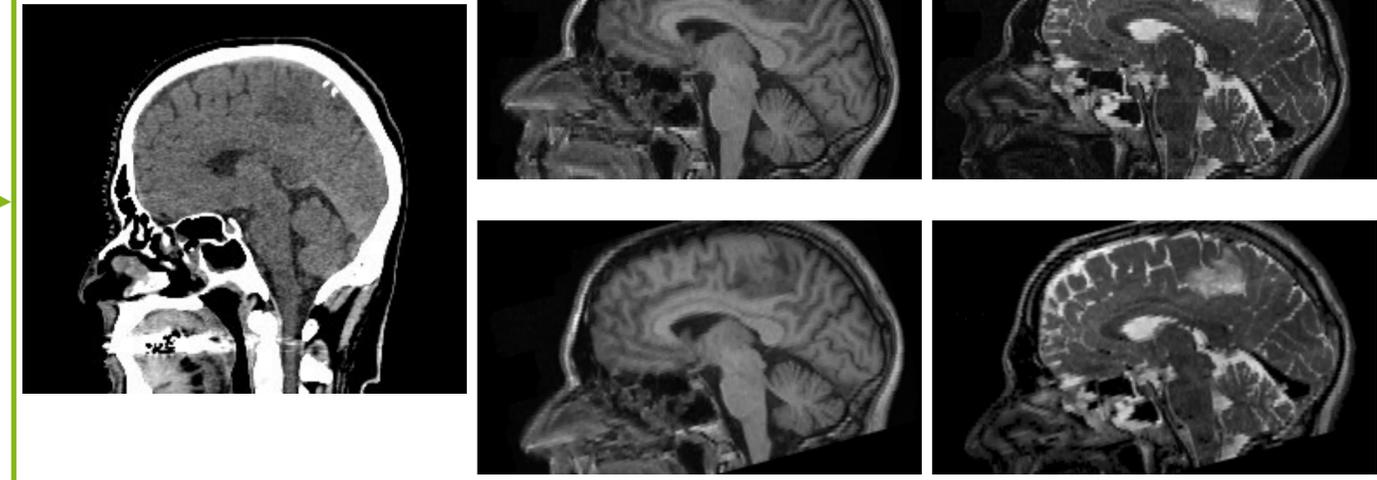


0,29	0,23	0,18	0,13	0,04	0,03	0,03	0,04	0,07
0,46	0,37	0,29	0,21	0,12	0,11	0,12	0,15	0,18
0,56	0,54	0,46	0,36	0,29	0,27	0,27	0,30	0,36
0,54	0,54	0,51	0,51	0,46	0,44	0,44	0,48	0,37
0,34	0,36	0,40	0,45	0,56	0,68	0,68	0,58	0,36
0,14	0,17	0,20	0,29	0,46	0,72	0,86	0,63	0,34
0,09	0,09	0,11	0,15	0,30	0,55	0,64	0,58	0,39
0,03	0,07	0,07	0,12	0,24	0,30	0,36	0,37	0,34
0,00	0,03	0,07	0,11	0,15	0,15	0,15	0,15	0,17

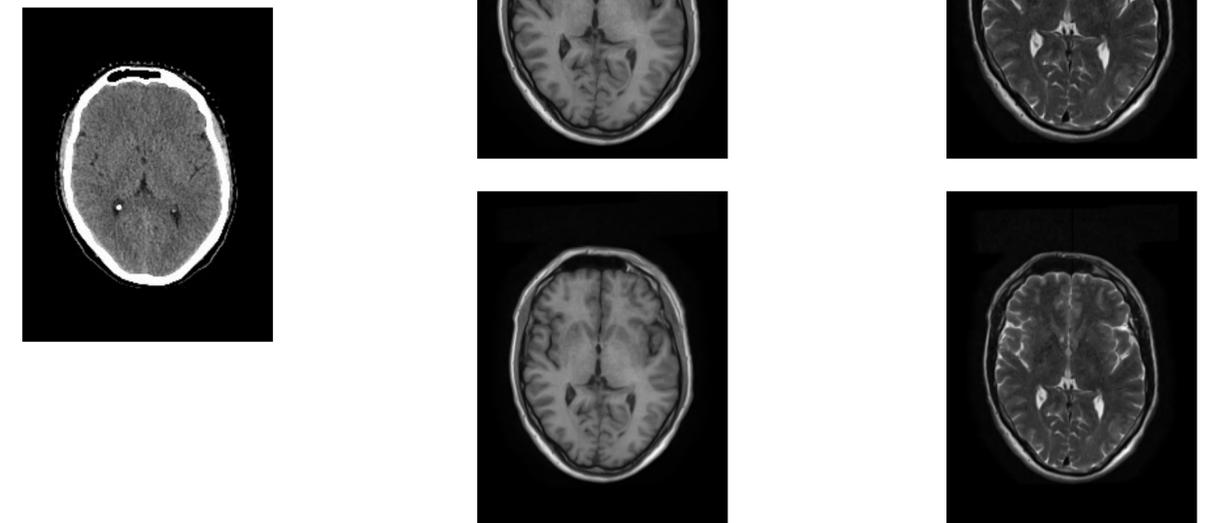
Vorbereitung der Aufnahmen

- CT-Aufnahme fixiert, T_1 und T_2 deformiert
- Rotation der MRT-Aufnahmen:
 - Rotation der MRT-Bilder in 1° -Schritten
 - Berechnung des NCC-Wertes für jede Rotation
 - Anwendung des Winkels mit höchstem NCC-Wert
- Auswahl gleicher Anzahl an Schichten:
 - Aufnahme mit kleinster Anzahl festgehalten (T_2)
 - Anfangsschichten durch NCC-Verfahren gewählt
 - Korrespondierende Schichten anderer Aufnahmen (CT und T_1) mittels Schichtabstand und -position ermittelt
- Anwendung der 2D Transformation auf alle Schichten

