Research

* Neural Networks
* Convolutional Network
* Precision vs Recall
* Inference Graphs
* Data Pipelines

Kaggle Notebook Understanding

* Confusion Matrices
* Resnet Model Differences
* Experimentation
  + Batch Size & Training Steps
  + Record Files
  + Pipeline

High Performance Computing

—

* Achievements
* successes/failures
* Testing
* Outcomes
* Prototyping
* figures/tables
* Original goals?

—

What Is the Notebook Doing?

CS - What We Did

* Read through and understood what Kaggle notebook does
  + Data pipeline
  + Confusion matrices
    - Precision vs recall
* Started determining parameters/inputs for change
  + Batch Size & Training Steps
  + Record Files
  + Pipeline
* Going forward: utilize high performance computing to implement mentioned changes

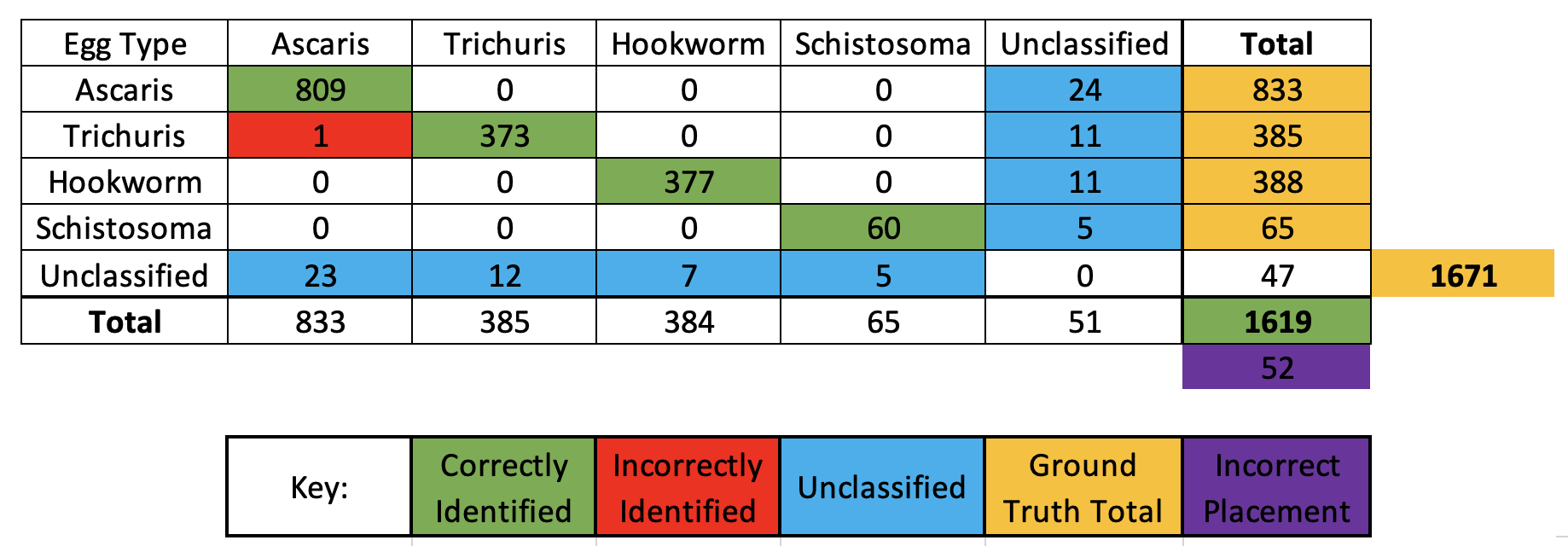
As of this previous summer, novel code in the field of ACH egg identification was released by a Johnson and Johnson (J&J) research group (Ward 2022a, Ward 2022b). The team sought to spend the majority of this semester fully understanding the J&J group’s code for its own research purposes. With that came additional research into specific aspects of machine learning as it pertained to the code to enhance comprehension.

The J&J code was stored in a Kaggle notebook, a crowd-source platform for data science and machine learning. As this was a new environment to all group members, it took some time to understand how specific aspects of the website worked, how the prewritten code could be executed, and how the code could be manipulated in a different medium independent yet connected to the original source. This led to the development of a notebook duplicated from the original copy with different wording to ease comprehension among the group and future members (Jose-Maria, Hart, & Tanguturi, 2022). Re-translating the original text seems menial and unnecessary, but it took a great amount of time to digest all the inner workings of the notebook and thus required adequate documentation to share that understanding with the rest of the team. Moreover, once the team was able to better understand what exactly was occurring, then the team could work better to propose changes that would distinguish our work from the original J&J team. The main goal of the semester was to get to a point of manipulating the original code and proposing changes with subsequent, valuable effects; however, this ended up being a future goal as comprehension of this notebook took much longer than expected, and the notebook could not run independently due to its storage limitations, which is detailed further.

