Small note on installing the cuda for usage in my project file:

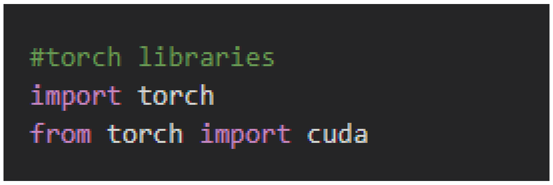
Download the cuda toolkit through nvidia’s website:
https://developer.nvidia.com/cuda-downloads

Run the following command in the terminal:

pip install torch==1.13.1+cu117 torchvision==0.14.1+cu117 --extra-index-url https://download.pytorch.org/whl/cu117

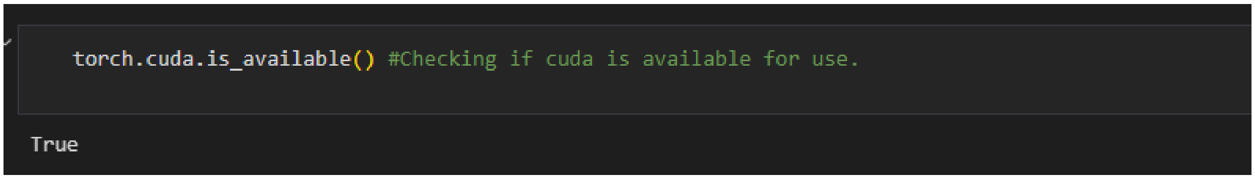
- Installs the torch package through pip with cuda implementation.
- The command above could appear on the document as two separate lines. However, It should be treated as one line within the terminal.
- Before running the command, Make sure to uninstall torch (As attempting to install the torch package WITH cuda support while having the regular torch package installed could cause an error saying that torch is already installed)
- There are other versions of cuda available for torch to be downloaded, As well as other device support variants. Such as; CPU only (the variant that is installed by default when downloading just torch through the terminal if none is specified otherwise).
-

Afterwards installing everything, You can import torch as you would regularly as;



```
#torch libraries
import torch
from torch import cuda
```

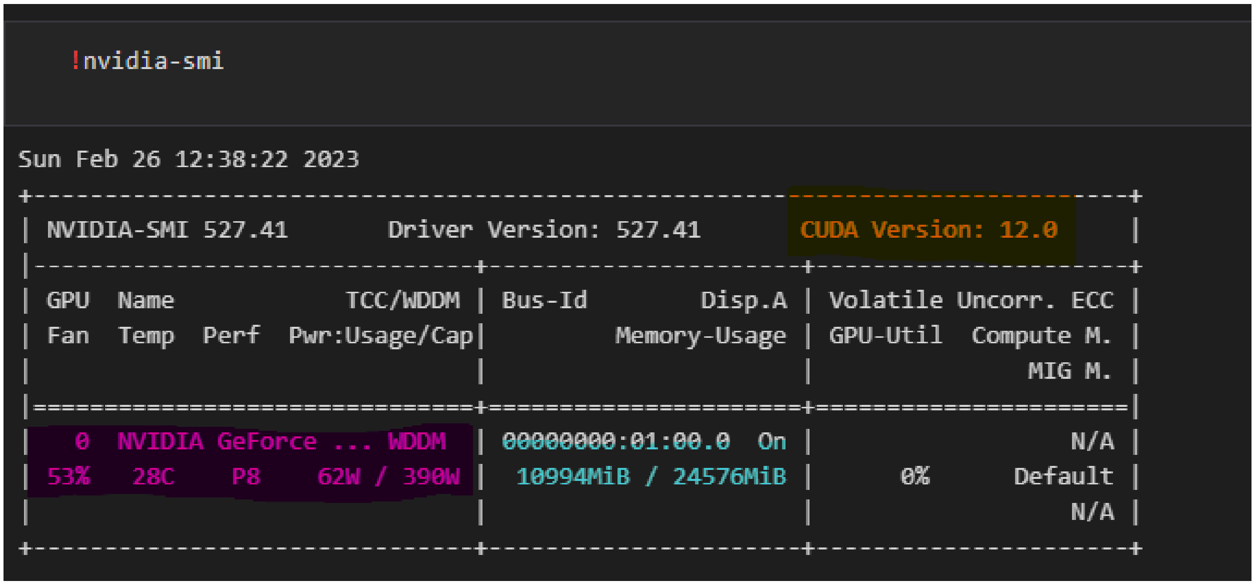
Then, Run the following commands and for each you should expect the following output:



```
torch.cuda.is_available() #Checking if cuda is available for use.

True
```

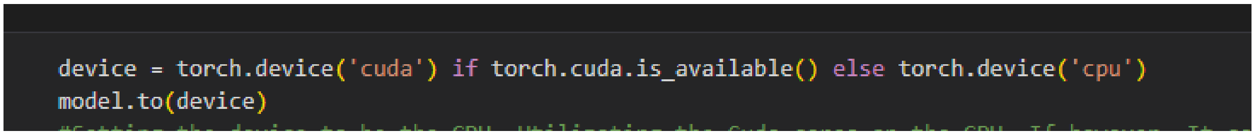
^ Returns a bool value if cuda can be used (the GPU in our case)



```
!nvidia-smi
```

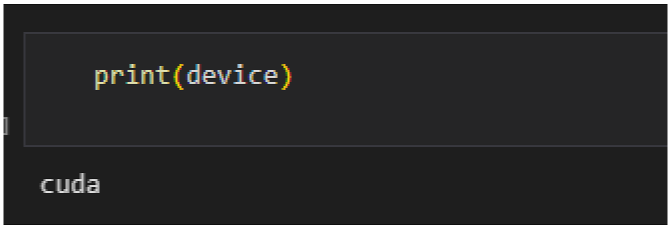
```
Sun Feb 26 12:38:22 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 527.41       Driver Version: 527.41       CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name            TCC/WDDM | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA GeForce ... WDDM  | 00000000:01:00.0  On |                  N/A |
| 53%   28C    P8    62W / 390W |  10994MiB / 24576MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

Orange - CUDA version that is installed on the GPU.
Purple - Name of the GPU that is detected and can be used.
Cyan / Blue (? I don’t know) - VRAM that can be used. As well as currently used VRAM out of the total amount of VRAM available. (NOTE THAT THIS WILL ALWAYS BE HIGH WHEN RUNNING THE “ !nvidia-smi “ COMMAND)



```
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
model.to(device)
#Setting the device to be the GPU. Utilizing the Cuda cores on the GPU. If however, It ca
```

Command for setting the device of which the model will run on as well as putting the model on the defined device.
Note: We add a statement for a secondary option in case the user’s GPU can not be used, Is not CUDA capable or any other problem that prevents from using the GPU.



```
print(device)
```

```
cuda
```

Simple print that gives a string of what device the model is running on.
Cuda : The device is set to be the GPU of the machine.
CPU : The device is set to be the CPU of the machine.

Important note: Overuse of the GPU in one run can cause some issue with VRAM. In the case of being told that there is not enough VRAM available, Run the following command and Restart the kernel / Restart the application the program is running on. Afterwards, Wait a few seconds (about 10 seconds) and your VRAM should be flushed.

```
torch.cuda.empty_cache()
```

^ The command flushes the cache on the GPU which clears some VRAM on the GPU but will not clear all the VRAM. For a complete flush, Please read the statement above marked in yellow.