nnUStappenplan: NnU-net installeren en gebruiken (Luna)

# Samenvatting zonder afbeeldingen

**I. Installatie van nnU-Net**

1. **Log in op de Luna-computer**  
   Navigeer naar je my-scratch map.
2. **Maak een Virtual Environment**  
   Maak een virtuele omgeving aan voor het installeren van nnU-Net.
   * Volg de instructies op de volgende link: [Install packages in a virtual environment using pip and venv - Python Packaging User Guide](https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/).
3. **Laad de juiste versie van CUDA**  
   Gebruik het volgende command om CUDA te laden:

module load cuda 12.1

1. **Installeer PyTorch**  
   Ga naar de PyTorch Start Locally website en selecteer de juiste configuratie voor jouw installatie. Gebruik het volgende command:

pip3 install torch torchvision torchaudio

1. **Activeer de Virtual Environment**  
   Voordat je verdergaat, activeer je de virtual environment (mits je directory goed is ingesteld):

source py\_env/.venv/bin/activate

1. **Installeer nnU-Net**  
   Installeer nnU-Net met de volgende command:

pip install nnunetv2

1. **Optionele Stap (Voor code-aanpassingen)**  
   Clone de nnU-Net repository als je de code wilt aanpassen:

git clone https://github.com/MIC-DKFZ/nnUNet.git

cd nnUNet

pip install -e .

**II. Mappenstructuur Instellen**

Om nnU-Net goed te laten werken, moet de data in een specifieke structuur worden opgeslagen.

1. **Maak een hoofddirectory voor je data**  
   Maak drie submappen aan binnen deze directory:
   * nnUnet\_raw: Hierin plaats je je ruwe data.
   * nnUnet\_preprocessed: Deze map wordt automatisch gevuld door nnU-Net tijdens preprocessing.
   * nnUnet\_results: Hier worden de resultaten van het trainen opgeslagen.
2. **Maak een submap voor je dataset**  
   Gebruik de [documentatie van nnU-Net](https://github.com/MIC-DKFZ/nnUNet/blob/master/documentation/dataset_format.md) om de naamgeving en structuur te controleren. Doe dit in de map nnUnet\_raw.
3. **Maak drie submappen in je dataset directory**:
   * imagesTr: Bevat de trainingsdata (MRI-scans).
   * labelsTr: Bevat de segmentatiemasks (labels).
   * imagesTs: Bevat de testafbeeldingen (MRI-scans).
4. **Maak het dataset.json bestand aan**  
   Maak in de dataset directory een bestand aan genaamd dataset.json. Dit bestand bevat de metadata zoals beschreven in de [GitHub documentatie](https://github.com/MIC-DKFZ/nnUNet/blob/master/documentation/dataset_format.md). Je kunt meerdere labels toevoegen, afhankelijk van je data.

**III. Voorbereiden van Data voor nnU-Net**

Als je beelden en labels nog in DICOM-format zijn, moeten deze worden geconverteerd naar NIfTI-formaat.

1. **Open PyCharm**  
   Laad PyCharm via het terminal commando:

module load pycharm/community-23.1.2

pycharm.sh

1. **Installeer dicom2nifti**  
   Installeer het dicom2nifti-pakket in je virtual environment:

pip install dicom2nifti

1. **Gebruik een conversiescript**  
   Maak een Python-script (zie convim\_dicom2nifti.py en convla\_dicom2nifti.py) voor het automatisch omzetten van DICOM naar NIfTI. Hierin zijn for-loops gebruikt om alle mappen met DICOM-afbeeldingen te converteren.
2. **Controleer de richting van de data**  
   Gebruik fsleyes om te controleren of de richting van de beelden en de labels correct is:

fsleyes image.nii.gz label.nii.gz

1. **Corrigeer mask-values indien nodig**  
   Als de waarden in de masks niet kloppen (nnU-Net verwacht integers), pas deze aan met een script. Voorbeeldcode is te vinden in image\_label\_012.py.
2. **Converteer 4D MRI naar 3D**  
   Gebruik een script zoals 4D\_to\_3D.py om de eerste tijdstap van cine-MRI-data te isoleren, aangezien de segmentatie in mijn geval op die tijdstap was gebaseerd.