Querschnitt bei $x/2$

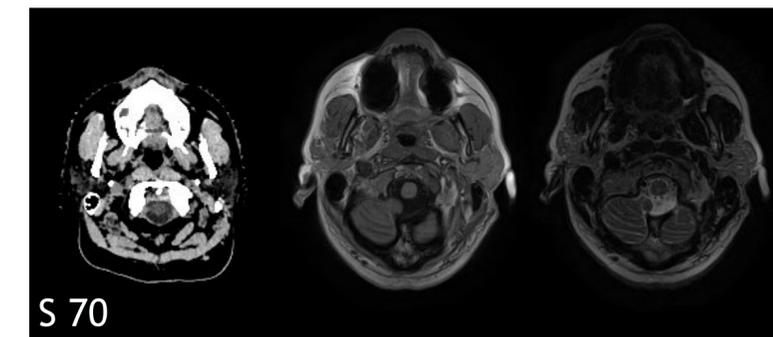
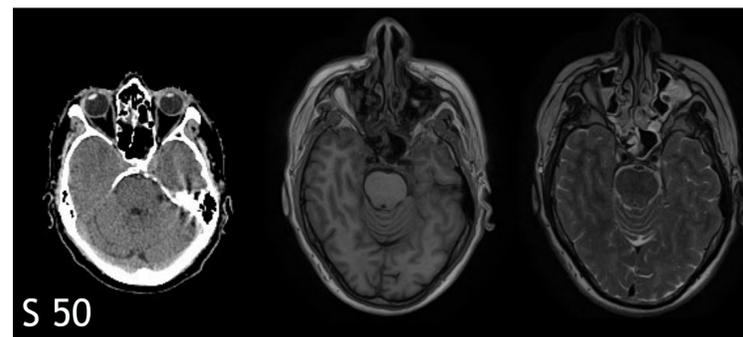
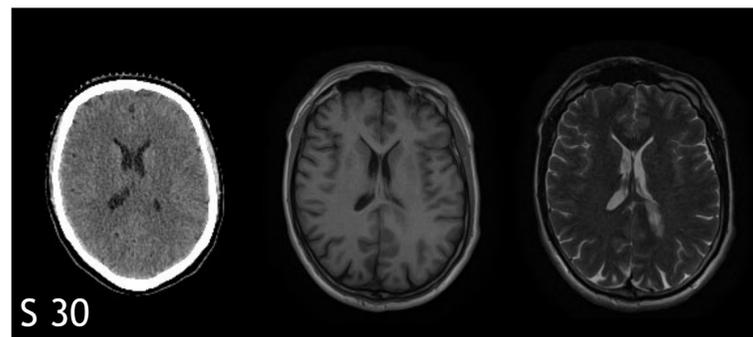
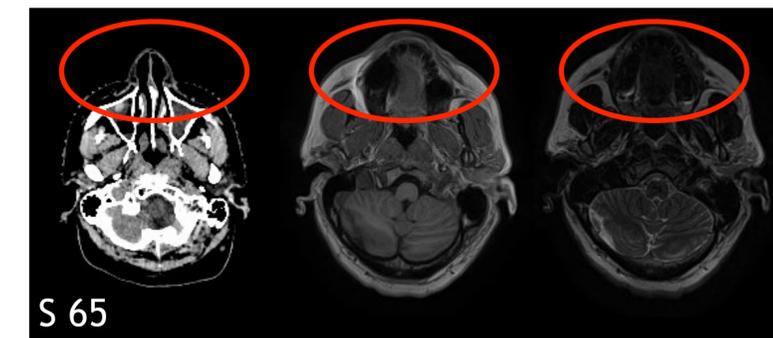
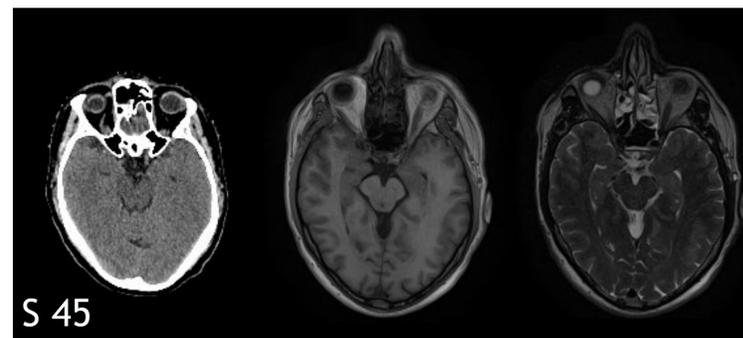
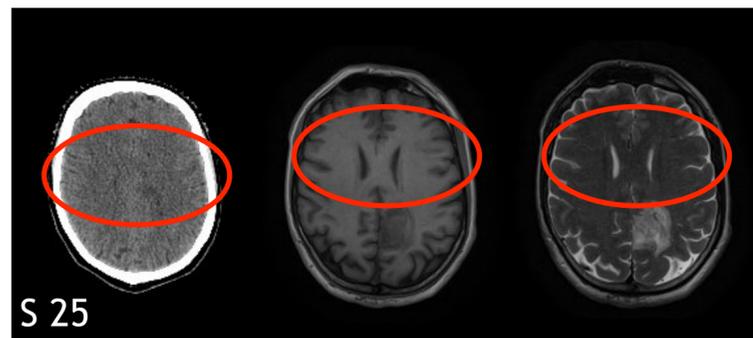
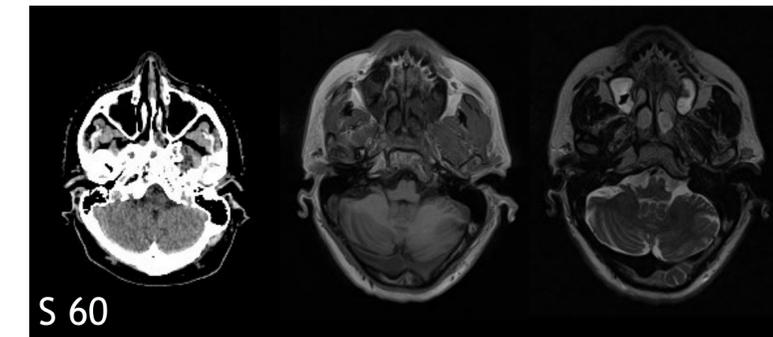
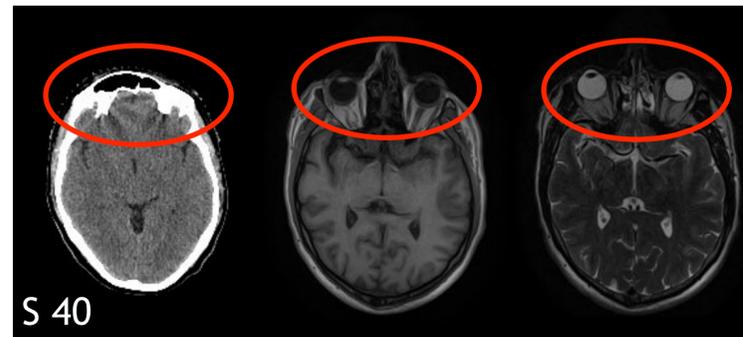
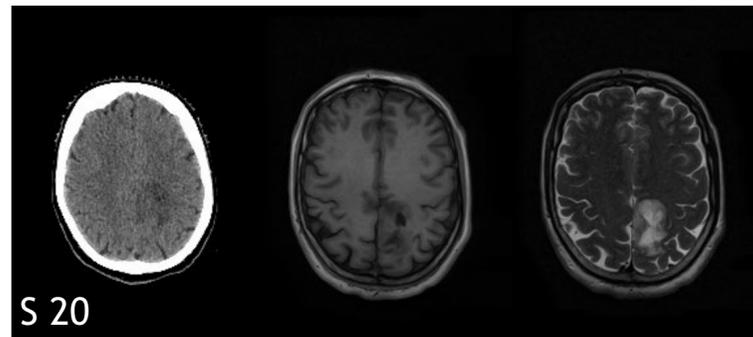
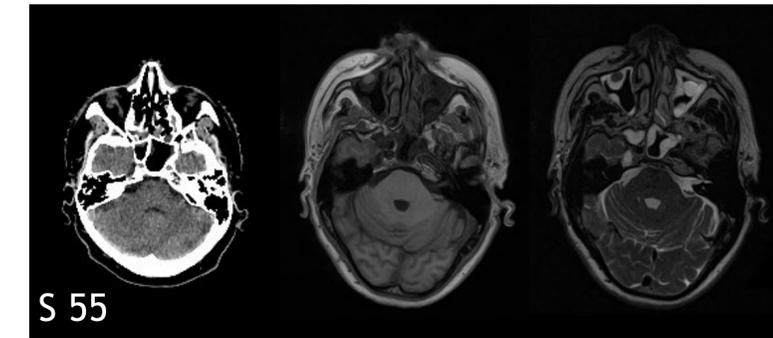
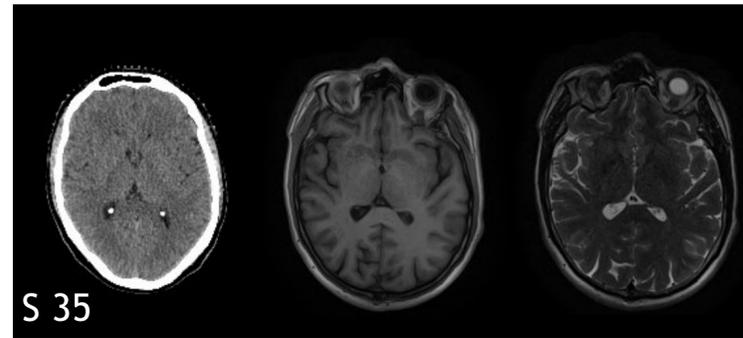
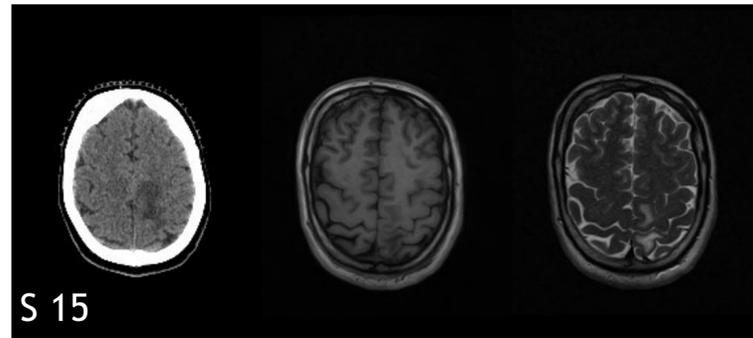


