Journal Report 3

9/14/19-9/23/19
Avyuk Dixit
Computer Systems Research Lab
Period 4, White

Daily Log

Monday September 16

Profiled my code to attempt to understand what was taking up so much memory. Turned it was pickle that was using insane amounts of memory and all of the video processing was not really part of the problem. A little more research revealed that pickle is really bad at working with large amounts of data. Knowing this, I decided it would make sense to switch to working with JSON files or cPickle instead, as those are much more lightweight.

Tuesday September 17

I was absent on Tuesday because I was sick.

Thursday September 19

Did a little bit more research on cPickle vs JSON vs pickle just to make sure I was making the right decision. Learned about the JSON library in python and how to export objects and save them as file. Modified my code to use JSON and resolved new errors that came up as a result of modification.
My code crashed on my laptop even after switching to JSON, though it crashed much later than when I was using Pickle. This was surprising considering I had really thought Pickle was the root of the problem. I spent some time trying to figure out what alternative causes may have existed for my code overloading my memory, but couldn't come up with any. Switched to cPickle just to see if that would work, but didn't have time to test my code at the end.

Timeline

Date	Goal	Met
Today minus 2 weeks	Settle on a dataset to use for the project.	Yes, received access to the NTU CCTV-Fights Dataset
Today minus 1 week	Run pre-processing code to decompose surveillance feed into individual frames marked as violent or non-violent through entire dataset.	A number of memory related errors made it so that I was not able to meet this goal.
Today	Review OpenPose code and determine best statistical representation for poses.	Was not able to achieve this - still focused on memory issues from last week.
Today plus 1 week	Run OpenPose code through dataset of frames marked as violent or non-violent.	Probably need to push this back a little bit now.
Today plus 2 weeks	Write code that loads new dataset, preprocesses it, and gets it ready as input data for the neural network	

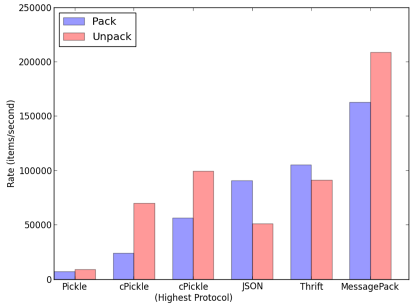
Reflection

Well I thought I had the root of my memory errors figured out, but I guess not. I'm going to keep trying to work with JSON and cPickle for at least another class period, because I still feel I can get them to work and have my code running on my computer. I'll also try and run everything remotely on machines like zoidberg and hpc12 again to see if that helps.

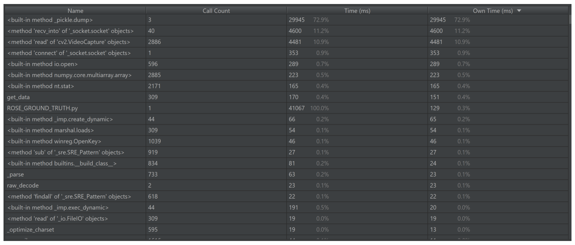
We just learned about MPI in parallel so maybe, worst case, I can try and run my code parallelly and that might help. Not exactly sure how that would work though – would I need to split videos into smaller parts and have each part be run in parallel? Hopefully it won't come to that so soon.

This is a really important thing to figure out because if I don't, then the rest of my project will probably be plagued by memory errors as well.

Here's a graphic that compares pickle, cPickle, JSON, and other methods of exporting stuff.



Here's a screenshot of results from my code profiling.



Name	Call Count	Time (ms)	Own Time (ms) ▼
<built-in method _pickle.dump>	3	29945 72.9%	29945 72.9%
<method 'recv_into' of '_socket.socket' objects>	40	4600 11.2%	4600 11.2%
<method 'read' of 'cv2.VideoCapture' objects>	2886	4481 10.9%	4481 10.9%
<method 'connect' of '_socket.socket' objects>	1	353 0.9%	353 0.9%
<built-in method io.open>	596	289 0.7%	289 0.7%
<built-in method numpy.core.multiarray.array>	2885	223 0.5%	223 0.5%
<built-in method nt.stat>	2171	165 0.4%	165 0.4%
get_data	309	170 0.4%	151 0.4%
ROSE_GROUND_TRUTH.py	1	41067 100.0%	129 0.3%
<built-in method _imp.create_dynamic>	44	66 0.2%	65 0.2%
<built-in method marshal.loads>	309	54 0.1%	54 0.1%
<built-in method winreg.OpenKey>	1039	46 0.1%	46 0.1%
<method 'sub' of '_sre.SRE_Pattern' objects>	919	27 0.1%	27 0.1%
<built-in method builtins.__build_class__>	834	81 0.2%	24 0.1%
_parse	733	63 0.2%	23 0.1%
raw_decode	2	23 0.1%	23 0.1%
<method 'findall' of '_sre.SRE_Pattern' objects>	618	22 0.1%	22 0.1%
<built-in method _imp.exec_dynamic>	44	191 0.5%	20 0.0%
<method 'read' of '_io.FileIO' objects>	309	19 0.0%	19 0.0%
_optimize_charset	595	19 0.0%	13 0.0%