**IV. Slurm voor nnU-Net Training**

1. **Interactive Slurm Sessies**  
   Gebruik een Slurm interactieve sessie voor het trainen van nnU-Net. Dit is handig voor kleinere datasets of kortere trainingstijden:

salloc --job-name "interactive-shell" --partition=luna-gpu-short --cpus-per-task 8 --mem 10G --time 1:00:00 --gres=gpu:a100:1

**V. Preprocessing**

1. **Voer de Preprocessing Uit**  
   Gebruik de volgende command om preprocessing uit te voeren. Dit checkt ook of de naamgeving en structuur van de data correct is:

nnUNetv2\_plan\_and\_preprocess -d 001 --verify\_dataset\_integrity

**VI. Training**

1. **Start de Training**  
   Begin met trainen van het model. Bijvoorbeeld met 50 epochs:

nnUNetv2\_train 001 3d\_fullres 0 -tr nnUNetTrainer\_50epochs

**VII. Voorspellingen Maken**

1. **Gebruik het getrainde model om voorspellingen te doen**  
   Vul het pad in naar je input en output mappen:

nnUNetv2\_predict -i /path/to/input/images -o /path/to/output -d 001 -c 3d\_fullres -tr nnUNetTrainer\_50epochs -f 0

**VIII. Evaluatie van het Model**

1. **Vergelijk voorspellingen met de ground-truth segmentaties**  
   Gebruik het volgende commando om evaluaties uit te voeren. Zorg ervoor dat je het juiste pad naar je dataset.json en plans.json toevoegt:

nnUNetv2\_evaluate\_folder /path/to/ground\_truth/ /path/to/predictions/ -djfile /path/to/dataset.json -pfile /path/to/plans.json

# Gedetailleerde uitleg met afbeeldingen

Op de volgende websites staat stap voor stap uitgelegd hoe installatie en gebruik van nnU-net werkt. Ik zou hier vooral ook naar kijken.

* [How I Use nnUNet for Medical Image Segmentation: A Comprehensive Guide - PYCAD](https://pycad.co/nnunet-for-medical-image-segmentation/)
* [nnUNet/documentation/installation\_instructions.md at master · MIC-DKFZ/nnUNet · GitHub](https://github.com/MIC-DKFZ/nnUNet/blob/master/documentation/installation_instructions.md)
* [nnUNet/documentation/how\_to\_use\_nnunet.md at master · MIC-DKFZ/nnUNet · GitHub](https://github.com/MIC-DKFZ/nnUNet/blob/master/documentation/how_to_use_nnunet.md)

**I. Installatie van nnU-Net**

1. **Log in op de Luna-computer**  
   Navigeer naar je my-scratch map.
2. **Maak een Virtual Environment**  
   Maak een virtuele omgeving aan voor het installeren van nnU-Net.
   * Volg de instructies op de volgende link: [Install packages in a virtual environment using pip and venv - Python Packaging User Guide](https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/).
3. **Laad de juiste versie van conda**  
   Gebruik het volgende command om CUDA te laden:

module load cuda/12.1  
  
*(je kunt dingen in je terminal copy/pasten met control + shift c of v)*

1. **Activeer de Virtual Environment**  
   Voordat je verdergaat, activeer je de virtual environment (mits je directory goed is ingesteld):

source py\_env/bin/activate

Ik heb nu een conda environment gemaakt omdat daarin alle versies van de software die je installeert worden aangepast aan elkaar. Op die manier zijn ze dus compatible met elkaar. Het conda environment is gemaakt in: /home/rnga/eaalbregt/.conda/envs/.

Je kunt het activeren door: conda activate nnUNet\_done te typen.