Schicht 37



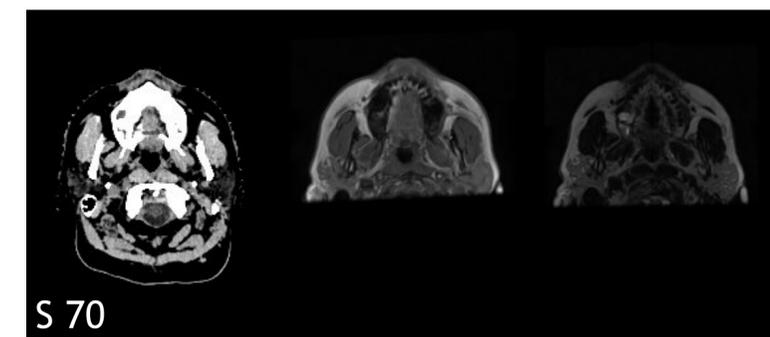
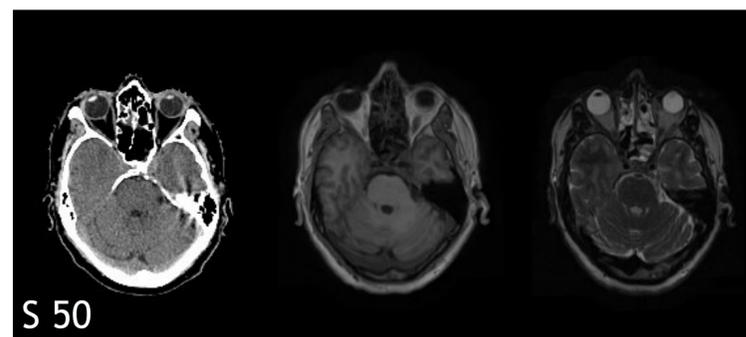
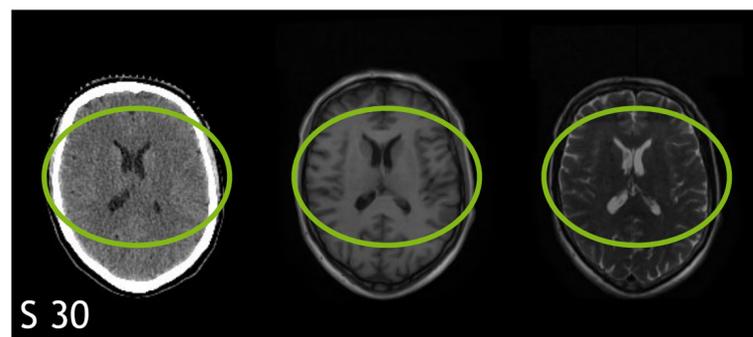
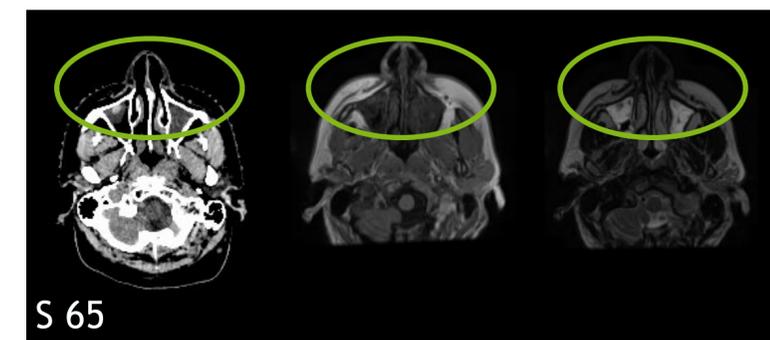
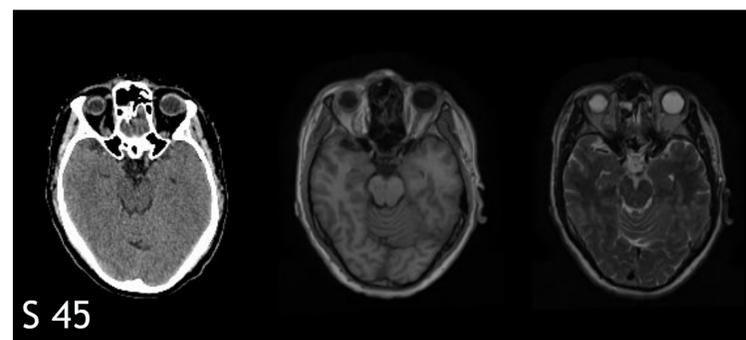
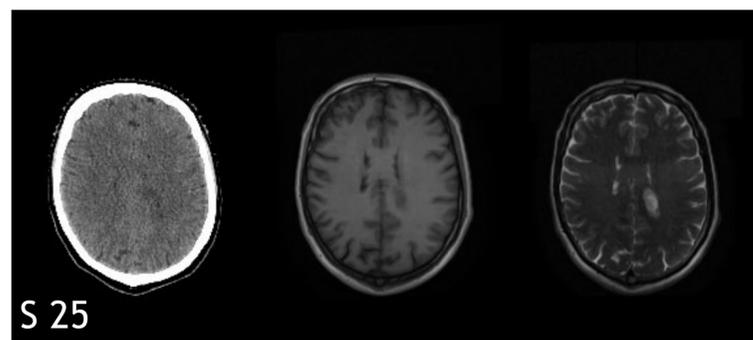
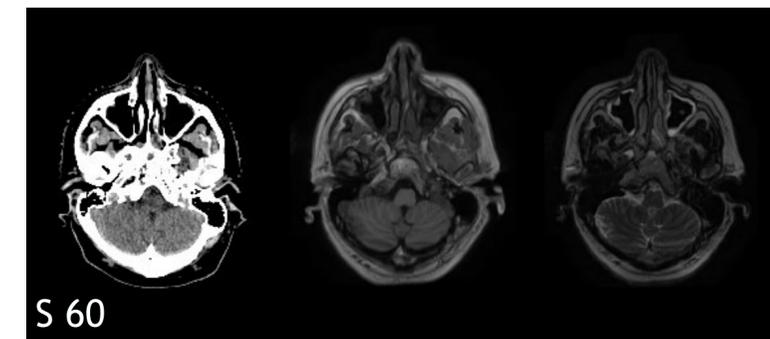
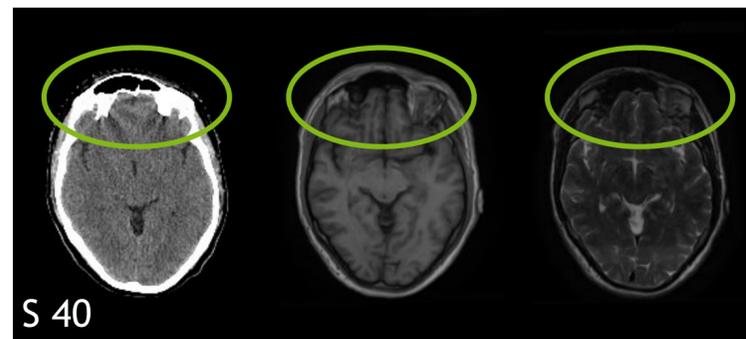
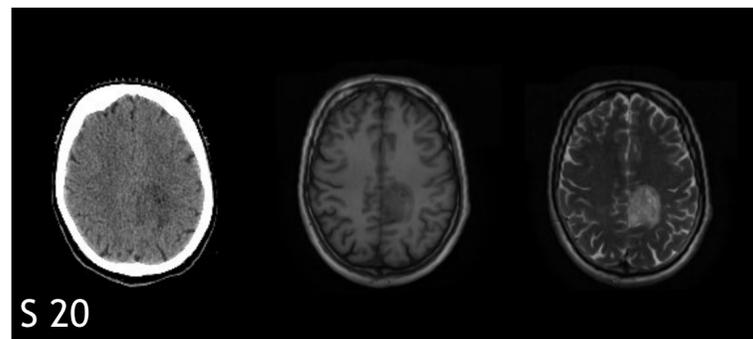
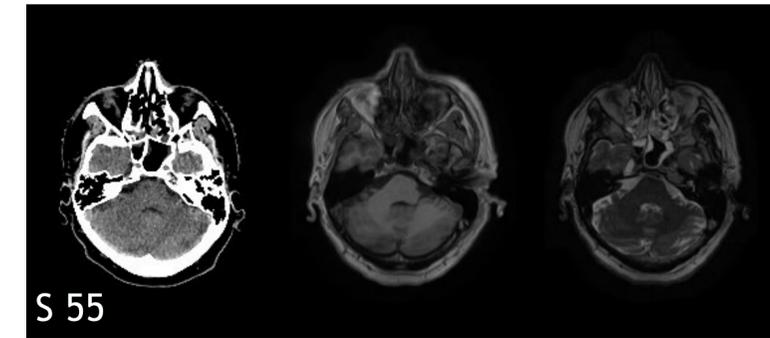
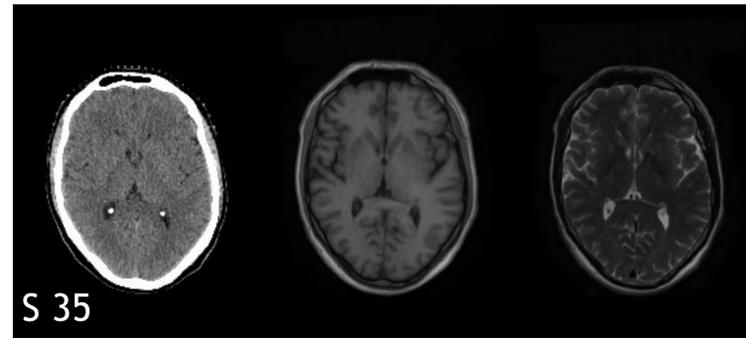
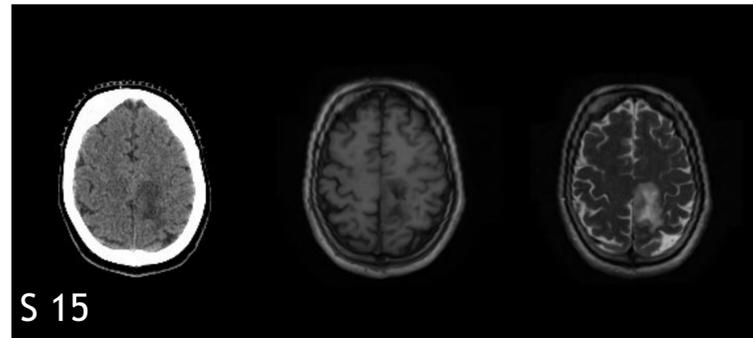
Vorbereitung der Aufnahmen

Vergleich mit alten Schichten:



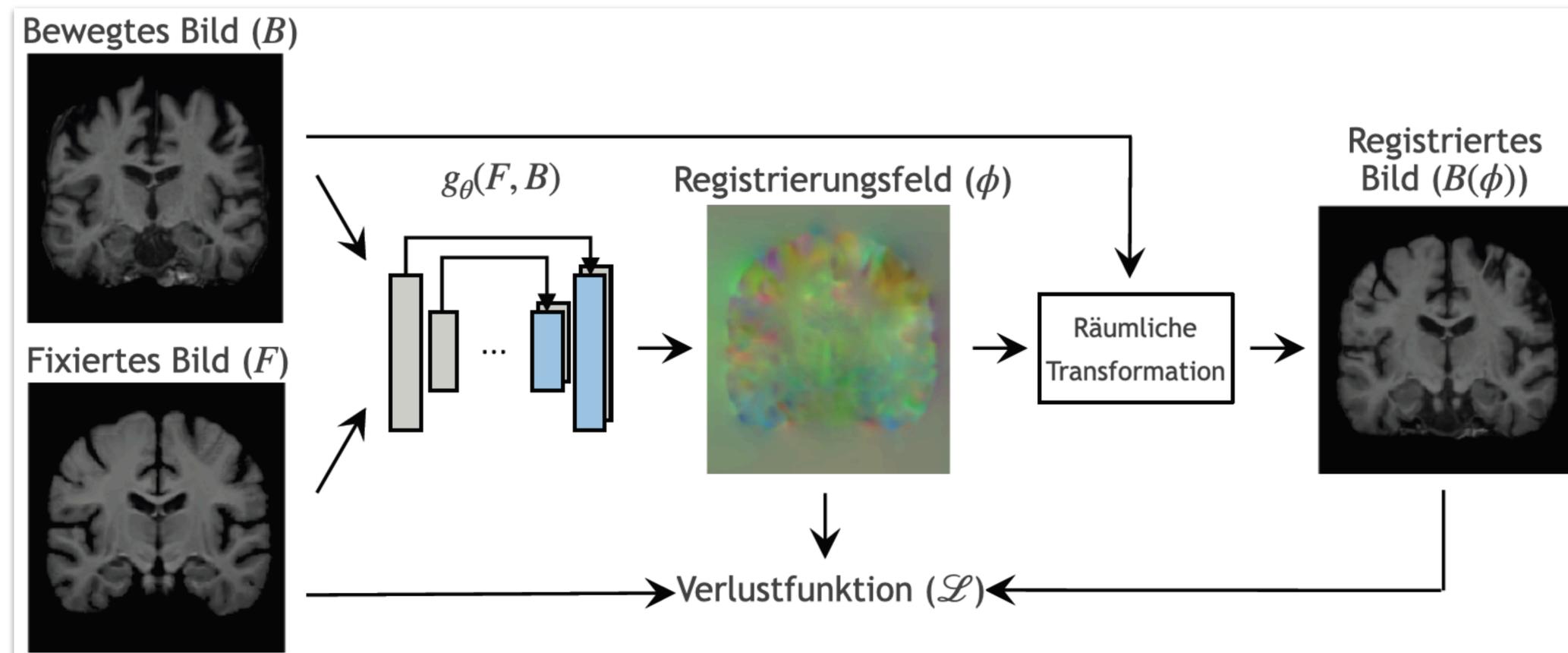
Vorbereitung der Aufnahmen

Vergleich mit neuen Schichten:



