Nerf-Pytorch

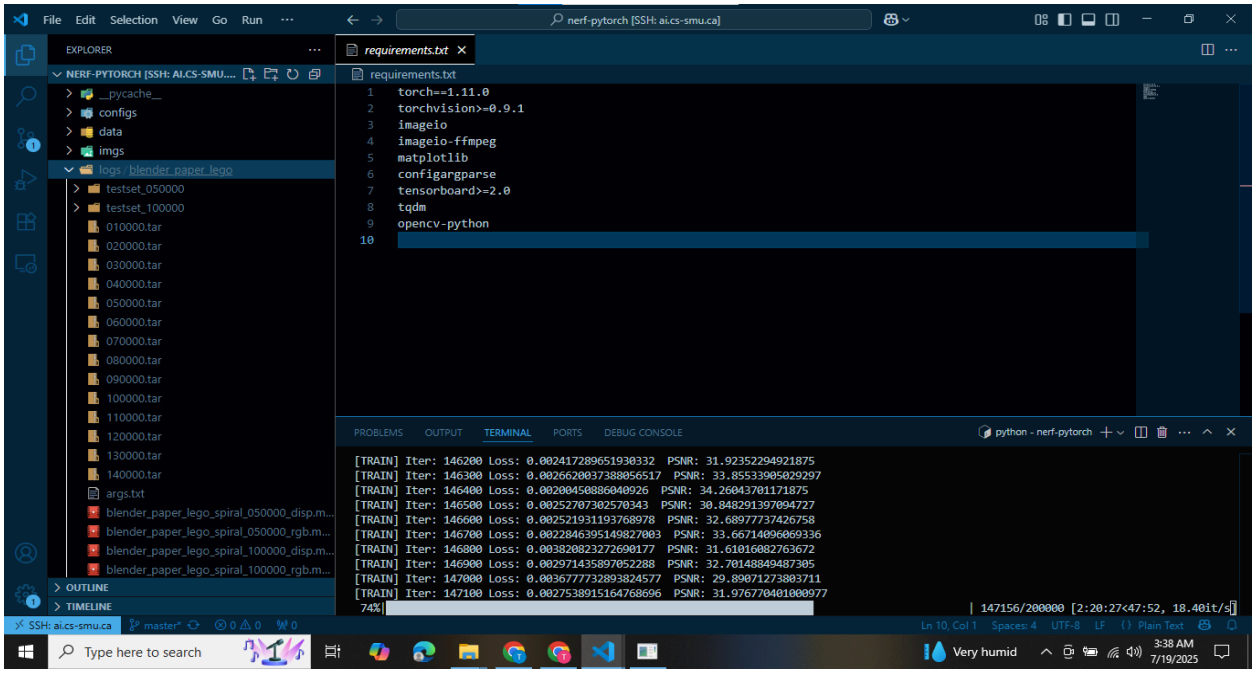
Task 1: Setting up the env and reconstructing lego

The setup was completed with some conflict in dependencies (torch and numpy), the lego was reconstructed

