**Intestinal Worms Fall 2023 Final Technical Report**

Global Solutions VIP

Team Members:

Nathaly Jose-Maria

Advisors:

Dr. Kelsey Kubelick, PhD

**Executive Summary**

Our team’s objective is to eliminate the morbidity of soil-transmitted helminths (STH) in under-served global communities by improving diagnoses in these communities. Given the environmental context of the disease, we seek to reduce the technical prowess necessary for utilizing many modern medical devices by replacing these devices with more simplified yet equally effective approaches. These approaches must be able to detect the existence, species, and concentration of intestinal worm eggs with adequate accuracy and efficiency ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); [Children Without Worms, 2021b](https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086)). This semester, the team was reduced to a single contributor given the termination of the VIP group. Despite this limitation, the individual worked to localize the RAM-intensive full object detection process the team sought to execute in previous semesters. This meant extensively researching locally available high-performance computing options and led the team to working with Georgia Tech’s own Partnership for an Advanced Computing Environment (PACE) service. The individual had to reimplement the team’s previous work in a different manner for successful integration into the service. Given the various complications experienced and limited resources, the individual had to reevaluate the localization process.

**Introduction**

Our project seeks to mitigate the impact of intestinal worms in areas with endemic soil transmitted helminth (STH) infections ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); [Children Without Worms, 2021b](https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086)). STH infections pose particular risks to children and pregnant mothers and affects more than one billion people globally ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); [Children Without Worms, 2021b](https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086)). The World Health Organization considers STH a widespread community issue given that more than 2% of the population has a moderate or severe STH infection ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); [Children Without Worms, 2021b](https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086)). To facilitate resource allocation and deworming medication administration, workers need an effective method of determine higher-need communities ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); Children Without Worms, 2021b). We are primarily concerned with four major species of STH: roundworms, whipworms, threadworms, and hookworms ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154)). Many methods of analyzing stool samples like PCR have demonstrated better diagnostic accuracy; however, these methods require more costly equipment. Thus, the Kato-Katz method remains the “WHO golden standard” (Mbong et al., 2020). The technique performs a manual count of the number of worm eggs in the community’s stool samples. However, the method is taxing for health professionals as it requires exposure to noxious materials for long periods of time ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); [Children Without Worms, 2021b](https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086)). Consequently, contributors deliver a lower quality of work and err the data collection process ([Akintayo et al., 2018](https://doi.org/10.1038/s41598-018-27272-w)). This is troublesome as data processing and collection is cited as a time-costly step; thus, it needs to be minimized with automated sampling and analysis ([Stuyver & Levecke, 2021](https://doi.org/10.1371/journal.pntd.0009422)). Automation would process a larger number of samples accurately. Knowing the egg counts in more individuals sooner and in a convenient manner aids healthcare professionals in the care they provide patients ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); [Children Without Worms, 2021b](https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086)). A previous Capstone team proposed a simplified solution with an installed microscope where users can intuitively image slides inside the slide container. Thus, we seek to aid their efforts by providing technical support with a machine learning-backed automated diagnosis system. The primary areas of operation regarding STH research are Uganda, Bangladesh, and Kenya - areas known to have limited electricity, transportation, and laboratories ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); [Children Without Worms, 2021b](https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086); Jain & Keith, 2021). The decision to allocate de-worming medication and other resources needs information from a reliable, efficient, and standardized method of quantifying STH infections in communities ([Children Without Worms, 2021a](https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154); [Children Without Worms, 2021b](https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086); [Butploy et al., 2021](https://doi.org/10.1155/2021/6648038)).

**Project Goals & Scope**

The goal of this semester was to have an executable version of the code and data presented in the paper “Affordable artificial intelligence-based digital pathology for neglected tropical diseases: A proof-of-concept for the detection of soil-transmitted helminths and *Schistosoma mansoni* eggs in Kato-Katz stool thick smears” ([Ward et al., 2022](https://doi.org/10.1371/journal.pntd.0010500)). The code had already been analyzed and adapted to fit the team’s project requirements in the team’s previous work “GTVIPFall22 STH ML Notebook” ([Jose-Maria et al., 2022](https://www.kaggle.com/code/nathalyjosemaria/gtvipfall22-sth-ml-notebook?scriptVersionId=111027532)). The two resources, Ward et al.’s work and the Fall 2022 teamwork, were to serve as references for localizing the code for execution. The term ‘localization’ is being used to refer to the process of getting Ward et al.’s work recreated and actively executable on a computer tangible by the team - either on personal computers or Georgia Tech PACE computers. It was decided that PACE computers would be involved in this process in a previous team report due to two factors: first, the computers had far superior RAM storage necessary executing Ward et al.’s work, and second, that the resource was readily available to the team as VIP students ([Hart et al., 2022b](https://docs.google.com/document/d/1RYWKkyzSzECGPsMKJpO_tUcqUoXJWaytBaS01V0QSjg/edit?usp=sharing.)). The individual working on behalf of the team narrowed the project into specific goals to guide progress throughout the semester as follows.