Hier staat hoe je het conda environment kan maken: [Managing environments — UN 24.11.1.dev21 documentation](https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html)

1. **Installeer PyTorch**  
   Ga naar de [Start Locally | PyTorch](https://pytorch.org/get-started/locally/) website en selecteer de juiste configuratie voor jouw installatie. Gebruik het volgende command:

pip3 install torch torchvision torchaudio

1. **Installeer nnU-Net**  
   Installeer nnU-Net met de volgende command:

pip install nnunetv2

1. **Optionele Stap (Voor code-aanpassingen)**  
   Clone de nnU-Net repository als je de code echt wilt aanpassen:

git clone https://github.com/MIC-DKFZ/nnUNet.git

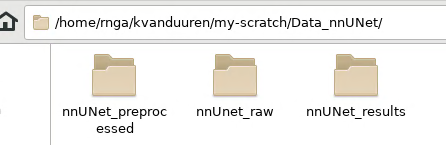
cd nnUNet

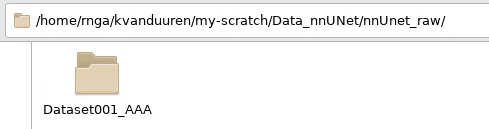
pip install -e .

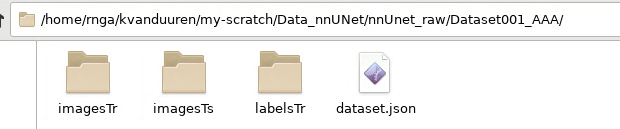
**II. Mappenstructuur Instellen**

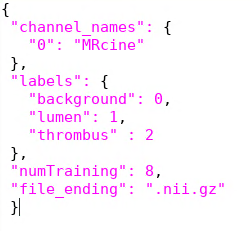
Om nnU-Net goed te laten werken, moet de data in een specifieke structuur worden opgeslagen, let hier goed op!! Hier nogmaals de link waar je dit kan vinden: [nnUNet/documentation/dataset\_format.md at master · MIC-DKFZ/nnUNet · GitHub](https://github.com/MIC-DKFZ/nnUNet/blob/master/documentation/dataset_format.md)

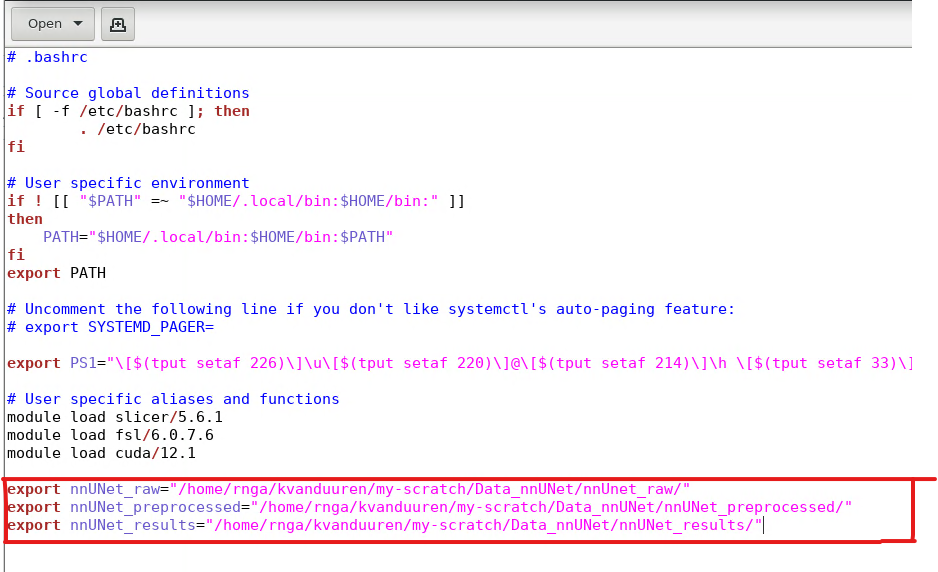
1. **Maak een hoofddirectory voor je data (/Data\_nnUNet/)**  
   Maak drie submappen aan binnen deze directory:
   * nnUnet\_raw: Hierin plaats je je ruwe data (enige map waar je zelf dingen in moet zetten).
   * nnUnet\_preprocessed: Deze map wordt automatisch gevuld door nnU-Net tijdens preprocessing.
   * nnUnet\_results: Hier worden de resultaten van het trainen opgeslagen.