Bildregistrierung

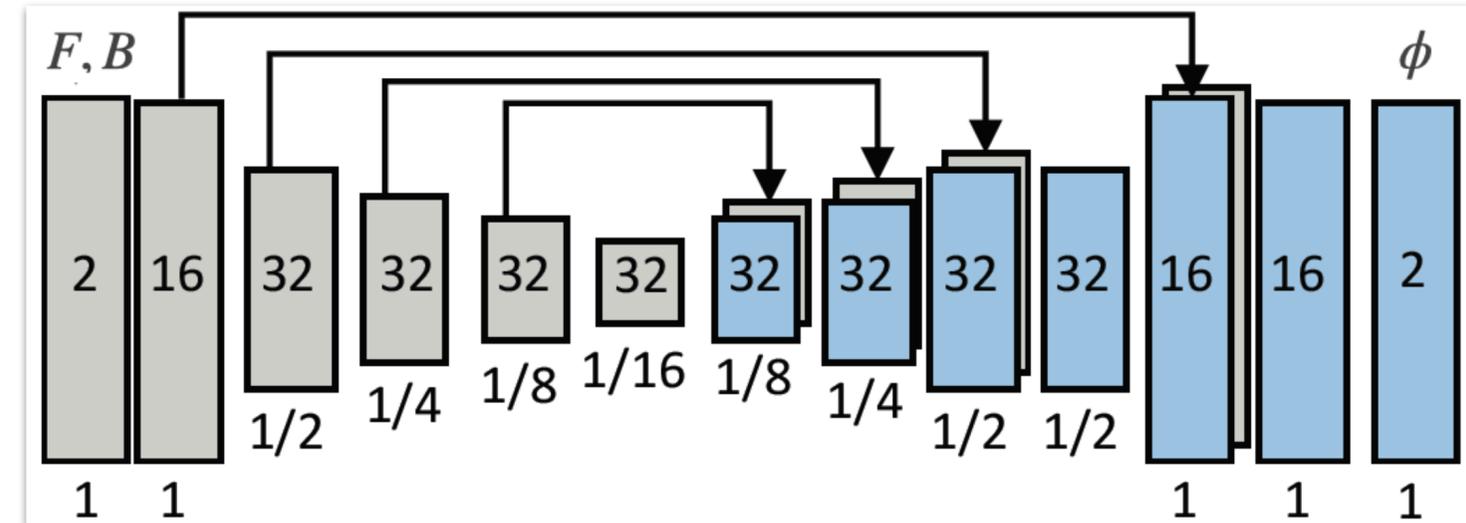
1. Input: Zwei 2D Aufnahmen (B , F)
2. Convolutional Neural Network (CNN):
 - Gefaltetes, neuronales Netz mit Filter
 - U-Net-Architektur
3. Registrierungsfeld zur räumlichen Transformation
4. Ähnlichkeitsüberprüfung von $B(\phi)$ mit Verlustfunktion:
 - Geeignete Wahl von \mathcal{L}
 - Minimierung von \mathcal{L}
5. Wiederholung des Ablaufs durch Angabe der Epochenanzahl



Quelle: „An Unsupervised Learning Model for Deformable Medical Image Registration“, G. Balakrishnan et al., 2018

Bildregistrierung

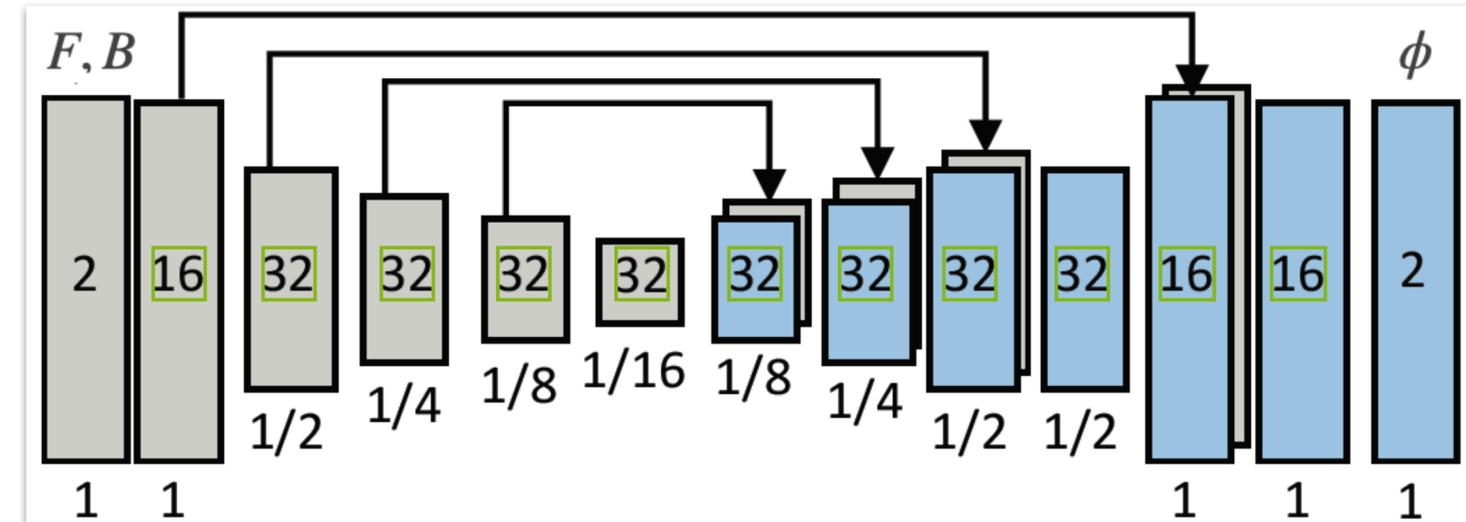
- Aufbau aus Encoder- und Decoder-Pfad
 - Anwendung von Faltungs- und Reduzierungsschichten
 - Anfangsfläche wird wieder erreicht
 - Resultat: Registrierungsfeld ϕ mit Informationen über Verschiebung jedes Pixels in x - und y -Richtung



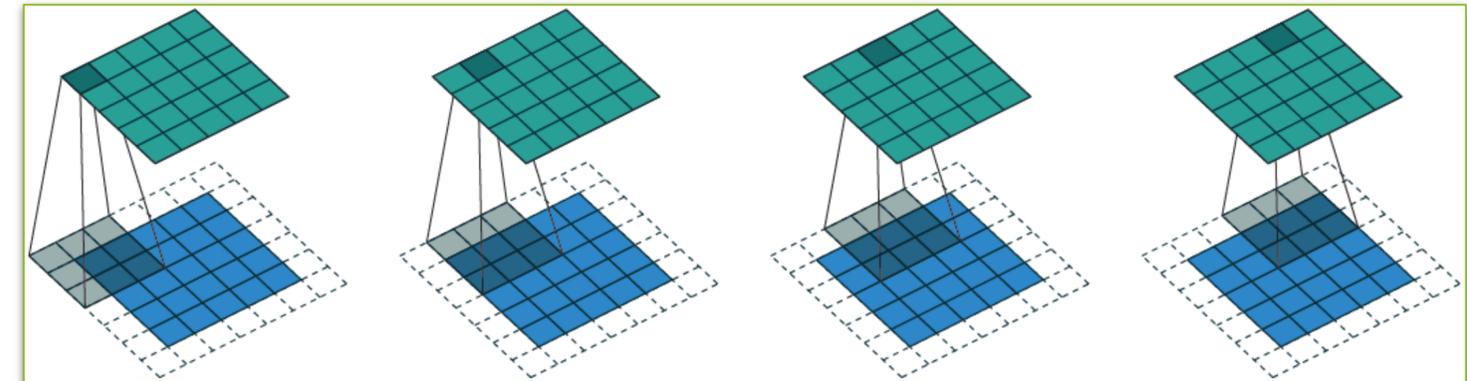
Quelle: „An Unsupervised Learning Model for Deformable Medical Image Registration“, G. Balakrishnan et al., 2018

Bildregistrierung

- Aufbau aus Encoder- und Decoder-Pfad
 - Anwendung von Faltungs- und Reduzierungsschichten
 - Anfangsfläche wird wieder erreicht
 - Resultat: Registrierungsfeld ϕ mit Informationen über Verschiebung jedes Pixels in x - und y -Richtung

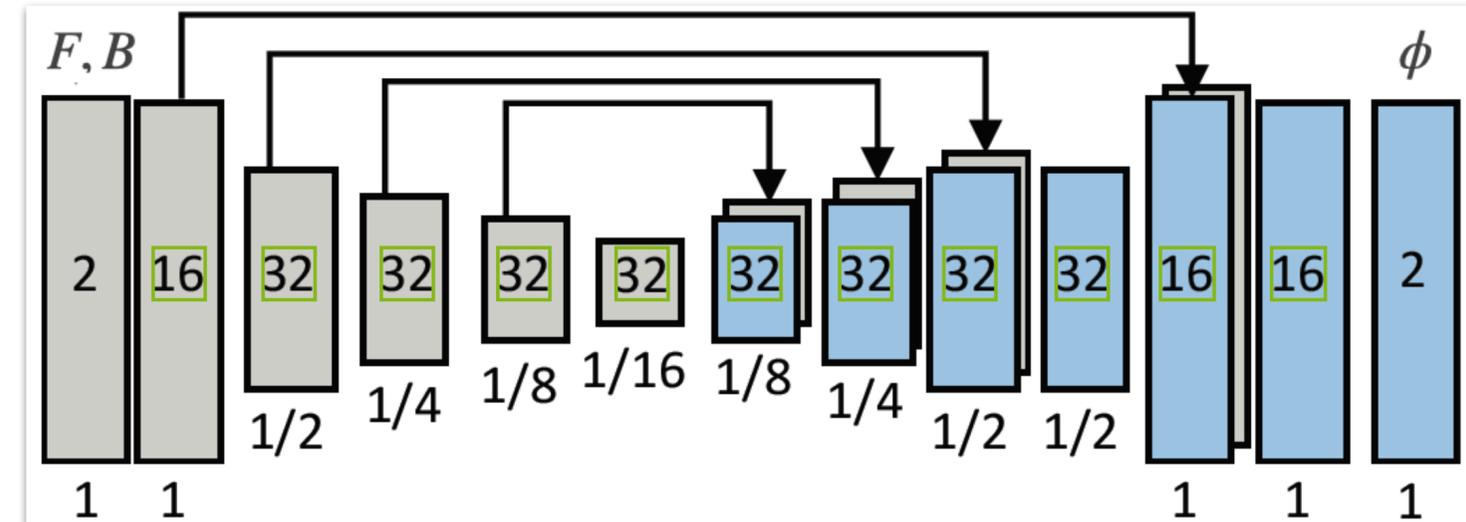


- **Faltungsschicht:**
 - Filterung der Bildmatrix mit 3×3 -Gewichtungskern
 - Regulierung der Faltung mit Polsterungsparameter und Schrittweite