**Preparations**

The team was to review all previous teamwork conducted in relation to the recreation and execution of Ward et al.’s work – including the Fall 2022 teamwork. Additionally, the team was to prepare for PACE ICE cluster usage on all devices associated in the localization process, including any downloads, permissions, or program executions.

**Kaggle Notebook Recreation in PACE ICE**

First, the team was expected to successfully import the Python code presented in Ward et al.’s work over into the PACE ICE cluster for execution. Next, the team was to run a successful execution of a small batch of data within the cluster. Lastly, the team was to run a successful execution of a larger batch of data than Ward et al.’s work was originally set up for in the same cluster.

**Kaggle Notebook Experimentation in PACE ICE**

Upon the successful localization of Ward et al.’s work, the team was tasked with manipulating specific variables to start creating Ward et al.’s work into something more of the team’s own design. Some of these variables included batch size, training steps, class label mapping, the selected base pre-trained model, and the model pipeline configuration. Next, the team was to compare the results of the various property changes and narrow the optimal choices for the project’s needs to finalize the team’s personalized update of Ward et al.’s work.

**Implement Parallel Algorithms**

As part of the work toward manipulating Ward et al.’s work into something more of the team’s own, the team was tasked with incorporating various other algorithms previously explored. One such was a circular object detection algorithm detailed in “Circular Object Detection Using a Modified Hough Transform” ([Smereka & Duleba, 2008](https://doi.org/10.2478/v10006-008-0008-9)). Other prominent ideas for consideration were detailed in the papers “A deep learning framework to discern and count microscopic nematode eggs” and “Towards an automated medical diagnosis system for intestinal parasitosis” ([Akintayo et al., 2018](https://doi.org/10.1038/s41598-018-27272-w); [Beaudelaire et al., 2019](https://doi.org/10.1016/j.imu.2019.100238)).

I**ntroduce Alternate Data Sets**

To test the robustness of the team’s model, the team was tasked with brainstorming various new data sets to test the model against. One such idea was to test with the false image set of glitter particles representing worm eggs collected in previous semesters ([Hart et al., 2022a](https://docs.google.com/document/d/1ug0yd40aWh1lJKnj1bPNfc4ENcmw4WaBULYlWT6N3K4/edit?usp=sharing)). Another idea was to test with a smaller subject of images from Ward et al.’s work. If possible, the team also sought to introduce any new images from external field work should they have been available.

**Project Status**

**Preparations**

To prepare to work with the Georgia Tech PACE ICE cluster, the team needed to investigate the full functionality of the service. This research consisted of various findings. One such was the specific CPUs available within the ICE cluster, including their GPUs and memory capabilities (Figure 7; [PACE 2023a](https://docs.pace.gatech.edu/ice_cluster/ice_resources/)). This helped inform what specific computer the project would need. Thankfully the only requirement was to have large enough RAM, and all computers listed fulfilled this requirement, so the team was able to queue jobs under any of these computers. Another learning was how the cluster was structured with its various head nodes, compute node, storage servers, and schedulers ([PACE 2023a](https://docs.pace.gatech.edu/ice_cluster/ice_resources/)). The team also learned a substantial amount about job executions: what they were, how they worked within ICE cluster queues, the different types available, how to compose them, and how to monitor their execution ([PACE 2017](https://docs.pace.gatech.edu/software/PBS_script_guide/)). The team did not expect the immense complexity that would come with learning the PACE service despite working within a singular, minor cluster. As such, it was deemed wise to allocate more time toward fully understanding the service to avoid confusion and issues during development. Moreover, the team determined to create a sort of documentation better defining the service on the ICE cluster side of things for other unknowing VIP students wanting to utilize the daunting service ([Jose-Maria 2023](PACE%20Documentation.docx)). The document details the mentioned learnings on cluster CPUs, the cluster structure, and jobs. The document also details an easier method of interaction with the cluster: Open OnDemand. Open OnDemand is a graphical user interface designed to simplify the creation, assignment, and execution of jobs within the cluster ([PACE 2023b](https://docs.pace.gatech.edu/ood/guide/)). Both the team-made documentation and Open OnDemand’s access point further detail how to get started, including connecting to the Georgia Tech campus VPN for access to the service.

