Scaling Up: Incorporating HPC experience into an undergraduate Introduction to Data Science course using Gateways with Jupyter notebooks and GPU-enabled instances

SGX3 award # 2231406

Revised Course Description

The **revised course description** will add a new module, ***Data Science using a GPU-enabled HPC*** at the end of the quarter-length course.

CS 356: Special Topics- Introduction to Data Science
The goals of this course are to learn how to acquire, clean, analyze, and visualize data using Python, libraries including Pandas, and Jupyter notebooks.

In the revised description, the added module will have students explore an HPC platform, detect GPUs in their Jupyter notebooks, and load a large dataset to build a machine learning model. Given code to train a model, students will learn how to run jobs on the HPC cluster, and store results for analysis.

Learning outcomes:
Students will

- Familiarize themselves with HPC environments and workflows to do analytic tasks
- Move data and code into the compute space, build a machine learning model, save it, and test it
- Utilize GPUs for training and explore the exciting realm of HPCs for scientific research

Implementation Schedule

Fall 2023:
Explore Gateways resources, using ACCESS-CI obtain accounts for instructor and ensure availability for ~25 students to have their own spawned containers

Winter 2024:
Develop Jupyter notebooks, load data and notebooks into an instance for testing, check for compute-credits needed to run the module, budget as needed

Spring 2024:
Begin course, add and test student accounts, implement module

Summer 2024:
Refine/refactor material, prepare for second offering Fall 2024/Winter 2025

Your feedback is welcome!