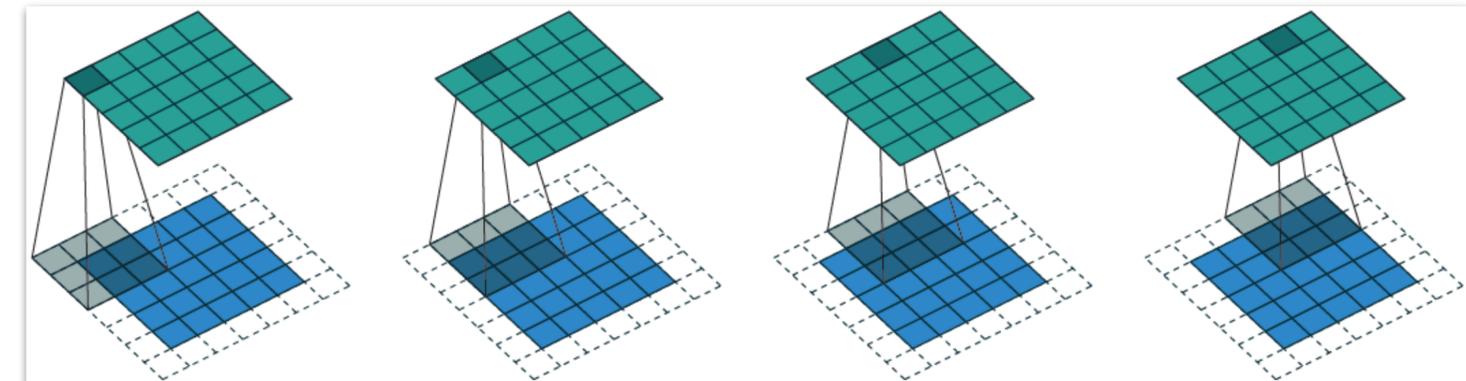


Bildregistrierung

- Aufbau aus Encoder- und Decoder-Pfad
 - Anwendung von Faltungs- und Reduzierungsschichten
 - Anfangsfläche wird wieder erreicht
 - Resultat: Registrierungsfeld ϕ mit Informationen über Verschiebung jedes Pixels in x - und y -Richtung

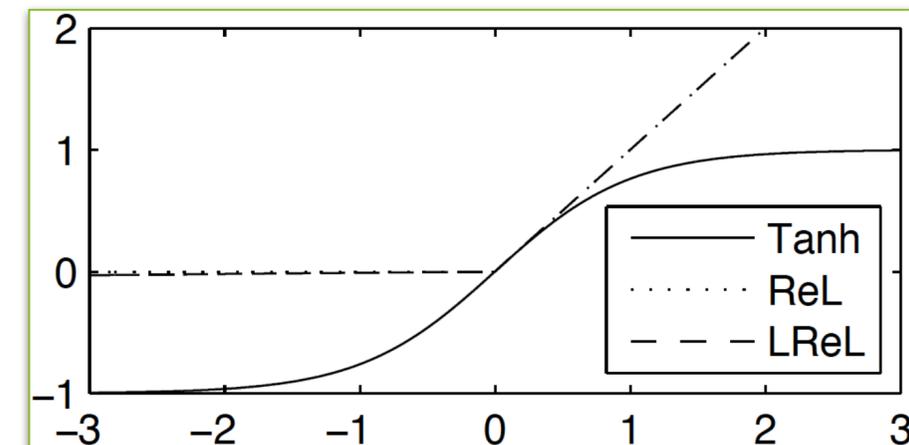


- Faltungsschicht:
 - Filterung der Bildmatrix mit 3×3 -Gewichtungskern
 - Regulierung der Faltung mit Polsterungsparameter und Schrittweite



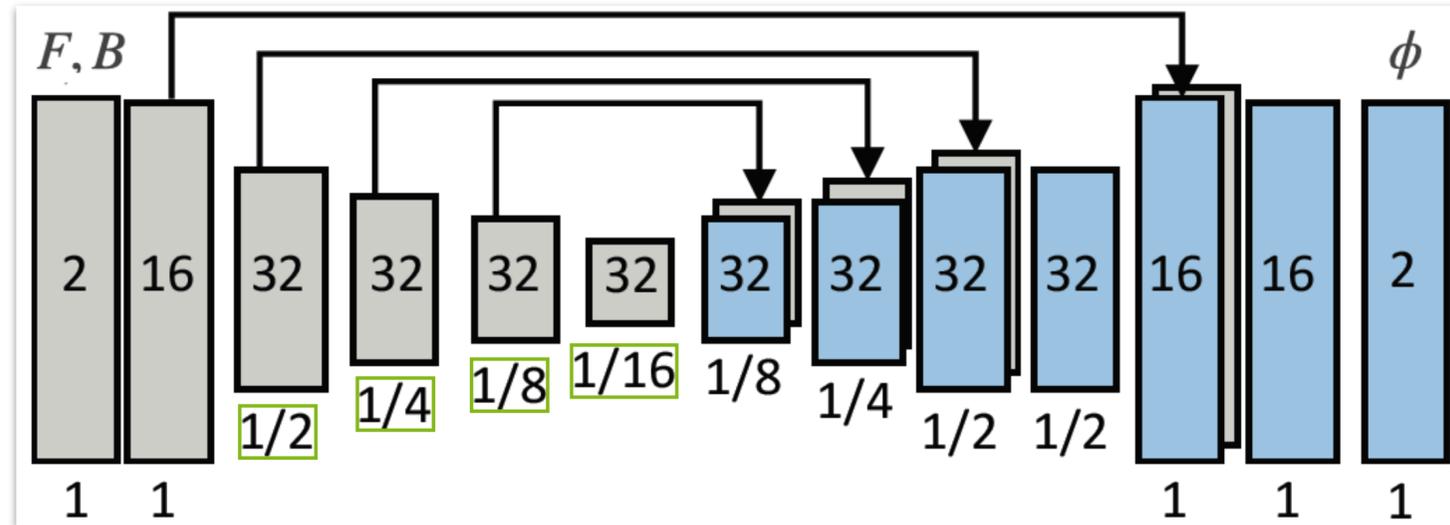
- **Aktivierungsfunktion:**
 - Abschwächung negativer Werte in der Faltungsschicht

$$\text{LeakyReLU} = \begin{cases} x & , x \geq 0 \\ 0,2 \cdot x & , x < 0 \end{cases}$$

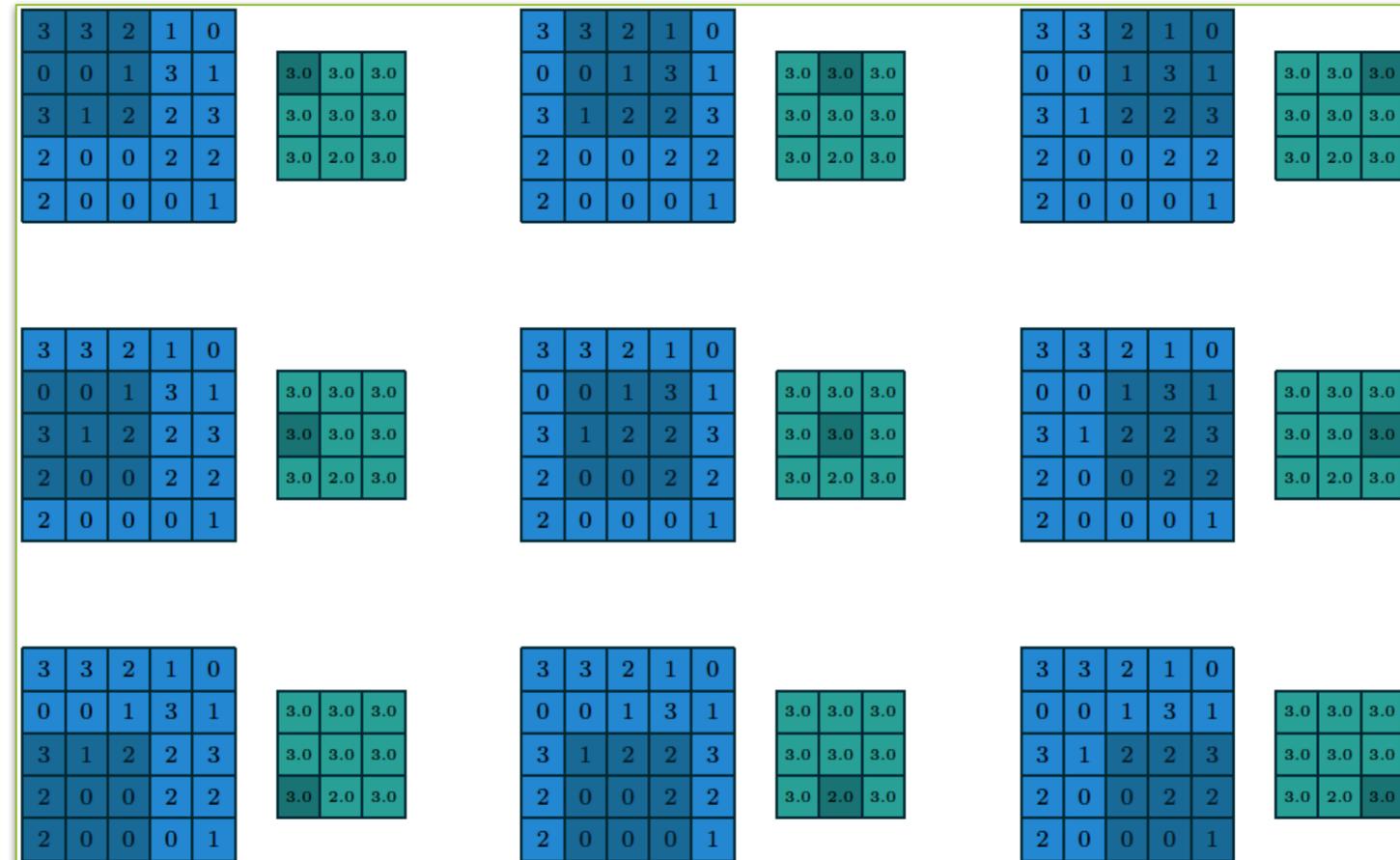


Bildregistrierung

- Reduzierungsschicht:
 - Halbierung der Größe jeder Faltungsschicht
 - Weitergabe des stärksten Werts aus 2×2 -Fenster