1. **Maak een submap voor je dataset**  
   Gebruik de [documentatie van nnU-Net](https://github.com/MIC-DKFZ/nnUNet/blob/master/documentation/dataset_format.md) om de naamgeving en structuur te controleren. Doe dit in de map nnUnet\_raw. {CASE\_IDENTIFIER}\_{XXXX}.{FILE\_ENDING}.  
     
   
2. **Maak drie submappen in je dataset directory**:
   * imagesTr: Bevat de trainingsdata (MRI-scans).
   * labelsTr: Bevat de segmentatiemasks (labels).
   * imagesTs: Bevat de testafbeeldingen (MRI-scans).



1. **Maak het dataset.json bestand aan**  
   Maak in de dataset directory een bestand aan genaamd dataset.json. Dit bestand bevat de metadata zoals beschreven in de [GitHub documentatie](https://github.com/MIC-DKFZ/nnUNet/blob/master/documentation/dataset_format.md). Je kunt meerdere labels toevoegen, afhankelijk van je data. Zo zag mijn file eruit:   
     
   
2. Set de environment variables, dit heeft het nnU-Net nodig om te weten waar de data staat. Ga naar .bashrc file in de home directory, alles wat hierin staat wordt altijd standaard uitgevoerd zo gauw je de terminal opent. Hier is het handig om drie dingen zelf toe te voegen:
   * De code in het groene blok zorgt voor een overzichtelijkere opmaak van je terminal
   * De code in het blauwe blok zijn modules die je vaak/altijd nodig hebt en die hierdoor automatisch al worden ingeladen. Nu kan je ze voortaan gemakkelijker openen vanuit je terminal.
   * De code in het rode blok bestaat uit ‘export’ gevolgd door het path naar de respectievelijke mapjes die het nnU-Net nodig heeft, zie het voorbeeld hieronder.

Sla dit op en sluit de file.   
  


**III. Voorbereiden van Data voor nnU-Net als we gaan trainen**

Als je beelden en labels nog in DICOM-format zijn, moeten deze worden geconverteerd naar NIfTI-format.

1. **Open PyCharm**  
   Laad PyCharm via het terminal commando:

module load pycharm/community-23.1.2

pycharm.sh

1. **Installeer dicom2nifti**  
   Installeer het dicom2nifti-pakket in je virtual environment (dit hoeft niet dezelfde te zijn als waar je nnU-Net in is geïnstalleerd):

pip install dicom2nifti

1. **Gebruik een conversiescript**  
   Maak een Python-script (zie conv\_im\_dicom2nifti.py en conv\_la\_dicom2nifti.py) voor het automatisch omzetten van DICOM naar NIfTI. Hierin zijn for-loops gebruikt om alle mappen met DICOM-afbeeldingen te converteren. Dit is een simpel voorbeeld waar de scripts op zijn gebaseerd:   
     
     
     
   NB noem dit script geen dicom2nifti, dan ,krijg je een error, convert\_dicom2nifti o.i.d kan wel gewoon.

import dicom2nifti

# Define the source directory containing DICOM files

dicom\_directory = '/home/rnga/kvanduuren/my-scratch/4d\_segmentation/Dicom/MB1\_06/'

# Define the output directory where NIfTI files will be saved

output\_directory = '/home/rnga/kvanduuren/my-scratch/4d\_segmentation/nifti\_split/MB1\_06/'

# Convert DICOM files to NIfTI format

dicom2nifti.convert\_directory(dicom\_directory, output\_directory, compression=True, reorient=True)

print("Conversion completed successfully.")