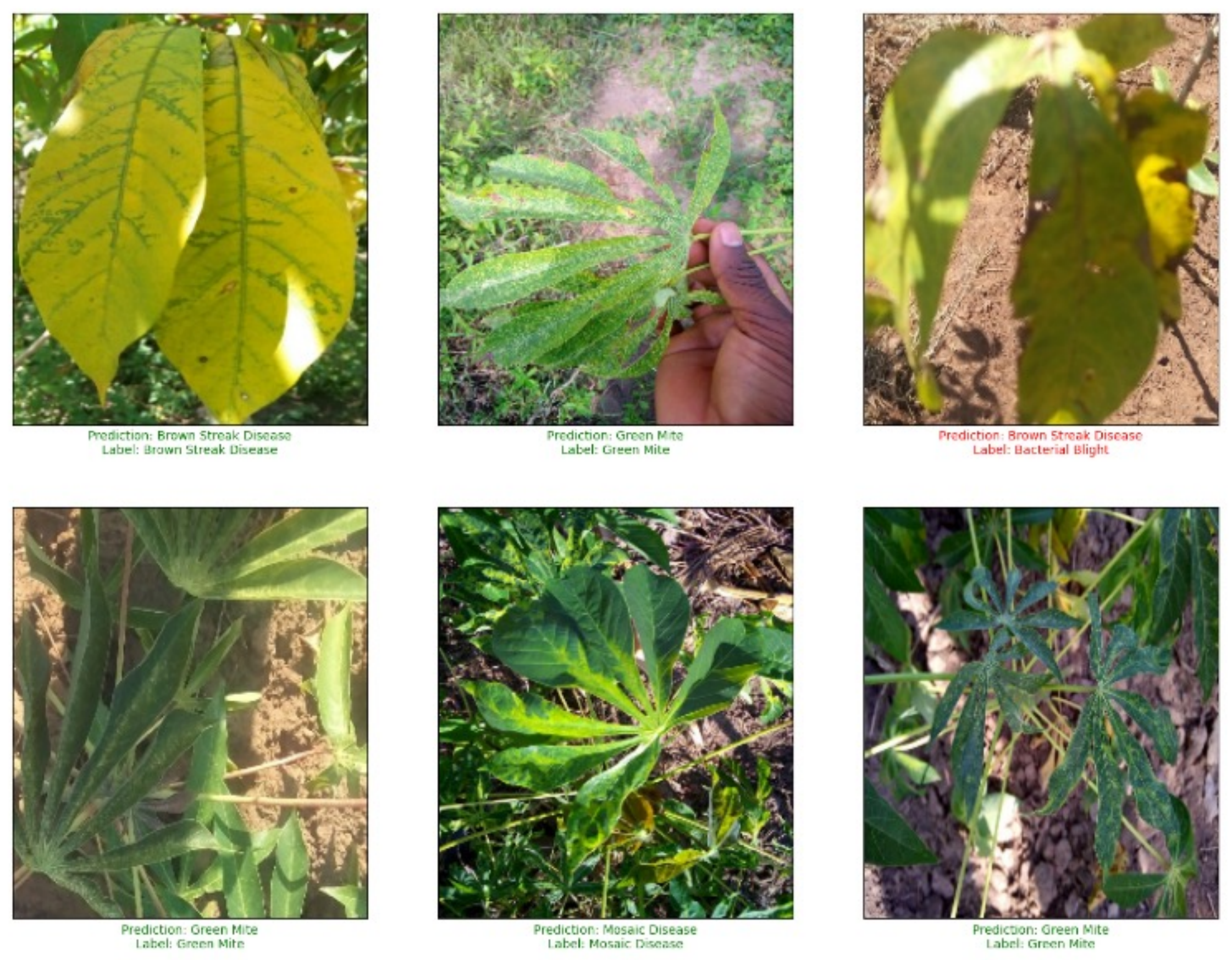
Sample HPC/Gateways Exercise

Image classification using machine learning is an effective way to introduce HPCs and the necessity of GPUs to newcomers.

We explore Cropnet classifier[1], a Tensorflow model that takes images of cassava leaves as input and detects various diseases if present. Cassava root is a major source of food across the world.

Previous to this exercise, students will have already studied the cassava project and its aims, and tested the existing trained model and code. Now, the focus is on retraining the model using GPUs on a cluster, in a platform. In this exercise, students write code to 1) detect GPUs, load the training, test, and validation sets, and train the model and save it.

```
[ ]  plot(examples, predictions)
```



[1] https://tfhub.dev/google/cropnet/classifier/cassava_disease_V1/2

Resource Needs/List

- ○ HPC Platform that allows for ~25 student accounts GPU access, ideally GitHub authentication to access containers
- ○ Jupyter notebooks in HPC environment, each student with JupyterHub spawned Kubernetes cluster
- ○ Way to load and download data
- ○ Compute credits to train ~25 ML models

Gateway Community Mentor Syllabus Suggestions

Mentors suggested to show the utility of using HPCs at scale, and to experience what happens when local computers cannot compute certain tasks requiring a GPU.

Proposed activity: Have students attempt to train the model on their laptops or desktops. It will likely fail, or take a considerable amount of time.

Next, have students try to train the model on the free version of Colab. Here, too, it will likely fail, timeout, or take a considerable amount of time to run.

Next, have students train the model on an HPC instance. Students will see the necessity and efficacy of using HPC instances to train machine learning models requiring GPUs.

Resources / Science Gateways

- JupyterHub: for notebooks
- TACC: for clusters and jobs
- Jetstream2: for clusters and jobs
- Exosphere: for interfaces
- SciServer: for domain science resources

Use Cases

- Students will access TACC]to learn how to familiarize themselves with cloud-browser-based platforms, log in to an instance, and understand how to run jobs
- Students will use Jetstream2 to have their own container spawned for them, see pre-loaded configurations and then add libraries as needed
- Students will use exosphere to explore GUI interfaces to Jetstream2
- Students will use SciServer for domain science examples and compute environments

Special Thanks

Charlie Dey, Texas Advanced Computing Center
Je'aime Powell, Texas Advanced Computing Center
Linda Hayden, Elizabeth City State University

Datasets

- Cassava leaf image dataset: https://www.tensorflow.org/datasets/catalog/cassava
- Kaggle competition: https://www.kaggle.com/c/cassava-disease/overview

Possible Expansions

- From the Cassava disease detection model, students can create a TF-lite app to deploy and run on mobile phones to detect plant disease in the field

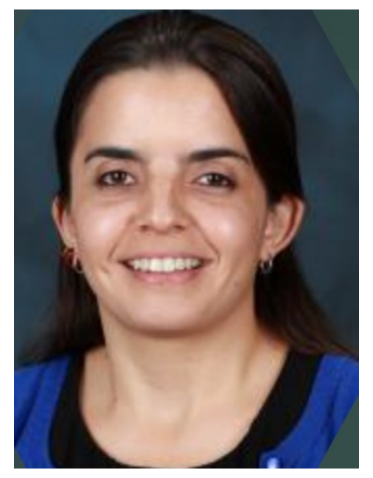
-Deploy a Cassava disease detection model to run on an endpoint and provide a method to upload images and detect them in batch or real-time

-With more cassava plant disease data or even different plant disease data, build a transfer learning model and test its efficacy and deploy it via an endpoint or a mobile app

Authors

Bernie Boscoe
Assistant Professor, Computer Science
Southern Oregon University
boscoeb@sou.edu

HPC/Gateways Mentor
Mohamed Elbakary, PhD
Associate Professor of Electrical and Computer Engineering
Elizabeth City State University
melbakary@ecsu.edu



HPC/Gateways Mentor
Veronica Vergara
Oak Ridge National Laboratory
vergaravg@ornl.gov

MORE INFORMATION → https://hackhpc.github.io/facultyhack-gateways23