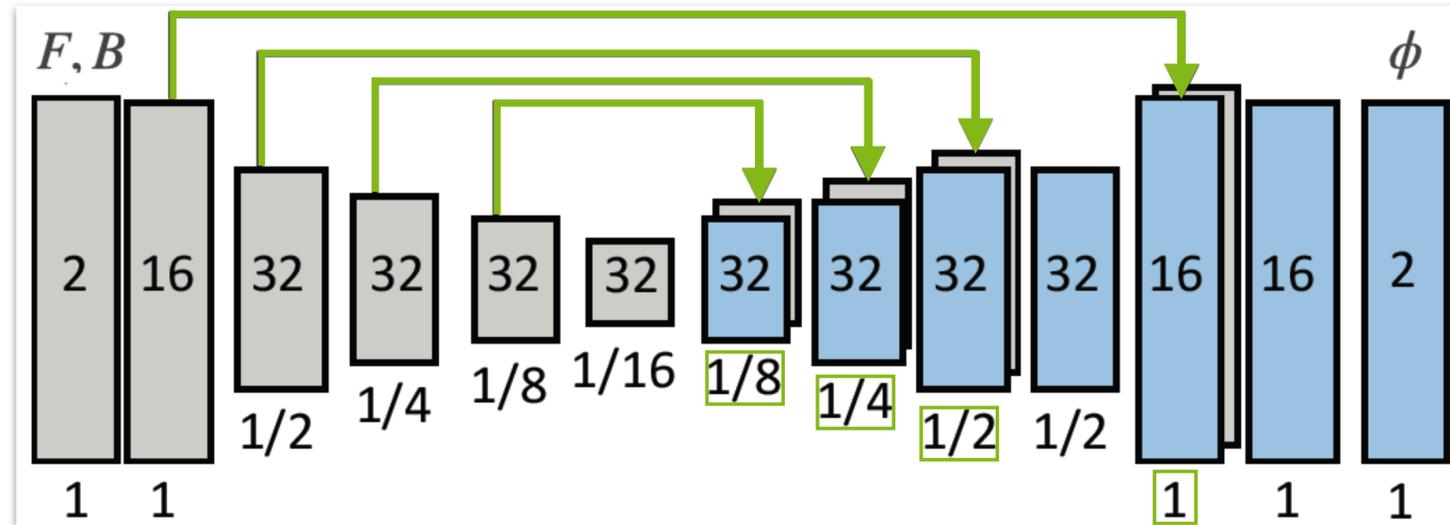
Quelle: „An Unsupervised Learning Model for Deformable Medical Image Registration“, G. Balakrishnan et al., 2018



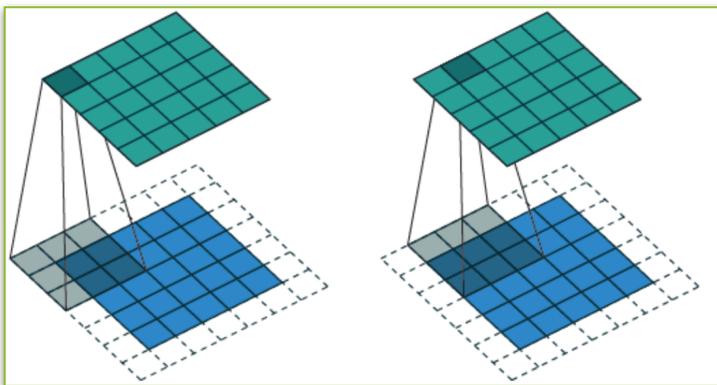
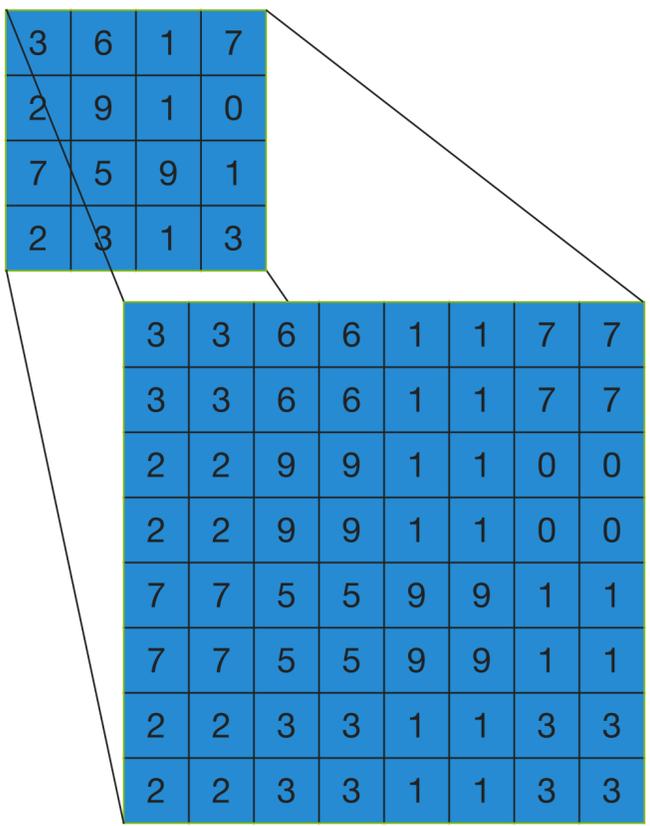
Quelle: „A guide to convolution arithmetic for deep learning“, V. Dumoulin and F. Visin, 2018

Bildregistrierung

- Reduzierungsschicht:
 - Halbierung der Größe jeder Faltungsschicht
 - Weitergabe des stärksten Werts aus 2×2 -Fenster
- Auffaltungsschicht:
 - Verdopplung der Größe
 - Anwendung von Faltungen mit Verkettung aus dem Encoder-Pfad



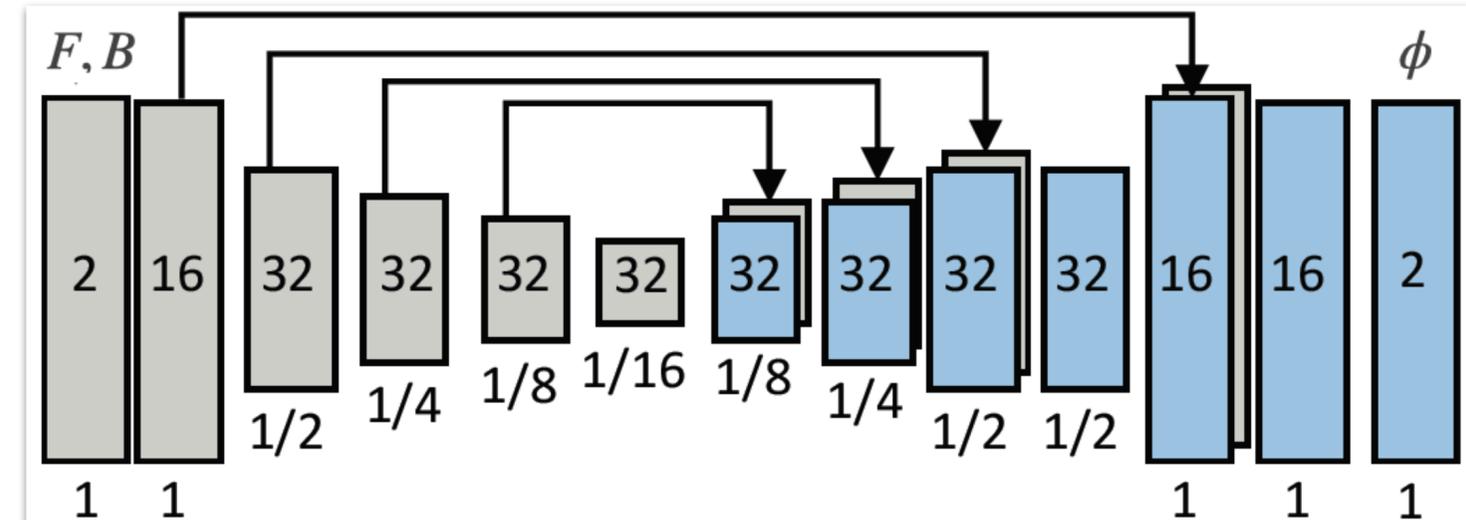
Quelle: „An Unsupervised Learning Model for Deformable Medical Image Registration“, G. Balakrishnan et al., 2018



Quelle: „A guide to convolution arithmetic for deep learning“, V. Dumoulin and F. Visin, 2018

