Progress report 8

Jaison Lewis

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

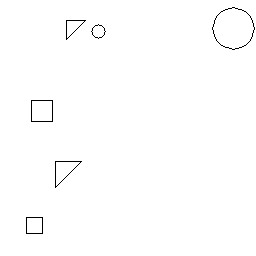
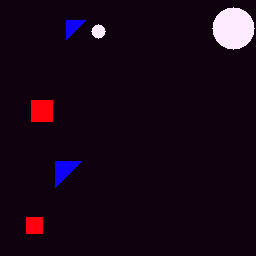
1: Scripts and programs: <https://github.com/jaisonlewis/HPC-scripts>

a: Script: Initialisation of interactive Slurm:   
Wrote a small script that initialises Slurm in interactive mode.

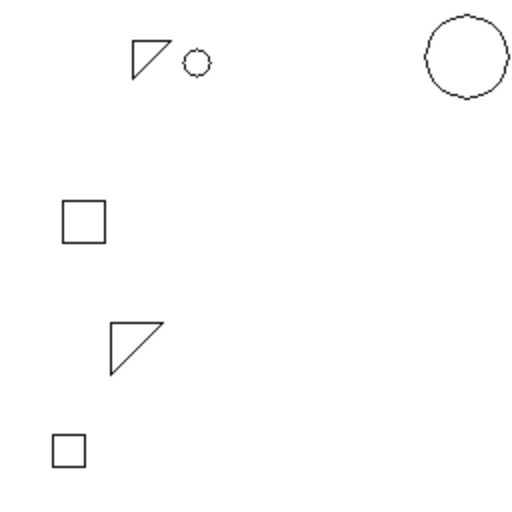
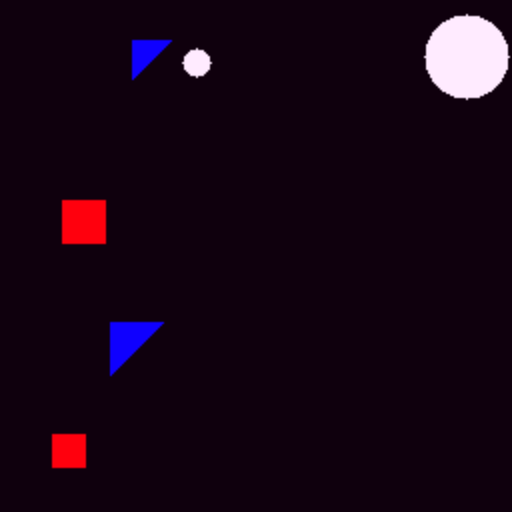
b: Script to set env and run program:   
A small script that loads the Anaconda module, then activates the environment and then runs the program.

c: Sbatch script to use GPU for running segmentation program:

2: Modified version of synthesiser program: [Syn-ble-resize/syn-ble-resize.py at KidneyKaggleChallenge · jaisonlewis/Syn-ble-resize (github.com)](https://github.com/jaisonlewis/Syn-ble-resize/blob/KidneyKaggleChallenge/syn-ble-resize.py)  
I had to modify the program because there was some error in creating the shapes at higher image sizes. It was taking too long to figure out, so I modified the program to create the necessary number of files, resized them and modified the CSV file to match. It still makes the 256x256 images but can now resize them to any size specified. Will work on the original program to figure out what is going wrong, but this is a stop-gap solution till that happens.

Original 256:  


Resized to 512:



Partial coordinates for the image:   
Original: Image\_ID,Shape,Coordinates

0,triangle,"[(66, 20), (85, 20), (66, 39)]"

0,square,"[(31, 100), (52, 100), (52, 121), (31, 121)]"

0,triangle,"[(55, 161), (81, 161), (55, 187)]"

0,square,"[(26, 217), (42, 217), (42, 233), (26, 233)]"…

Modified:   
Image\_ID,Shape,Coordinates

0,triangle,"[(132, 40), (170, 40), (132, 78)]"

0,square,"[(62, 200), (104, 200), (104, 242), (62, 242)]"

