**Inside Ubuntu terminal**

prabal@Prabal:~$ ssh prghosh@access.grid5000.fr

prghosh@access-north:~$ ssh sophia

Creation of virtual environnement

Module load cuda/11.8

Module load conda

Conda create -n virtual\_prabal python=3.10.0

Conda activate virtual\_prabal

**Installation of pytorch with gpu cuda 11.8- check these 2 methods inside your virtual environment**

pip install torch==2.3.1 torchvision==0.18.1 torchaudio==2.3.1 --index-url <https://download.pytorch.org/whl/cu118>

*From <*[*https://pytorch.org/get-started/previous-versions/*](https://pytorch.org/get-started/previous-versions/)*>*

conda install pytorch==2.3.1 torchvision==0.18.1 torchaudio==2.3.1 pytorch-cuda=12.1 -c pytorch -c nvidia

*From <*[*https://pytorch.org/get-started/previous-versions/*](https://pytorch.org/get-started/previous-versions/)*>*

pip install TotalSegmentator

Now the setup is done. There might be some errors, which you will have to fix by looking at the error messages during code execution..

---------------------------------------------------------------------------------------------------

**This is the python script**

**Vim cluster\_test\_0005.py**

import nibabel as nib

from pathlib import Path

from totalsegmentator.python\_api import totalsegmentator

# import numpy as np

# import os

# import matplotlib.pyplot as plt

input\_path\_1 = Path("/home/prghosh/prabal\_ghosh/Inria\_Medical\_Imageing\_Internship\_prabal/s0005/mri.nii.gz")

output\_path\_1 = Path("/home/prghosh/prabal\_ghosh/Inria\_Medical\_Imageing\_Internship\_prabal/s0005/segmentations\_test\_2")

img = nib.load(input\_path\_1).get\_fdata()

print(img.shape)

print(f"\*\*\*\*\*The .nii files are stored in memory as numpy's: {type(img)}.\*\*\*\*\*")

if \_\_name\_\_ == "\_\_main\_\_":

    # Segment the first MRI image

    print(f"\*\*\*\*\*\*Segmenting {input\_path\_1}\*\*\*\*\*\*\*")

    # totalsegmentator(input=input\_path\_1, output=output\_path\_1,device='gpu', task="total\_mr", roi\_subset= ["lung\_left", "lung\_right"])

    totalsegmentator(input=input\_path\_1, output=output\_path\_1,device='gpu', task="total\_mr")

    print(f"\*\*\*\*\*\*\*\*\*\*\*Segmentation completed for {input\_path\_1}. Results saved to {output\_path\_1}.\*\*\*\*\*\*\*\*\*\*")

    print("thanks for using TotalSegmentator! prabal\_ghosh")

============================================================================

1. **Method1:**

**This is the shell script**

vim prabal\_test\_shell.sh

Inside this .sh file write the following code

#!/bin/bash

#OAR -q production

#OAR -l host=1/gpu=1

#OAR -l walltime=00:30:00

#OAR -p gpu\_count > 0

#OAR -O OAR\_%jobid%.out

#OAR -E OAR\_%jobid%.err

# display some information about attributed resources

echo "=== Host and GPU Info ==="

hostname

nvidia-smi

nvcc --version

echo "=== Loading environment ==="

module load conda

module load cuda/11.8

conda activate mri\_2025\_4

echo "=== Checking PyTorch GPU Availability ==="

python3 -c "import torch; print('CUDA available:', torch.cuda.is\_available()); print('Device:', torch.cuda.get\_device\_name(0))"

echo "=== Starting TotalSegmentator Job ==="

cd /home/prghosh/prabal\_ghosh/Inria\_Medical\_Imageing\_Internship\_prabal

# Run your script

python cluster\_test\_0005.py

echo "=== Done ==="

conda deactivate

Now to run the shell script use the Following commands

chmod +x prabal\_test\_shell.sh # is to check that its exécutable or not

oarsub -S ./prabal\_test\_shell.sh # to run the script

oarstat -u # Check status

cat OAR\_${OAR\_JOB\_ID}.out # View output

cat OAR\_${OAR\_JOB\_ID}.err # View errors

1. **Method2:**

If you want to run it interactively, use the following commands — you don't need to write any shell script file.

oarsub -I -q production -l gpu=1,walltime=0:05:00

module load conda

module load cuda/11.8

conda activate mri\_2025\_4

cd ~/prabal\_ghosh/Inria\_Medical\_Imageing\_Internship\_prabal

python cluster\_test\_0005.py