

Matt Morrow

CIS5898

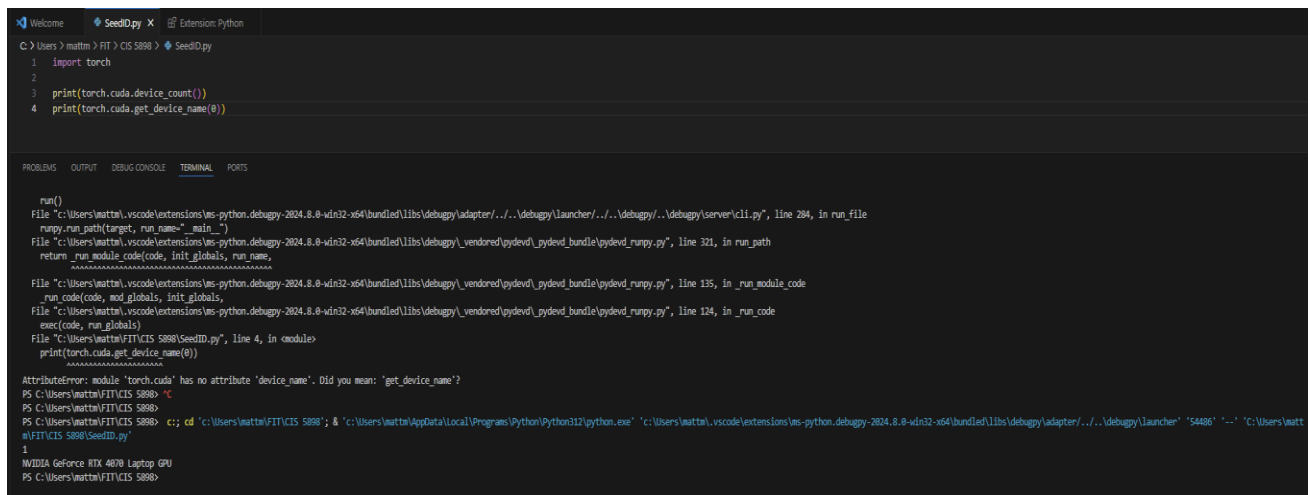
First Progress Report

09/14 – The project proposal has been submitted.

09/15 - The initial steps have been taken in the development of a SeedID. The program now connects to an Azure Blob Storage, uploads images, and processes them using PyTorch; with CUDA support for GPU acceleration.

CUDA 12.4: This version of CUDA leverages NVIDIA GPUs for high-performance computing. This will speed up deep learning tasks that involve processing large datasets or performing neural network computations.

PyTorch: Installed as the primary deep learning library. PyTorch provides seamless integration with CUDA, enabling the use of GPUs for faster image processing and model training.