**Kaggle Notebook Recreation in PACE ICE**

Once the team felt adequately informed to move forward, the team then started interacting with the ICE cluster. The team thought the best first step was to write the PBS script. They thought this was the best first step as the script was interpreted to be the catalytic step to launching the code involved in Wade et al.’s work. The script did not necessarily do that, and the team is still unclear on the intention of the script given how it was phased out of focus during development. However, this was still a valuable starting point.

The team created an initial version of the PBS script based on the recommended starter script provided by PACE ([PACE 2017](https://docs.pace.gatech.edu/software/PBS_script_guide/); Figure 1; Figure 2). The script was thought to execute successfully through Open OnDemand based on the output (Figure 3). However, this was not the case as no changes were made to the output directory. The output directory in PACE’s storage should have reflected Ward et al.’s work following the PBS script execution (Figure 4).

The team suspected that an individual file or group of files might have been blocking execution. So, the team downloaded the project files onto a personal computer to isolate them and pick-and-choose only those necessary for a minimum viable product (MVP). The files determined to be involved in the MVP and the reasoning as to why is demonstrated in Table 1. After isolating the MVP, the team reapproached the ICE cluster and reran the PBS script with the MVP as the new input to minimize the work required of the service and the confusion for potential errors.

When trying to execute Ward et al.’s work, the specifics changed depending on the approach we attempted, but they generally followed the same direction: follow the instructions in Ward et al.’s work. This meant executing each code cell consecutively from Ward et al.’s work.

When returning to the ICE cluster, the first cells from Ward et al.’s work - under the section “TensorFlow Object Detection API setup” - executed successfully (Figure 5; Figure 6; [Ward et al., 2022](https://doi.org/10.1371/journal.pntd.0010500)). However, there were several consequential errors. Many of these errors were related to the PACE CPU’s configurations, such as not having protoc downloaded or not having the correct version of pip for the project. The only way to resolve these issues was to run sudo commands, which required administrative access on a CPU. Because PACE CPUs are shared, it was impossible for the team to make these changes to the CPU in use. Thus, the team then tried to get the entire set up process working locally on a personal computer outside of Open OnDemand and executed Ward et al.’s work as far as the personal computer could support.

Local file compilation eventually reached successful execution from “TensorFlow Object Detection API setup” up to “Explore the data sets” exclusive after extensive troubleshooting ([Ward et al., 2022](https://doi.org/10.1371/journal.pntd.0010500)). This is because of the flexibility in administrative access on a personal computer compared to the shared PACE resource. Execution was successful only up to “Explore the data sets” because the following steps required a large amount of RAM that the team’s personal computer could not provide ([Ward et al., 2022](https://doi.org/10.1371/journal.pntd.0010500)). As such, the team was forced to move back to the ICE cluster.

The hope was that most of the work completed locally would require less from the PACE service and therefore minimize the number and types of errors to mitigate. However, when it got to attempting to execute the evaluation of the data set, PACE was not executing the Python file. When reaching out to PACE administrators for help, the team was instructed to simply restart execution on another service – Jupyter - within PACE as opposed to continuing development on the command line.

Despite having to undo several weeks of work, the team started the Jupyter notebook implementation optimistically because of the seemingly straightforward copy-and-paste of Ward et al.’s work since it was likewise formatted as a Python notebook. However, this was not the case. The Jupyter notebook experienced nearly all the same issues as local personal computer installation and execution, such as not being able to find the TensorFlow package, needing to reinstall a specific version of numpy, and much more (Figure 8; Figure 9; Figure 10; Figure 11; Figure 12).