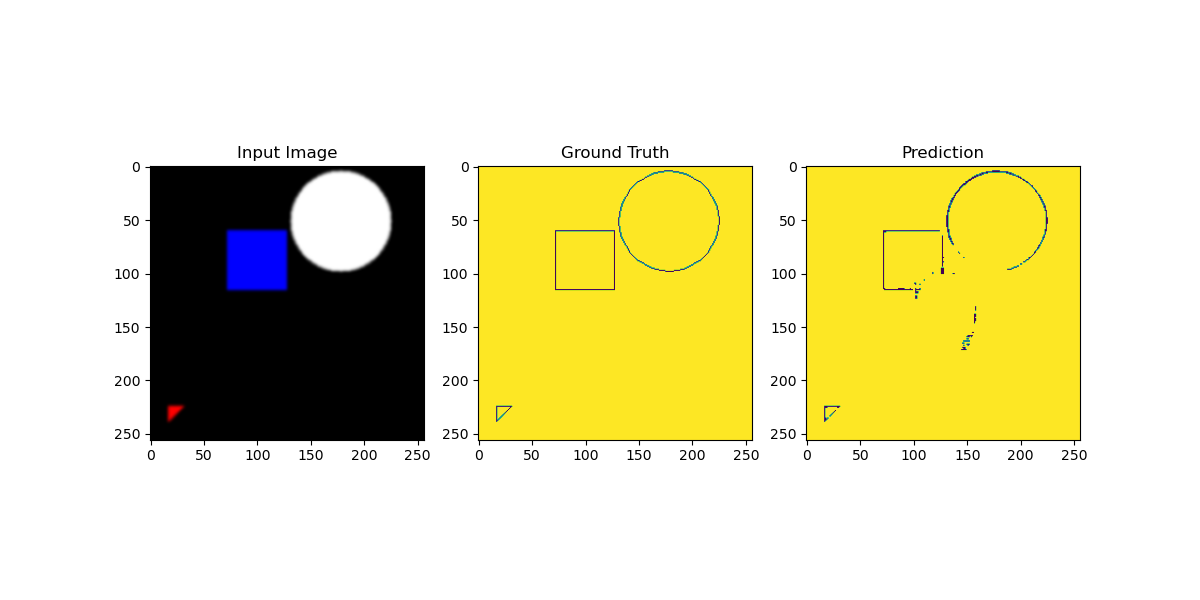
0,triangle,"[(110, 322), (162, 322), (110, 374)]"