Bildregistrierung

- Reduzierungsschicht:
 - Halbierung der Größe jeder Faltungsschicht
 - Weitergabe des stärksten Werts aus 2×2 -Fenster
- Auffaltungsschicht:
 - Verdopplung der Größe
 - Anwendung von Faltungen mit Verkettung aus dem Encoder-Pfad
- Gewichte aus den Faltungen:
 - Anfangsgewichte werden vorgegeben oder zufällig gewählt
 - Erlernen weiterer Gewichte während des Trainierens
 - Wiederholung nach Berechnung der Verlustfunktion



Quelle: „An Unsupervised Learning Model for Deformable Medical Image Registration“, G. Balakrishnan et al., 2018

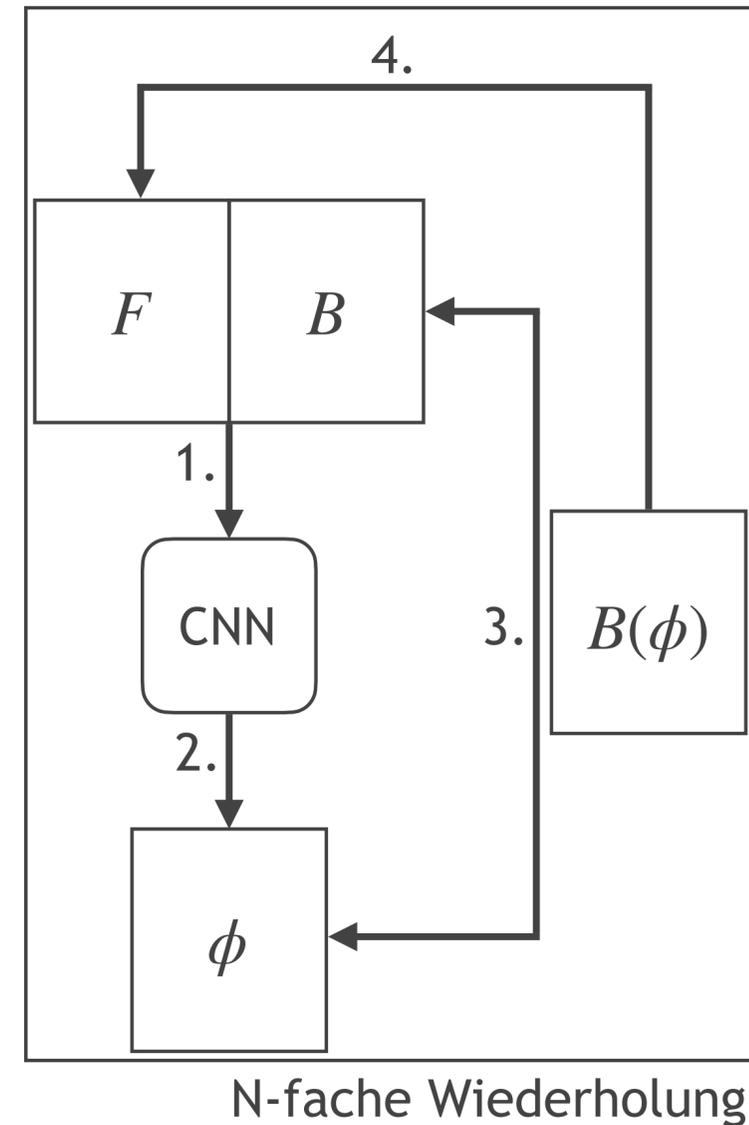
Bildregistrierung

- Räumliche Transformation von B mit ϕ durch bilineare Interpolation
- Verlustfunktion: $\mathcal{L}(F, B, \phi) = \mathcal{L}_{NCC}(F, B(\phi)) + \lambda \mathcal{L}_G(\phi)$
 - Untersuchung von $B(\phi)$ und ϕ
 - Verwendung einer unbeaufsichtigten Verlustfunktion
 - Minimierung von $\mathcal{L}(F, B, \phi)$
- Komponenten:
 - Ähnlichkeitsüberprüfung mittels NCC

$$\mathcal{L}_{NCC}(F, B(\phi)) = -NCC(F, B(\phi))$$

- Glattheit von ϕ durch Regulierungsparameter λ

$$\mathcal{L}_G(\phi) = \sum_p \left\| \nabla g_\theta(p) \right\|^2$$

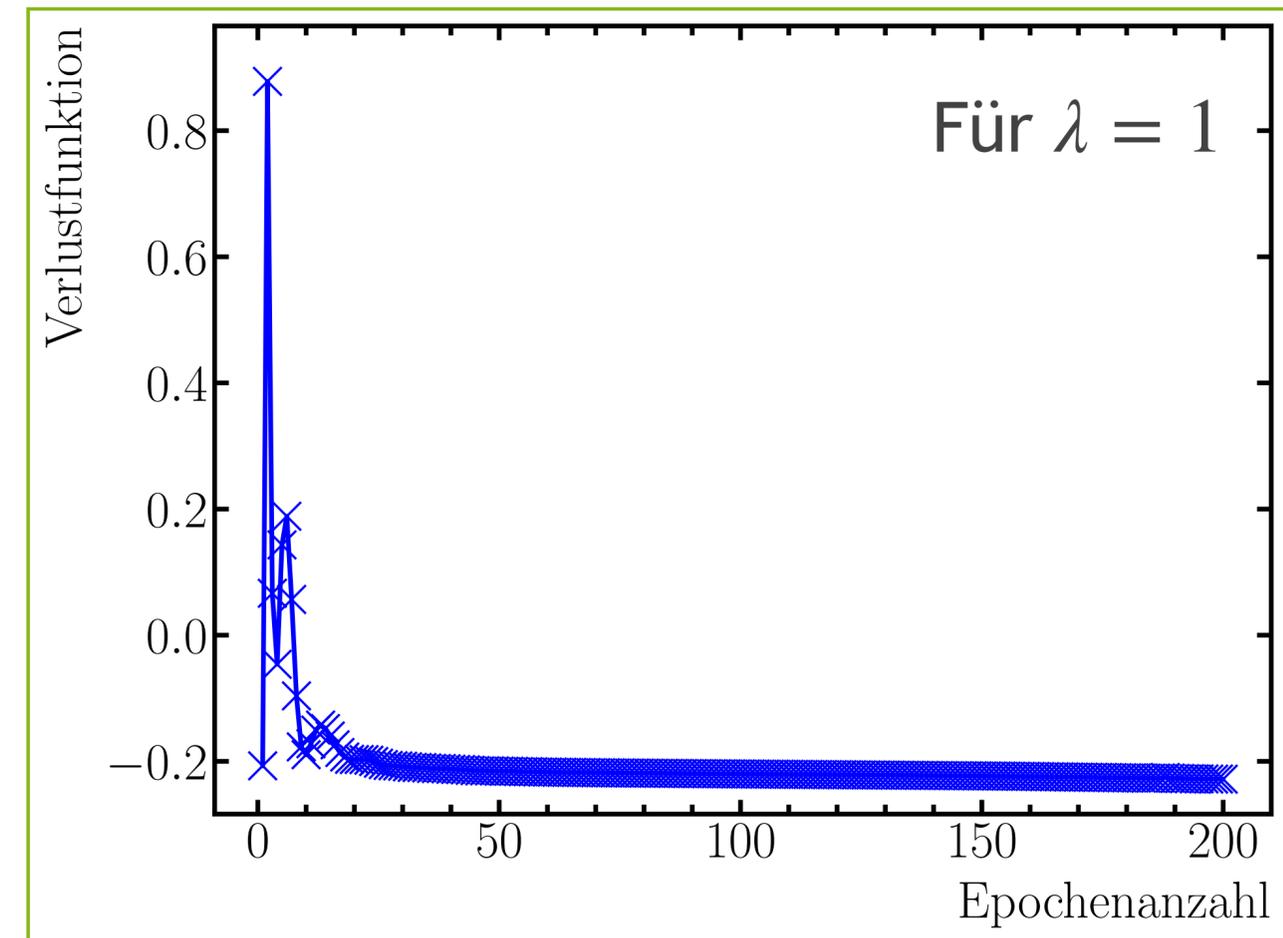
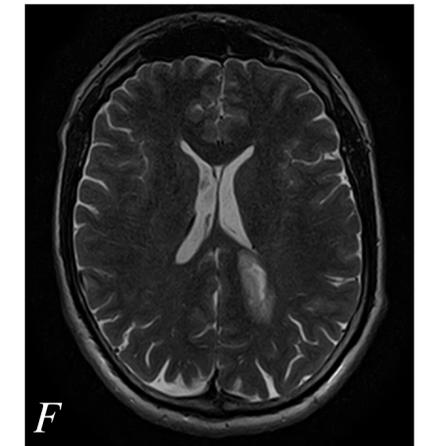
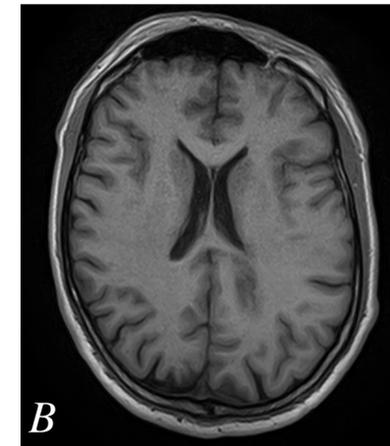


- Variation der Parameter:
 - Epochenanzahl: bis 200
 - Regulierungsparameter: $\lambda = \{1, 2, 3, 4\}$

Bildregistrierung

- **Evaluierung der Verlustfunktion:**
 - Werte der Verlustfunktion in Abhängigkeit von der Epoche
 - Ähnliches Verhalten für alle λ -Werte:
 - Anfangs: Starke Variation (0 bis 20 Epochen)
 - Hohe Epochenanzahl: Grenzwertannäherung

Input: T_1 (B) und T_2 (F)