Development grew quite frustrating at this point. The team was essentially redoing the same troubleshooting on a different platform for the third time, leading them to feel completely stalled in progress. One issue unique to Jupyter that was particularly challenging was the fact that two different versions of TensorFlow were competing and disrupting execution because the two different versions existed in two separate Python package locations (Figure 12).

The team thought to create an environmental variable to point to the two Python packages given the team’s understanding of environmental variables (Figure 12). However, the Jupyter notebook continued to have issues, so the team reached out to PACE administrators once again as things were starting to grow well out of the team’s scope of understanding. The PACE administrator advised to remake the project within an Anaconda environment such that pulling from several Python package locations would be possible to resolve the issue. However, upon hearing this news, the team was not willing to restart the same implementation and troubleshooting on a fourth platform. Too much time had already been spent on this endeavor, and the team did not want to waste anymore. Thus, the team decided to step back and reevaluate the steps taken in development thus far.

Given the unfortunate loss of time, it was determined that the group would leverage the remaining time to work on a literature review to better understand where things erred and how the project could improve in hopes of better advising future development. While the project did not reach adequate completion nor reach the original goals the team established, the effort spent towards development was still valuable in understanding how high-performance computing works particularly with python-based biological data analysis. It was rewarding to the team to learn more about this application of computer science within the life sciences.

**Conclusions**

This project highlighted that the team was simply under-informed and under-resourced when deciding to undertake such a large task. This proves to be a valuable lesson in learning how to establish realistic and sufficiently informed expectations. However, the team worked well when dealing with rapid changes and rescoping to still have evident learnings, achievements, and areas for improvement. The team’s struggles with the PACE ICE cluster will hopefully help future team members or other VIP students trying to use the cluster. Ultimately, the team’s hope is that others can learn from their issues and to not repeat the actions that hindered their own development.

Regarding future work, the team would advise having more than sufficient knowledge on high performance computing to extensively understand each step that should be involved in a program execution process. The team would still advise working with the PACE ICE cluster given the easy accessibility to such powerful resource; however, the team would also advise having a knowledge mentor or advisor that can better aid troubleshooting and project errors in a way that PACE administrators cannot understand having not worked directly with the project. Moreover, the team would advise to continue the localization of Ward et al.’s work to get an object detection model for STH egg counting functional despite not being necessarily unique or original quite yet.

**Addendum**

**Table 1. Minimum Viable Product Files and Reasoning**

|  |  |
| --- | --- |
| **File Name** | **Reason for Inclusion** |
| Egg\_label\_map.pbtxt | This file contains the mapping between a number and specific STH species necessary throughout training, evaluating, and testing. |
| Pipeline.config | This file specifies how to execute the train to evaluate to test procedure. |
| Rfcn\_resnet101\_sthsch\_1  /frozen\_inference\_graph.pb | This file specifies the structure of this particular network. |
| Rfcn\_resnet101\_sthsch\_1  /pipeline.config | This file specifies how to execute the train to evaluate to test procedure for this particular network. |
| Train.record | This file contains the input data set meant to be evaluated. |
| Test.record | This file contains the ground truth to train the input data set against. |
| Eval.record | This file contains the output from evaluating the input data set. |

When aiming to run a small subset of Ward et al.’s data through the PACE ICE cluster, the team realized that the large number of files involved may be adding to the confusion and errors the team experienced during execution. Thus, the team decided to select the most important input files for the project in hopes of simplifying the outputs seen and the resulting analyses.

**Figure 1. Version 1 PBS Script**

**A screenshot of a computer program

Description automatically generated**

This is the first PBS script the team attempted to run when starting to work with the PACE ICE cluster. This script was constructed based of PACE’s recommended starter script demonstrated in Figure 2. The script is likewise available within PACE’s documentation ([PACE 2017](https://docs.pace.gatech.edu/software/PBS_script_guide/)). Note that there were not many requirements detailed within the script besides the requirement for atleast 6GB of memory to start.

**Figure 2. Starter PBS Script**

**A computer screen shot of a computer error

Description automatically generated**

This is the recommended starter script provided by PACE to run jobs through their clusters ([PACE 2017](https://docs.pace.gatech.edu/software/PBS_script_guide/)). Note the difference between the team’s script and this script in that the team’s script detailed less requirements since the only limitation that needed to be met was sufficient RAM space. Limiting the number of requirements detailed in a PBS script increases the number of computers a job can run from since there are less restrictions for the scheduler when selecting.