1. **Controleer de richting van de data**  
   Gebruik fsleyes om te controleren of de richting van de beelden en de labels correct is en met elkaar klopt:

fsleyes image.nii.gz label.nii.gz

1. **Corrigeer mask-values indien nodig**  
   Als de waarden in de masks niet kloppen (nnU-Net verwacht integers), pas deze aan met een script. Dit kan je checken in fsleyes (zie foto hieronder). Ik wilde graag 0 = background, 1 = lumen en 2 = trombus, maar in plaats daarvan waren het hele andere waardes. Voorbeeldcode is te vinden in assign\_values\_012.py en hieronder zie je een andere versie waar dat op gebaseerd is:

import os

import nibabel as nib

import numpy as np

# This will convert the values of the pixels of the masks, they were first really high, the will now be replaces

# by the (integers) 0 en 1

# Define the path to the directory containing the gzipped NIfTI files

input\_dir = '/home/rnga/kvanduuren/my-scratch/nnunet\_test/ground\_truth/'

# Iterate through the files in the directory

for file\_name in os.listdir(input\_dir):

if file\_name.endswith('.gz'):

# Define the full path to the NIfTI file

nifti\_path = os.path.join(input\_dir, file\_name)

# Load the NIfTI file

img = nib.load(nifti\_path)

data = img.get\_fdata()

# Cast the data to an integer type to avoid floating-point precision issues

data = data.astype(np.int64)

# Replace values greater than zero with one

data[data < 1000] = 0

data[data >= 1000] = 1

# Create a new NIfTI image with the modified data

new\_img = nib.Nifti1Image(data, img.affine, img.header)

# Save the new NIfTI image to a new file

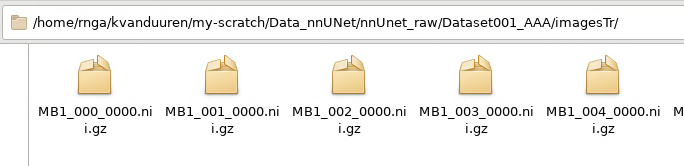
new\_nifti\_path = os.path.join(input\_dir, 'modified' + file\_name)

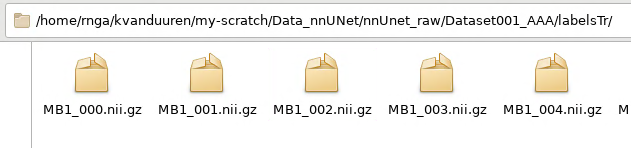
nib.save(new\_img, new\_nifti\_path)

print("Processing complete.")



1. **Converteer 4D MRI naar 3D**  
   Gebruik een script zoals 4D\_to\_3D.py om de eerste tijdstap van cine-MRI-data te isoleren, aangezien de segmentatie in mijn geval op die tijdstap was gebaseerd.
2. Als dit allemaal gelukt is kunnen we bijna beginnen met het trainen. Een voorbeeld hoe de inhoud van imageTr en LabelsTr eruit kan zien (let op de naamgeving van de files!!):





Voor de ImagesTr moet je de scans uit reconstruction/conv\_3D\_data/……nii.gz pakken en voor de labels in LabelsTr moet je de segmentaties uit combined\_mask(\no\_iliac)\.....permuted pakken.

**IV. Slurm voor nnU-Net Training**

1. **Interactive Slurm Sessies**  
   Om het nnU-Net te trainen kun je of een interactive session gebruiken of een slurm job indienen. Op de volgende sharepoint staat uitgelegd hoe dit precies werkt: [slurm (sharepoint.com)](https://amsterdamumc.sharepoint.com/sites/mip_wiki/SitePages/slurm.aspx).  
   Voor het trainen mijn nnU-Net (10 scans) heb ik alleen interactive sessions gebruikt, dit was genoeg om voor een beperkt aantal epochs te trainen, maar niet voor de default 1000 epochs. Als je een interactive slum sessie start moet je daarna wel weer even opnieuw je virtual environment activeren. Om een interactive slurm sessie te starten vul je het volgende in op je terminal.

salloc --job-name "interactive-shell" --partition=luna-gpu-short --cpus-per-task 8 --mem 40G --time 4:00:00 --gres=gpu:1g.10gb:1

Dit heeft uiteindelijk gewerkt maar was wel 16% efficiënt qua memory. Hij deed er met 50 epochs en 0 fold 2 uur en 20 minuten over.