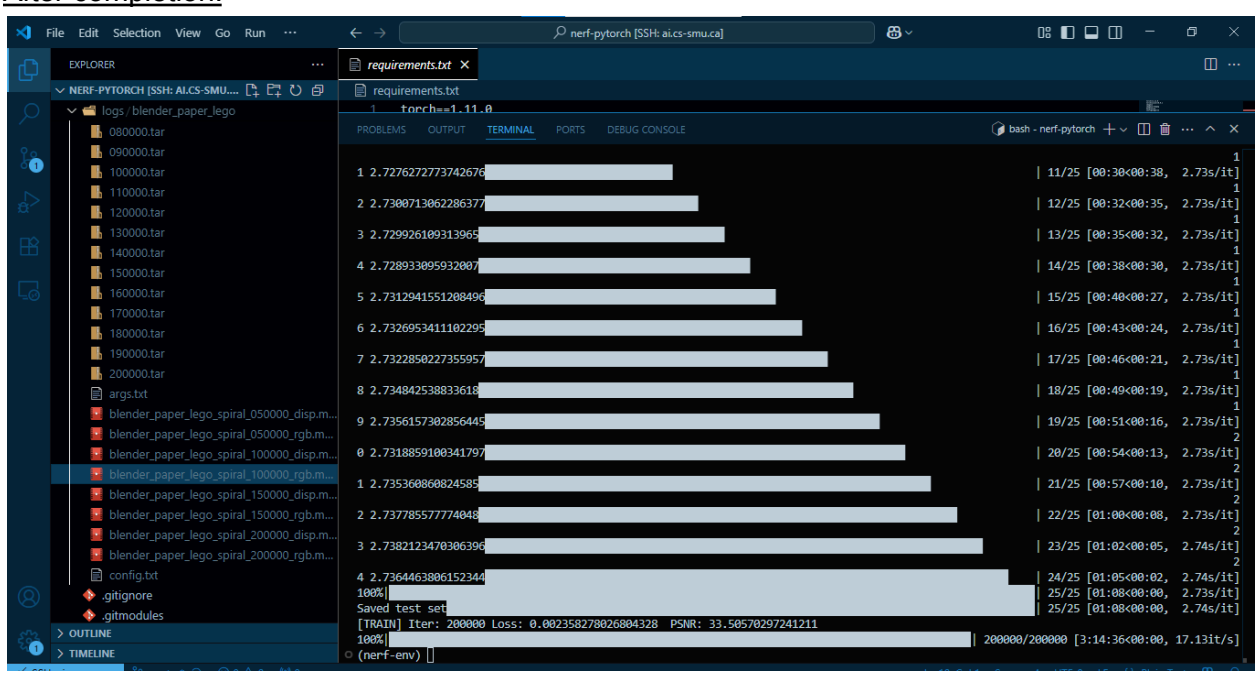
The video output is clear as shown in the repo and paper.

This folder consists of the videos generated at different iterations: Task 1

Implementation:



After completion:



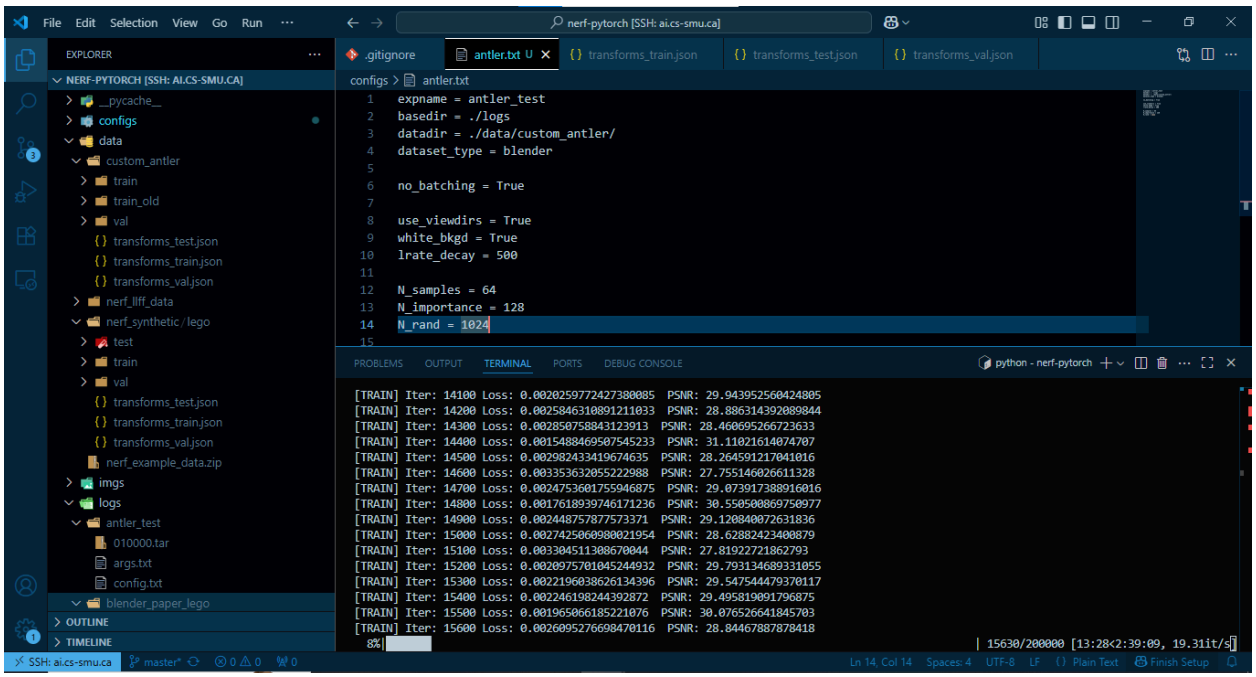
Talha Hussain Khan talhahkhan.thk@gmail.com

Task 2: Implementing on Custom dataset

First I extracted the frames and made transform.json from the .stl file.
I wanted to render it via blender but my system did not support it so I used an alternative approach, which I believe did not generate the transform.json. Hence the output of video is not optimal.