**Figure 3. Seemingly Succesful PBS Execution**

**A screenshot of a computer

Description automatically generated**

This was the output file when running the team’s PBS script through an ICE cluster. The output echos the requirements detailed in the original script, such as a memory requirement of 6GB, the job’s name, and the queue for placement. Given that the output solely printed information about the script led the team to believe that the script was not working as expected. The team expected some errors at the very least to work from.

**Figure 4. Expected Output File Directories**

**A screenshot of a computer

Description automatically generated**

This is the expected output directory of the PBS script execution as it should have appeared within PACE storage based of Ward et al.’s work ([Ward et al., 2022](https://doi.org/10.1371/journal.pntd.0010500)). Following the execution of the PBS script within Open OnDemand, it was determined that the following files were missing from the output directory despite being present in Ward et al.’s Kaggle notebook output directory: /confusion\_matrix, /dataset, /efficientdet\_d0\_coco17\_tp, /models, confusion\_matrix.py, and /rfcn\_resnet101/detections.tfrecord.

**Figure 5. Successful Execution of Ward et al.’s First Code Cell Command 1**

A computer screen shot of a program

Description automatically generated

This demonstrated a screenshot of the Open OnDemand terminal when running Ward et al.’s work’s first code cell’s first command. The command was “git clone <https://github.com/tensorflow/models>”, seen in line three of the image. This command was responsible for downloading TensorFlow models directly from the service. The team planned on using these models as base for subsequent model training so as not to ‘reinvent the wheel’.

**Figure 6. Successful Execution of Ward et al.’s First Code Cell Command 2**

A screenshot of a computer

Description automatically generated

This demonstrated a screenshot of the Open OnDemand terminal when running Ward et al.’s work’s first code cell’s second command. The command was “cd models && git checkout ac8d06519”, seen in line one of the image. This command was responsible for navigating into the specific branch that was downloaded in the previous command execution imaged in Figure 5. This execution ensures the git branch was successfully cloned and works appropriately on the PACE CPU.

**Figure 7. Available ICE Cluster Computer Nodes**

A screenshot of a computer

Description automatically generated

This image details the specific CPUs available in Georgia Tech’s PACE ICE cluster, and this information is available in PACE’s documentation ([PACE 2023a](https://docs.pace.gatech.edu/ice_cluster/ice_resources/)). Since first researching this topic, the available computer nodes has since changed. However, given the loose requirements of the project’s computing capabilities, this image goes to show that any of the CPUs available in the cluster are sufficient for the execution of the team’s program.

**Figure 8. Jupyter Notebook Error: Outdated Pip Version**



This screenshot of the Open OnDemand Jupyter notebook demonstrates one prominent error during development. The PACE CPU had issues with its configuration, particularly with its pip package. This was a familiar error that was also experienced during local personal computer file localization. The solution is simply by running one format or another of “pip install” with a specific, compatible pip version or an upgrade flag as recommended in the image. However, it is still frustrating when a number of these minor configurations add up and require action.

**Figure 9. Jupyter Notebook Error: Outdated Protoc Version**

A pink screen with black text

Description automatically generated

This screenshot of the Open OnDemand Jupyter notebook demonstrates one prominent error during development. The PACE CPU had issues with its configuration, particularly with its protoc package. This was a familiar error that was also experienced during local personal computer file localization. The solution is simply by running one format or another of “pip install protobuf” with a specific, compatible protobuf version specified. However, it is still frustrating when a number of these minor configurations add up and require action.

**Figure 10, Jupyter Notebook Error: Outdated Numpy Version**

A screenshot of a computer

Description automatically generated

This screenshot of the Open OnDemand Jupyter notebook demonstrates one prominent error during development. The PACE CPU had issues with its configuration, particularly with its numpy package. This was a familiar error that was also experienced during local personal computer file localization. The solution is simply by running “pip install numpy” with a specific, compatible numpy version specified. However, it is still frustrating when a number of these minor configurations add up and require action.