Het kan zijn dat je best een tijdje moet wachten tot je een sessie toegewezen krijgt, vooral midden op de dag is het vaak best druk bezet. 8G is de standaard memory, maar voor het trainen had ik al snel (min.) 10G nodig.

**V. Preprocessing**

1. **Voer de Preprocessing Uit**  
   Gebruik de volgende command om preprocessing uit te voeren. Dit checkt ook of de naamgeving en structuur van de data correct is:

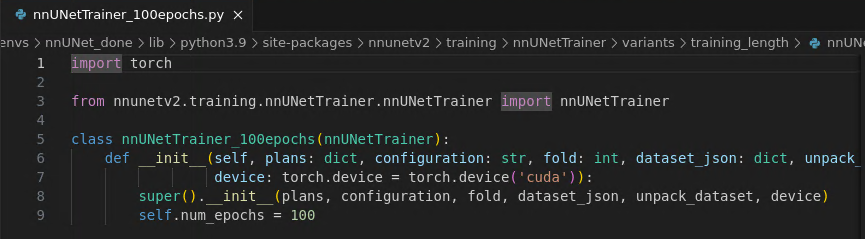
nnUNetv2\_plan\_and\_preprocess -d 001 --verify\_dataset\_integrity

**VI. Training**

1. **Pas epochs aan voor training**

Folder voor varianten op epochs: /home/rnga/eaalbregt/.conda/envs/nnUNet\_done/lib/python3.9/site-packages/nnunetv2/training/nnUNetTrainer/variants/training\_length/

Je moet dan nnUNetTrainer\_Xepochs.py openen. Je moet een nieuw bestand maken op dezelfde plek die noem je bijvoorbeeld nnUNetTrainer\_50epochs.py voor als je 50 epochs wil. Uit het eerste bestand haal je de twee eerste regels en dan de specifieke class die bij je aantal epochs hoort die je gekozen hebt. Je moet de naam van class veranderen met het aantal epochs en dan onderin in de class ook.



1. **Start de Training**  
   Begin met trainen van het model. Bijvoorbeeld met 50 epochs (zie [How I Use nnUNet for Medical Image Segmentation: A Comprehensive Guide - PYCAD](https://pycad.co/nnunet-for-medical-image-segmentation/)). Begin vooral niet met de default 1000 epochs want dat duurt waarschijnlijk lang en het is wel fijn om eerst even te zien hoe het werkt met minder epochs. De configurentie en aantal folds die je voor cross validatie het best kan gebruiken zijn afhandelijk van de data, ik heb het als volgt gedaan, maar experimenteer vooral. Dit heb ik gedaan:

nnUNetv2\_train 001 3d\_fullres 0 -tr nnUNetTrainer\_50epochs

Dit doe ik nu via de terminal met een scriptje zodat ik voor een langere periode kan trainen. De luna-gpu-long doet het alleen doet het alleen via zo’n scriptje en niet via een interactieve sessie. Je moet eerst in je virtual environment zitten en in het goede mapje zitten dan kun je het scriptje aanroepen. Het scriptje roep je zo aan: sbatch naam\_scriptje.sh

De scriptjes staan hier: /home/rnga/eaalbregt/my-scratch/imageReconstruction/.

Ik schrijf altijd even de slurm job id op zodat je de efficiëntie later kunt bekijken en ook goed weet wat je hebt getraind met welke fold.