The script (and config files) can be found here: https://github.com/Talha-Here/render_frames
The data can be found here: antler_data2

Implementation:

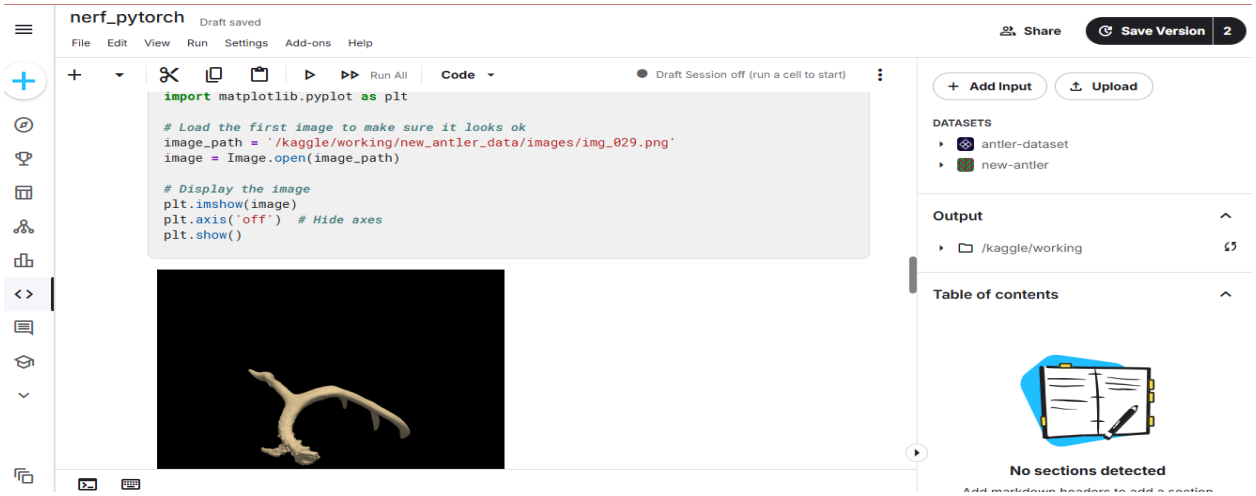


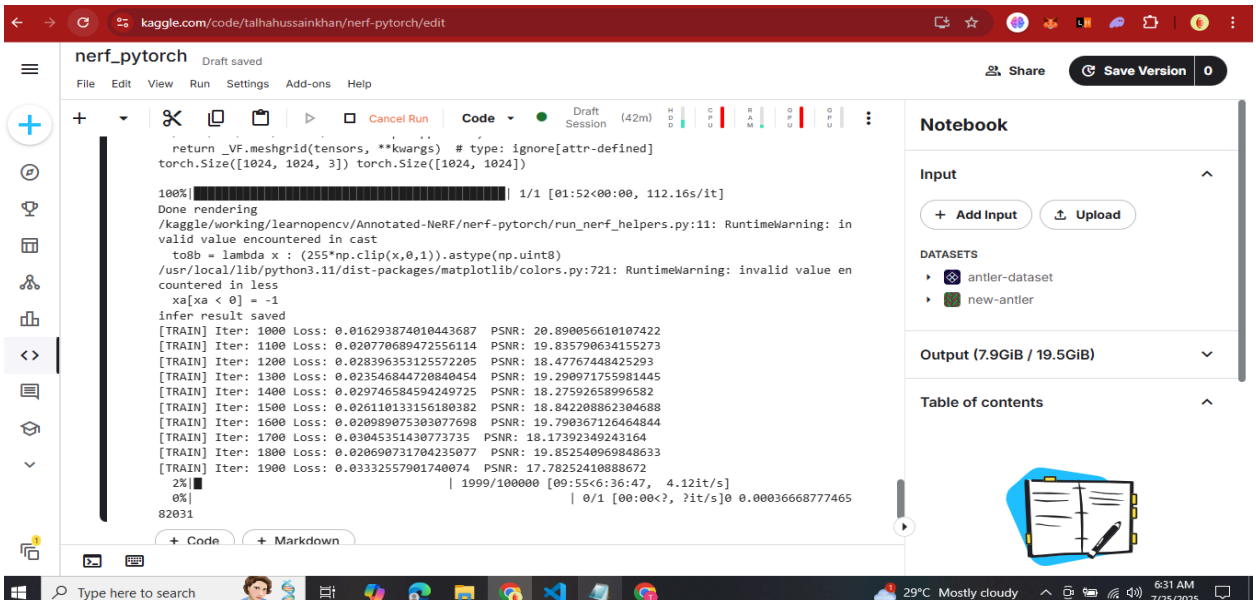
However the output is not as required (I changed white_bkgd to False, but could not get the correct results). I believe using blender would have given the correct results.
Results: type_blender