**Figure 11. Jupyter Notebook Error: Missing Tensorflow Package**

cd models/ research 
# You can optionally test the Object Dectection API is working correctly 
python object _ py 
Traceback (most recent call last) : 
File " 
1/ models/research/object line 22, in 
import tensorflow. compat.vl as tf 
File " /home/hice1/njosemaria3/ . local/Iib/python3.9 /site—packages/tensorflow/ init_.py", 1 
ine 41, in <module> 
from tensorflow.python.tools import module_util as _module_util 
File " /home/hice1/njosemaria3/ . local/ lib/python3.9 /site—packages/tensorflow/python/ init_ 
_.py", line 40, in <module> 
from tensorflow. python. eager import context 
File " ,/home/hice1/njosemaria3/ . loca1/Iib/python3.9/site—packages/tensorf10w/python/eager/co 
ntext.py" , line 32, in <module> 
from tensorflow. core. framework import function_pb2 
ModuleNotFoundError: No module named ' tensorflow. core ' 

This screenshot of the Open OnDemand Jupyter notebook demonstrates one prominent error during development. The PACE CPU had issues with its configuration, particularly with its Python package locations. This was a unique error that was not experienced during local personal computer file localization. This issue occurred because there were two locations being referenced for Python packages: a PACE CPU storage location, and a PACE user storage location. The former had a path name of “/usr/local/pace-apps/manual/packages/tensorflow-gpu/2.9.0/lib/python3.9/site-packages/tensorflow”, and the latter had a path name of “/home/hice1/njosemaria3/.local/lib/python3.9/site-packages/tensorflow/”. The full Tensorflow package, including the “core” folder that is missing in the image, was available in the PACE CPU storage location, and not in the PACE user storage location. However, the program kept looking at the PACE user storage location for the package. As such, the team responded by trying to add an environmental variable as seen in Figure 12.

**Figure 12. Jupyter Notebook Error: Environmental Variable Configuration**

In [16] : 
"bash 
models / research 
cd 
# remove njosemaria3 site—packages from psth 
python —c " import sys; print(sys.path); sys . path.remove( ' /home/hice1/njosemaria3/ . local/1ib/python3.9/site—packages ' ) ; 
# You can optionally test the Object Dectection API is working correctly 
python .py 
[ ' ' , ' ' /usr/local/ 
' / hom 
u/2.9.O/1ib/python3.9 , 
e/hice1/njosemaria3/ . local/1ib/python3.9/site—packages' , ' 
ib/python3.9/ site—packages ] 
[ ' ' , ' ' /usr/local/ 
pace—apps /manua1/packages/tensorf10w—gpu/2.9. O/ lib/python39. zip ' , 
' /usr/ local / pace—apps/manual /packages /tensorflow—gp 
, , /usr/ 
local/pace—apps /manual/packages /tensorf10w—gpu/2.9. O / lib/python3.9 / site—packages ' ] 
Traceback (most recent call last) : 
File " /home/hice1/njosemaria3/pace—ice—home—data/ondemand/data/sys/myjobs/projects/defau1t/1/mode1s/research/object 
, line 22, in 
import tensorflow.compat . VI as tf 
File " /home/hice1/njosemaria3/ . local/1ib/python3.9/site—packages/tensorf10w/ init .py", line 41, in <module> 
from tensorflow. python. tools import module_util as _module_util 
File " /home/hice1/njosemaria3/ . local/1ib/python3.9/site—packages/tensorf10w/python/ init .py", line 4(), in <modul 
from tensorflow. python. eager import context 
File " /home/hice1/njosemaria3/ . local/1ib/python3.9/site—packages/tensorf10w/python/eager/context.py", line 32, in < 
module> 
from core. framework import function_pb2 
ModuleNotFoundError: No module named 'tensorflow. core ' 
CalledProcessError 
Traceback (most recent call last) 
<cell line: 1>() 
Input 
In [16] 
1 ) . run_cell_magi 
sth\npython —c "import sys; print(sys.path); sys.path.remove(\' /home/hice1/njosemaria3/ .10ca1/1ib/python3.9/site—pack 
ages \ ' ) ; print(sys . path) You can optionall 

This screenshot of the Open OnDemand Jupyter notebook demonstrates one prominent error during development. The PACE CPU had issues with its configuration, particularly with its Python package locations. This was a unique error that was not experienced during local personal computer file localization. The proposed solution was to add an environmental variable within the notebook that would point to the PACE CPU storage location instead of the PACE user storage location detailed in Figure 11 to eliminate Tensorflow version conflicts.