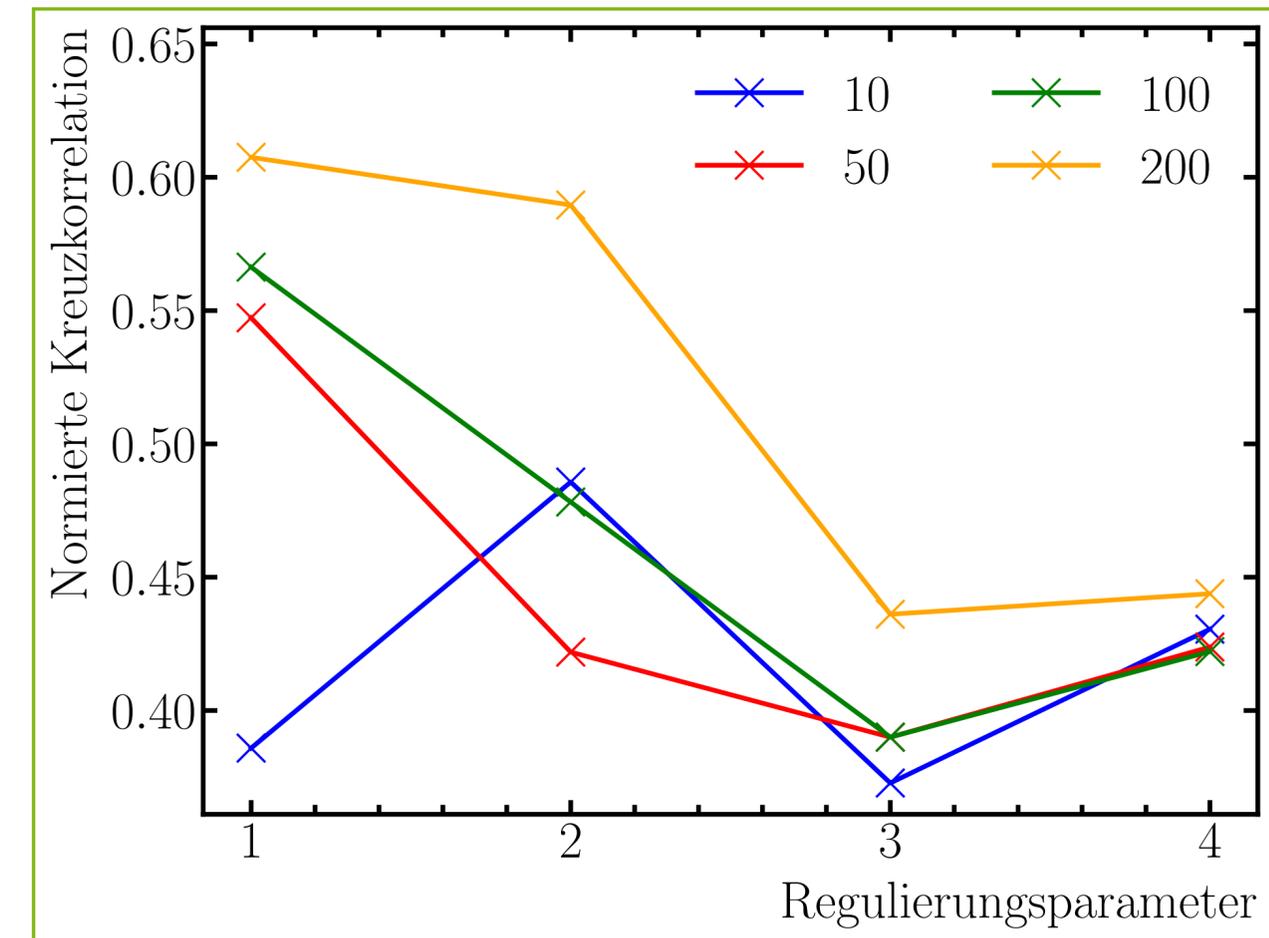
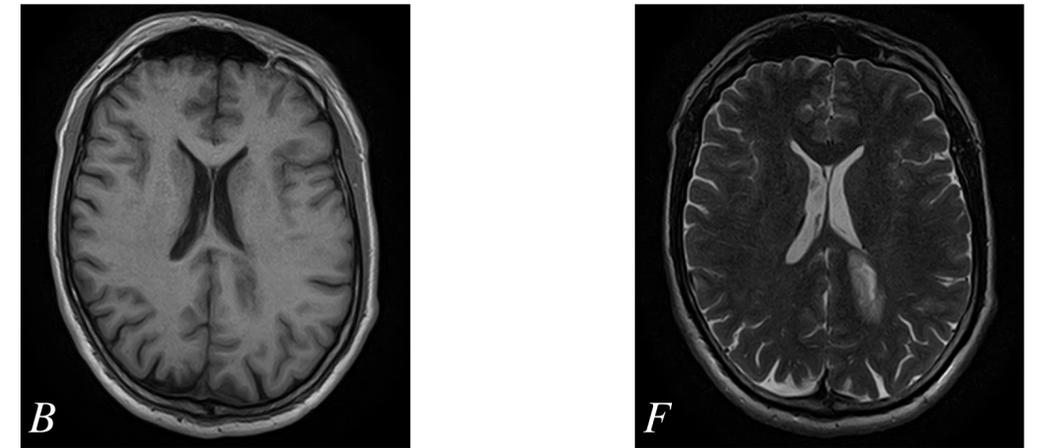
Bildregistrierung

- Evaluierung der Verlustfunktion:
 - Werte der Verlustfunktion in Abhängigkeit von der Epoche
 - Ähnliches Verhalten für alle λ -Werte:
 - Anfangs: Starke Variation (0 bis 20 Epochen)
 - Hohe Epochenanzahl: Minimierung der Verlustfunktion

- **Evaluierung des λ -Parameters:**

- Bessere Übereinstimmung von $B(\phi)$ und F mit höherer Epoche
- Höhere λ -Werte liefern kleinere NCC -Werte

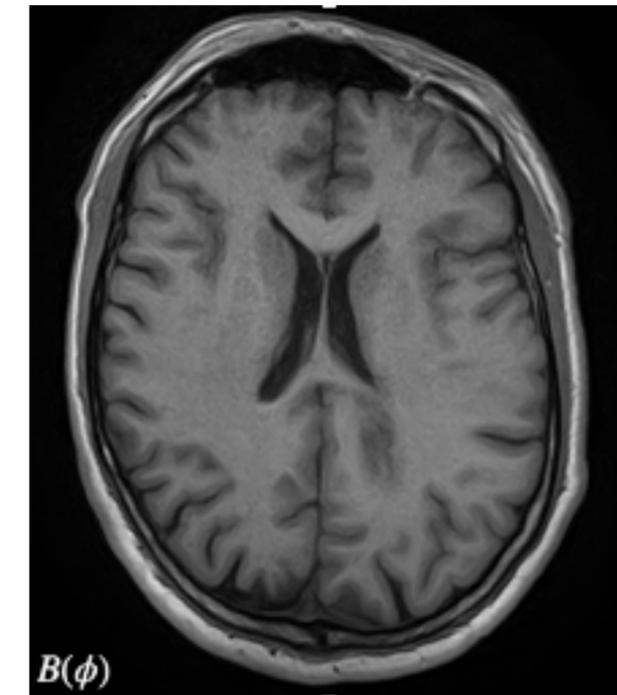
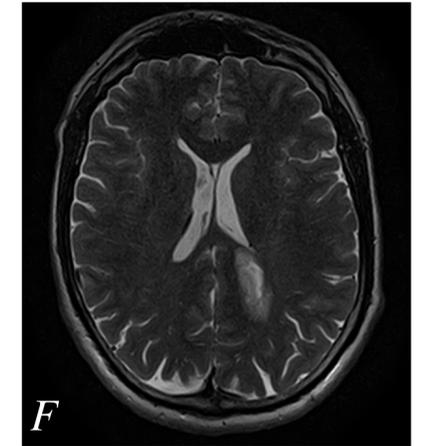
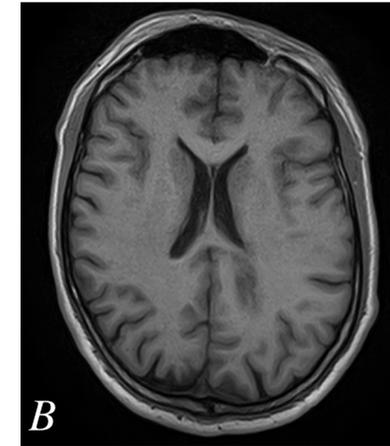
Input: $T_1 (B)$ und $T_2 (F)$



Bildregistrierung

- Evaluierung der Verlustfunktion:
 - Werte der Verlustfunktion in Abhängigkeit von der Epoche
 - Ähnliches Verhalten für alle λ -Werte:
 - Anfangs: Starke Variation (0 bis 20 Epochen)
 - Hohe Epochenanzahl: Minimierung der Verlustfunktion
- Evaluierung des λ -Parameters:
 - Bessere Übereinstimmung von $B(\phi)$ und F mit höherer Epoche
 - Höhere λ -Werte liefern kleinere NCC -Werte
- **Auswahl für die Bildfusion:**
 - $\lambda = 1$, Epochenanzahl 200

Input: T_1 (B) und T_2 (F)



Interaktive Vorführung

- Bildregistrierungsskripte
- Verwendung von HTCondor ([Hilfreiche Befehle](#))
- Ausführen von Skripten
- Begutachtung der Resultate

3. Aufgabe

- Nutze die 3D Bilder T1.npz und T2.npz aus dem Ordner *programmierkurs/ImageProcessing* als Input für die Bildregistrierung mit *medical-image-fusion/registration/scripts/tf/train.py*!
 - Erstelle einen neuen *branch* „programmierkurs“ und schreibe die notwendigen Skripte um!
 - Epochenanzahl = 200, Schritte pro Epoche = 1, $\lambda = 1$

4. Aufgabe

- Verwende nach dem Trainieren *register.py* zum Erzeugen des deformierten Bildes!
 - Schau dir die Resultate im Vergleich an:
 - Wie groß ist der NCC-Wert?
 - Welche Strukturen haben sich nach der Registrierung einander angepasst?

Ende

Viel Erfolg bei den Bachelorarbeiten!

Bei Fragen gerne im
privaten oder Bachelor 2021
Mattermostchannel schreiben!

Oder eine E-Mail an:
alexander.ratke@tu-dortmund.de