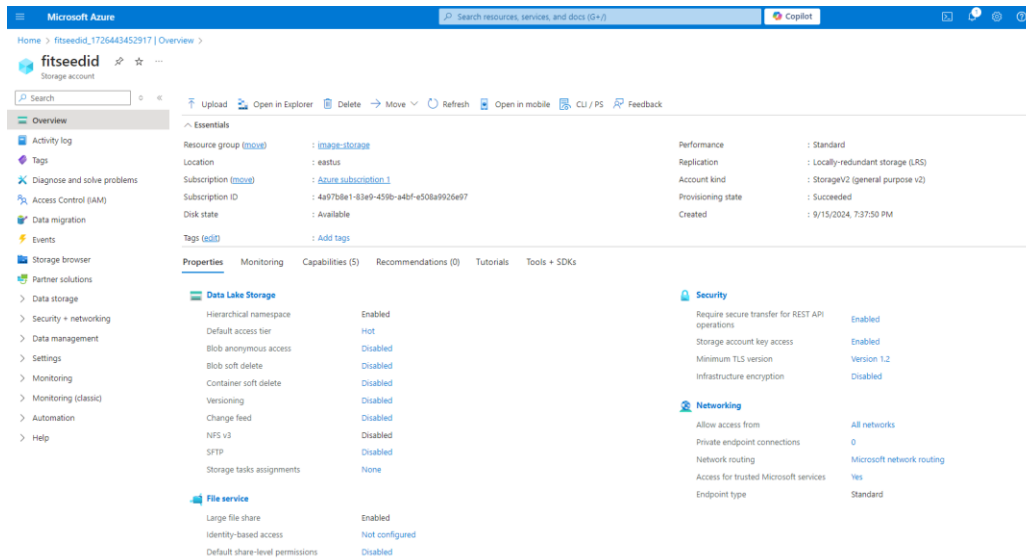
```

Welcome  SeedID.py X  Extension: Python
C:\Users\mattm> cd "C:\Users\mattm\Documents\CIS 5898" & cd SeedID.py
1  import torch
2
3  print(torch.cuda.device_count())
4  print(torch.cuda.get_device_name(0))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

run()
File "c:\Users\mattm\.vscode\extensions\ms-python.debugpy-2024.8.8-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher\..\..\debugpy\..\debugpy\server\cli.py", line 284, in run_file
  runpy.run_path(target, run_name="__main__")
File "c:\Users\mattm\.vscode\extensions\ms-python.debugpy-2024.8.8-win32-x64\bundle\libs\debugpy\_\vendored\pydevd\pydevd_bundle\pydevd_runpy.py", line 321, in run_path
  return _run_module_code(code, init_globals, run_name,
  ~~~~~
File "c:\Users\mattm\.vscode\extensions\ms-python.debugpy-2024.8.8-win32-x64\bundle\libs\debugpy\_\vendored\pydevd\pydevd_bundle\pydevd_runpy.py", line 135, in _run_module_code
  _run_code(code, mod_globals, init_globals,
  ~~~~~
File "c:\Users\mattm\.vscode\extensions\ms-python.debugpy-2024.8.8-win32-x64\bundle\libs\debugpy\_\vendored\pydevd\pydevd_bundle\pydevd_runpy.py", line 124, in _run_code
  exec(code, run_globals)
File "c:\Users\mattm\Documents\CIS 5898\SeedID.py", line 4, in <module>
  print(torch.cuda.get_device_name(0))
  ~~~~~
AttributeError: module 'torch.cuda' has no attribute 'device_name'. Did you mean: 'get_device_name'?
PS C:\Users\mattm\Documents\CIS 5898> cd "c:\Users\mattm\Documents\CIS 5898" & & "c:\Users\mattm\AppData\Local\Programs\Python\Python312\python.exe" "c:\Users\mattm\.vscode\extensions\ms-python.debugpy-2024.8.8-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher" "54486" "-." "c:\Users\mattm\Documents\CIS 5898\SeedID.py"
1
NVIDIA GeForce RTX 4070 Laptop GPU
PS C:\Users\mattm\Documents\CIS 5898>
```

09/16 - Azure blob storage has been partitioned. The container "seedimages" was created within the Azure Blob Storage account "fitseedid" to store the images. The Azure Blob service client was set up using the connection string to communicate with the container.



09/18 - The Program can now communicate with the Azure database and add an image from a local file path.

```

1
NVIDIA GeForce RTX 4070 Laptop GPU
Deleting blob: Corn.jpg
All blobs have been deleted.
Uploaded Corn.jpg to Azure Blob Storage.
Image tensor shape: torch.Size([3, 256, 256])
PS C:\Users\mattm\FIT\CIS 5898>

```

The core function at this point in development is:

```

defupload_image(self, local_file_path):

    upload_image_to_azure(self.connection_string, self.container_name, local_file_path)

```

Which passes the connection string, container name, and file path to a separate class that uses a built in function, `blob_service_client.get_container_client()`, to send the blob to the container.