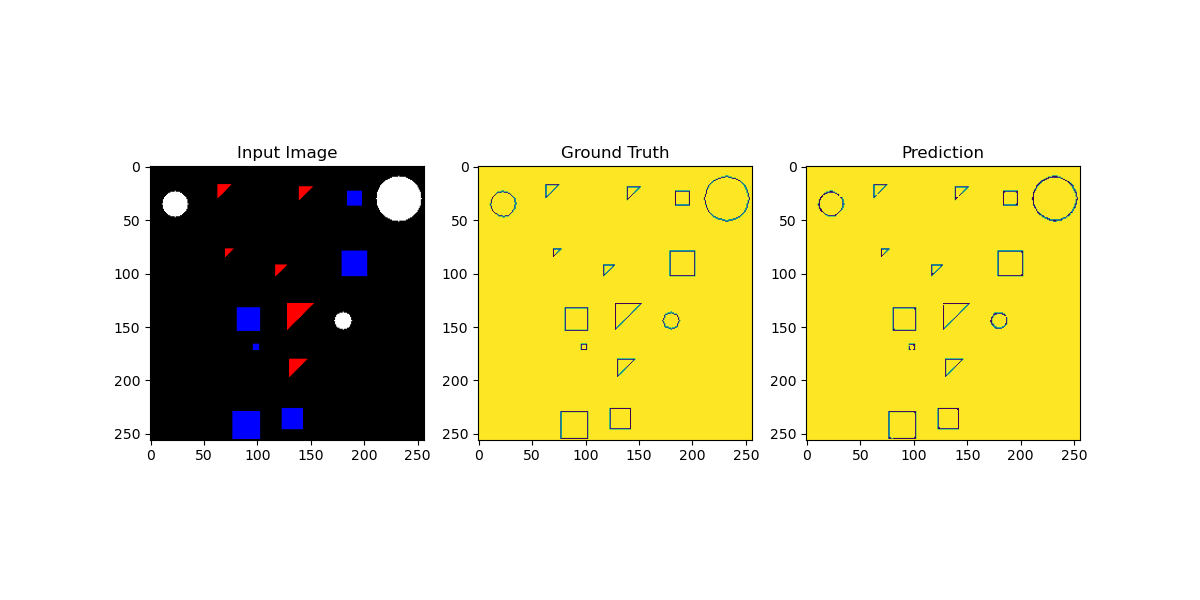
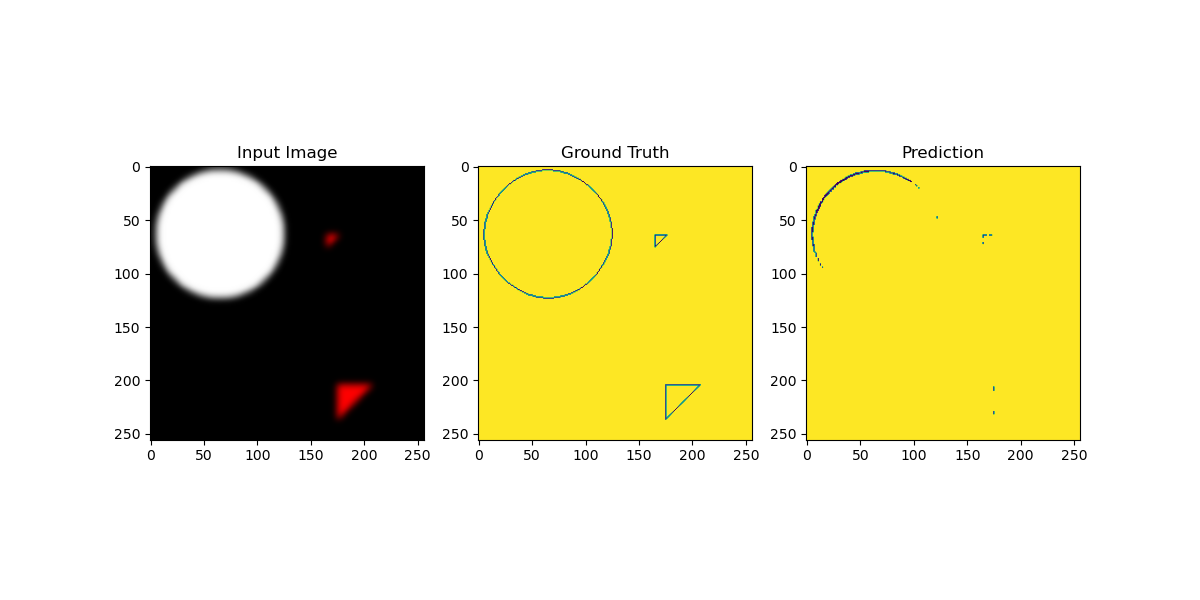
0,square,"[(52, 434), (84, 434), (84, 466), (52, 466)]"…

Tested with 1024 and 128 sizes as well.

Patched the resizing problem with the image generator by resizing the images after generation. It isn’t the right fix to the problem, but just a workaround. Will fix the core issue a little later.  
The modified Synthesizer program can be found here: <https://github.com/jaisonlewis/Syn-ble-resize/blob/KidneyKaggleChallenge/syn-ble-resize.py>  
Wrote a few scripts to work with the HPC. They are available on my github: <https://github.com/jaisonlewis/HPC-scripts>  
Getting the environment right for the tensorflow version of my script in Python 3.9.0 is proving challenging. Even after replicating the environment on my PC, the program refuses to work as desired. Hence, I decided to go the Pytorch route.

I have a working version of the Pytorch Segmentation model. <https://github.com/jaisonlewis/pysynseg>

Here are some samples from a 3 epoch test:  




The program was not showing me my graphs and comparisons when run in Sbatch. So, I tried to use Wandb on Narges's suggestion. Created a desired environment for it which is replicable.

However, the env was proving a little tricky to replicate in the HPC env. Narges suggested ways in which I could use the HPC space without installing things using pip install. This worked.

WanDB is working. However, this doesn’t solve my comparison chart problem.