**References**

Akintayo, A., Tylka, G.L., Singh, A.K., Ganapathysubramian, B., Singh, A., & Sarkar, S. (2018). A deep learning framework to discern and count microscopic nematode eggs. *Sci Rep* 8, 9145. <https://doi.org/10.1038/s41598-018-27272-w>

Beaudelaire Saha Tchinda, Michel Noubom, Daniel Tchiotsop, Valerie Louis-Dorr, Didier Wolf (2019). Towards an automated medical diagnosis system for intestinal parasitosis. Informatics in Medicine Unlocked, Volume 16, 2019, 100238, ISSN 2352-9148, <https://doi.org/10.1016/j.imu.2019.100238>.

Butploy, N., Kanarkard, W., & Intapan, P.W. (2021). "Deep Learning Approach for Ascaris lumbricoides Parasite Egg Classification", *Journal of Parasitology Research*, vol. 2021, Article ID 6648038, 8 pages.<https://doi.org/10.1155/2021/6648038>

Children Without Worms (2021, May 27). "Innovative Technology to Address the Elimination of Morbidity of Soil Transmitted Hemelinth."<https://gatech.instructure.com/courses/214292/files/24538503?module_item_id=1970154>

Children Without Worms (2021). "Welcome Georgia Tech BME Students!" <https://gatech.instructure.com/courses/214292/files/24797641?module_item_id=1982086>

Hart, A., Jose-Maria, N., Lee, S., & Li, C. (May 2022). Intestinal Worms Final Technical Report. Google Doc. <https://docs.google.com/document/d/1ug0yd40aWh1lJKnj1bPNfc4ENcmw4WaBULYlWT6N3K4/edit?usp=sharing>

Hart, A., Jose-Maria, N., Tanguturi, S., Jayanthi, R., & Shalom, E (December 2022). Intestinal Worms Final Technical Report. Google Doc. <https://docs.google.com/document/d/1RYWKkyzSzECGPsMKJpO_tUcqUoXJWaytBaS01V0QSjg/edit?usp=sharing>.

Jose-Maria N. (2023). *PACE Documentation*. Word Document. [PACE Documentation.docx](https://gtvault-my.sharepoint.com/:w:/g/personal/njosemaria3_gatech_edu/ETLoZafLY-dFjyJmeCzEa-wB4450KEGhHusnJwecore4zQ?e=yr3qdr)

Jose-Maria, N., Hart, A., & Tanguturi, S. (2022). GTVIPFall22 STH ML Notebook. (Version 7). Kaggle. <https://www.kaggle.com/code/nathalyjosemaria/gtvipfall22-sth-ml-notebook?scriptVersionId=111027532> Adapted from Ward et. al, 2022b.

Mbong Ngwese, M., Prince Manouana, G., Nguema Moure, P. A., Ramharter, M., Esen, M., & Adégnika, A. A. (2020). Diagnostic techniques of soil-transmitted helminths: Impact on control measures. *Tropical Medicine and Infectious Disease*, *5*(2), 93.

PACE (2023, May 12). ICE Cluster Resources. *PACE Cluster Documentation*. <https://docs.pace.gatech.edu/ice_cluster/ice_resources/>.

PACE (2023, May 8). Open OnDemand Instances for PACEs Clusters. *PACE Cluster Documentation*. <https://docs.pace.gatech.edu/ood/guide/>.

PACE (2017). PBS Scripting Guide. *PACE Cluster Documentation*. <https://docs.pace.gatech.edu/software/PBS_script_guide/>.

Smereka,M. & Dulęba,I.(2008, March). Circular Object Detection Using a Modified Hough Transform. International Journal of Applied Mathematics and Computer Science,18(1) 85-91. <https://doi.org/10.2478/v10006-008-0008-9>

Stuyver LJ, Levecke B (2021) The role of diagnostic technologies to measure progress toward WHO 2030 targets for soil-transmitted helminth control programs. PLOS Neglected Tropical Diseases 15(6): e0009422.<https://doi.org/10.1371/journal.pntd.0009422>

Ward P, Dahlberg P, Lagatie O, Larsson J, Tynong A, Vlaminck J, et al. (2022) Affordable artificial intelligence-based digital pathology for neglected tropical diseases: A proof-of-concept for the detection of soil-transmitted helminths and Schistosoma mansoni eggs in Kato-Katz stool thick smears. PLoS Negl Trop Dis 16(6): e0010500. <https://doi.org/10.1371/journal.pntd.0010500>