

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/>>

```
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/>>

```
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_total_mr =
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/segmentations_total_mr")
output_path_body_mr =
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/segmentations_body_mr")
output_path_vertebrae_mr =
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/segmentations_vertebrae_mr")
output_path_liver_mr =
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/segmentations_liver_mr")
output_path_appendicular_bones_mr =
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/segmentations_appendicular_bones_mr")
output_path_tissue_types_mr =
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/segmentations_tissue_types_mr")
output_path_thigh_shoulder_muscles_mr =
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/segmentations_thigh_shoulder_muscles_mr")
output_path_face_mr =
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/segmentations_face_mr")

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__":
    print("*****Total MRI Segmentation*****")
    # 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_total_mr,device= 'gpu', task="total_mr")

    print(f"*****Segmentation completed for {input_path_1}. Results saved to {output_path_total_mr}.*****")

    print("*****body mri segmentation*****")
    # 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_body_mr,device= 'gpu', task="body_mr")

    print(f"*****Segmentation completed for {input_path_1}. Results saved to {output_path_body_mr}.*****")

    print("*****vertebrae mr segmentation*****")
    # 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_vertebrae_mr,device= 'gpu', task="vertebrae_mr")

    print(f"*****Segmentation completed for {input_path_1}. Results saved to {output_path_vertebrae_mr}.*****")

    print("*****liver segmentations mr*****")
    # 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_liver_mr,device= 'gpu', task="liver_segments_mr")

print(f"*****Segmentation completed for {input_path_1}. Results saved to {output_path_liver_mr}.*****")

print("*****appendicular bones mr*****")
# 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_appendicular_bones_mr,device= 'gpu',
task="appendicular_bones_mr")

print(f"*****Segmentation completed for {input_path_1}. Results saved to {output_path_appendicular_bones_mr}.
*****")

print("*****tissue types mr*****")
# 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_tissue_types_mr,device= 'gpu', task="tissue_types_mr")

print(f"*****Segmentation completed for {input_path_1}. Results saved to {output_path_tissue_types_mr}.
*****")

print("*****thigh shoulder muscles mr*****")
# 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_thigh_shoulder_muscles_mr,device= 'gpu',
task="thigh_shoulder_muscles_mr")

print(f"*****Segmentation completed for {input_path_1}. Results saved to {output_path_thigh_shoulder_muscles_mr}.
*****")

print("*****face mr*****")

# 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_face_mr,device= 'gpu', task="face_mr")

print(f"*****Segmentation completed for {input_path_1}. Results saved to {output_path_face_mr}.*****")
print("*****all segmentations completed*****")

=====

```

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 executable 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
```