**VII. Voorspellingen Maken**

1. **Gebruik het getrainde model om voorspellingen te doen**  
   Vul het pad in naar je input en output mappen:

nnUNetv2\_predict -i /path/to/input/images -o /path/to/output -d 001 -c 3d\_fullres -tr nnUNetTrainer\_50epochs -f 0

Voor mij:

nnUNetv2\_predict -i /home/rnga/kvanduuren/my-scratch/Data\_nnUNet/nnUnet\_raw/Dataset001\_AAA/imagesTs -o /home/rnga/kvanduuren/my-scratch/results/ -d 001 -c 3d\_fullres -tr nnUNetTrainer\_50epochs -f 0

Voor mij Eva:

nnUNetv2\_predict -i /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/nnUNet\_raw/Dataset012\_AAA/imagesTs/ -o /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/012\_0/ -d 012 -c 3d\_fullres -tr nnUNetTrainer\_350epochs -f 0

nnUNetv2\_predict -i /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/nnUNet\_raw/Dataset018\_AAA/imagesTs/ -o /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/018/ -d 018 -c 3d\_fullres -tr nnUNetTrainer\_350epochs -f 0

Om dit te laten werken heb je wel weer een slurmjob met gpu nodig.

1. Ensemble maken van de five-fold cross-validation die je gedaan hebt. Algemeen gezien doe je dat met:

nnUNetv2\_ensemble -i /path/to/predictions\_from\_folds/ -o /path/to/final\_predictions/ -t 018

Ik heb gebruikt:

nnUNetv2\_ensemble

Het is daarbij wel van belang dat je de predictions op z’n manier hebt gedaan dat alle data aanwezig is. Je moet er namelijk wel de .npz en de .pkl files erbij hebben. Als je de predicties doet. Dus je moet er –save\_probablities bij zetten.

Bijvoorbeeld:

nnUNetv2\_predict -i /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/nnUNet\_raw/Dataset019\_AAA/imagesTs/ -o /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/019/baseline/fold\_0/ -d 019 -c 3d\_fullres -tr nnUNetTrainer\_350epochs --save\_probabilities -f 0

Als je dat voor alle folds hebt gedaan kun je de ensemble maken.

nnUNetv2\_ensemble -i /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/019/baseline/fold\_0 /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/019/baseline/fold\_1 /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/019/baseline/fold\_2 /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/019/baseline/fold\_3 /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/019/baseline/fold\_4 -o /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/results/019/baseline/all/

**VIII. Evaluatie van het Model**

1. **Vergelijk voorspellingen met de ground-truth segmentaties**  
   Gebruik het volgende commando om evaluaties uit te voeren. Zorg ervoor dat je het juiste pad naar je dataset.json en plans.json toevoegt:

nnUNetv2\_evaluate\_folder /path/to/ground\_truth/ /path/to/predictions/ -djfile /path/to/dataset.json -pfile /path/to/plans.json

Voor mij:   
nnUNetv2\_evaluate\_folder /home/rnga/kvanduuren/my-scratch/nnunet\_tests/ground\_truth/ /home/rnga/kvanduuren/my-scratch/nnunet\_tests/predictions/ -djfile /home/rnga/kvanduuren/my-scratch/Data\_nnUNet/nnUNet\_results/Dataset001\_AAA/nnUNetTrainer\_50epochs\_\_nnUNetPlans\_\_3d\_fullres/dataset.json -pfile /home/rnga/kvanduuren/my-scratch/Data\_nnUNet/nnUNet\_results/Dataset001\_AAA/nnUNetTrainer\_50epochs\_\_nnUNetPlans\_\_3d\_fullres/plans.json

Voor mij Eva:

nnUNetv2\_evaluate\_folder /home/rnga/eaalbregt/my-scratch/imageReconstruction/nnUNet\_test/ground\_truth/ /home/rnga/eaalbregt/my-scratch/imageReconstruction/nnUNet\_test/predictions/ -djfile /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/nnUNet\_results/Dataset012\_AAA/nnUNetTrainer\_350epochs\_\_nnUNetPlans\_\_3d\_fullres/dataset.json -pfile /home/rnga/eaalbregt/my-scratch/imageReconstruction/Data\_nnUNet/nnUNet\_results/Dataset012\_AAA/nnUNetTrainer\_350epochs\_\_nnUNetPlans\_\_3d\_fullres/plans.json

De voorspelling komt in je mapje met de predictions te staan.

Waar is conda?

/opt/aumc-apps-eb/software/Anaconda3/2024.02-1/condabin/conda