As for how the notebook works, it trains and tests a machine learning model that identifies and counts specific types of STH eggs from an input image dataset. The egg types are detailed in an input egg\_label\_map.pbtxt file and can be manipulated as needed. The notebook follows a traditional machine learning process of ingesting organized data into a pipeline that is inputted to train, test and run a model (Pachyderm, 2022). As such, the input image dataset is organized by using a base pretrained model and applying transfer learning to set up the STH egg detection model. The particular model chosen in the notebook is an EfficientDet-Do, the smallest model of the EfficientDet family storage-wise, but this can also be manipulated to lessen model training time. From there a model pipeline configuration is imputed in the model to train it, the model undergoes a primary evaluation, the model is exported, and the model undergoes a final secondary evaluation. The secondary evaluation outputs a confusion matrix that details the number of correct egg type predictions and the number of confused egg type predictions\*. These values are then used to generate a precision versus recall analysis to determine the effectiveness of the model compared to the ground truth that highlights the correct number of each egg type\*\*. Since the notebook cannot be run independently, Ward et al. go on to include models pre-trained offline to demonstrate how the code should run and what the output should be (2022b).

Given this comprehensive understanding of the code’s individual parts, the team can now go about recommending specific changes to the code, inputs, and parameters, to produce different outputs and demonstrate different results. Some of these changes include batch size, training steps, class label mapping, base pre-trained model, and model pipeline configuration. Once these changes are implemented and optimized, the team can work to incorporate other datasets and ensure the model’s validity. However, this may prove to be an issue in the short-term future as credible STH datasets are currently very difficult to come across. Additionally, these changes require a more powerful computer with more storage to actually see any changes without any errors as seen in the Kaggle notebook.

\*A confusion matrix works to identify information out of several classifications and errors (Saxena, 2018). Figure 1 below is a formally annotated confusion matrix output of the faster\_rcnn\_resnet101\_sthsch\_3 model pipeline configuration pre-trained offline. The purple cell is of note as it highlights the number of eggs that were not counted as part of the ground truth total number of eggs. Other pretrained models are roughly annotated in Figures 3-7.



\*\*For further context, precision and recall will be defined with the data from the output matrix and resulting precision versus recall results of the faster\_rcnn\_resnet101\_sthsch\_3 model pipeline configuration pre-trained offline (Figure 2). The precision statistic highlights the number of relevant results (Saxen, 2018). A relevant result is a correct egg tally. For example, for the 833 eggs classified as ascaris, the 809 that were correctly classified would be considered the relevant results as opposed to the 24 that were unclassified but legitimately of the ascaris-type in the ground truth. The recall statistic highlights the accuracy of the relevant results (Saxena, 2018). Continuing with the aforementioned pretrained model, an accuracy percentage can then be derived by looking at the 809 correctly identified ascaris eggs compared to the total 833 eggs identified as ascaris across all egg types presented. For these reasons, we end with a 97.1188% precision and recall rate for ascaris classification with this model.

Confusion Matrix: 
5.] 
[ [809. 
[ 23. 
o. 
373. 
o. 
o. 
12. 
0. 
0. 
377. 
0. 
o. 
o. 
o. 
60. 
5. 
24.] 
11.] 
11.] 
1 
2 
3 
category 
Ascaris 
Trichuris 
Hookwo rm 
Schistosoma 
preclsxon 
0.5101.1 
i. 971188 
o. 968831 
o. 981771 
o. 923077 
o. 971188 
o. 968831 
o. 971649 
o. 923077 

References

(1) 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>

(2) Ward P, Dahlberg P, Lagatie O, Larsson J, Tynong A, Vlaminck J, et al. (2022). AI4NTD KK2.0 P1.5 STH & SCHm. (Version 6). Kaggle.<https://www.kaggle.com/code/peterkward/ai4ntd-kk2-0-p1-5-sth-schm#Explore-the-data-sets>

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>

Pachyderm. (2022). "What is a Data Pipeline for Machine Learning?" *Pachyderm*. Retrieved November 7th from<https://www.pachyderm.com/blog/what-is-a-data-pipeline-for-machine-learning/>.

Saxena, S. (2018). Precision vs Recall. Medium.<https://medium.com/@shrutisaxena0617/precision-vs-recall-386cf9f89488>