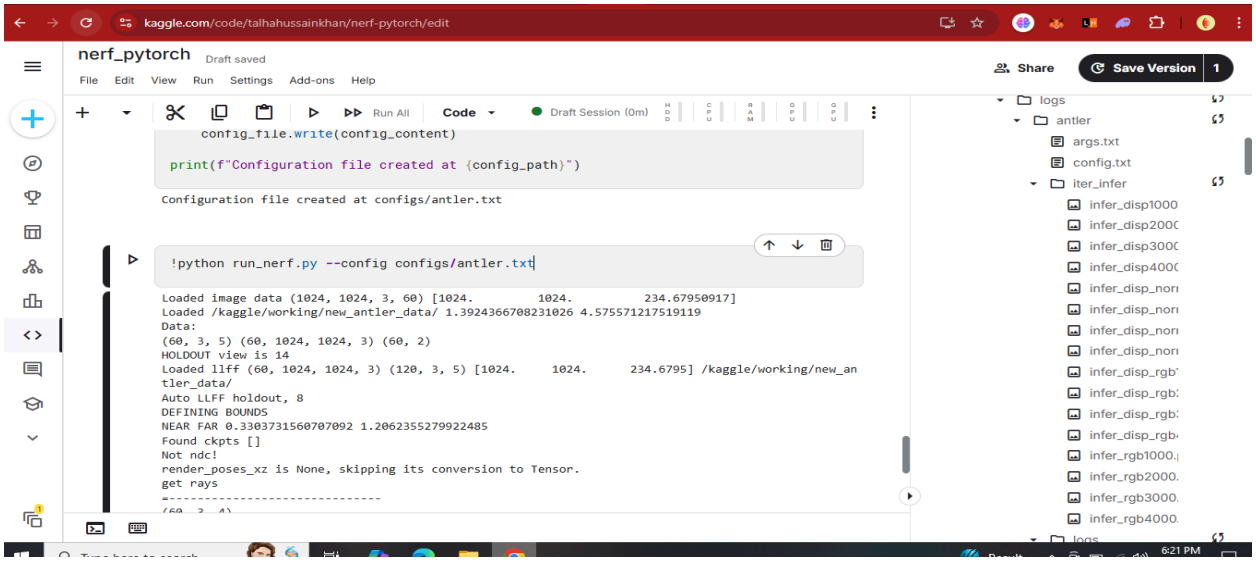
Then I tried to use the llff type, which needs colmap. Here I used Kaggle (GPU T4 x2) because I could not run it on my system.
This implementation failed multiple times, as colmap kept on crashing but after changing configuration it worked.

Implementation:

Notebook: nerf-pytorch_2.ipynb







Talha Hussain Khan talhahkhan.thk@gmail.com

The videos could not be compiled as I reached the weekly limit of GPU provided by Kaggle. Here are the png that were generated in between the process: 📷 type_llff

References:

1. A PyTorch implementation of NeRF (Neural Radiance Fields) that reproduces the results.
2. https://github.com/bryceschultz/nerf-pytorch
3. The Annotated NeRF – Training on Custom Dataset from Scratch in Pytorch