Bollworm Bonanza

ANLY 590 Project Proposal | October 10, 2022

Group Members

Sam Pastoriza, Haley Roberts, Scott Johnson, Matt Harding, Matt Gnanadass

Project Goal

This study seeks to build a neural network model capable of efficiently and accurately counting the number of two different types of bollworms in a given image.

Background & Objective Summary

Zindi's "Wadhwani AI Bollworm Counting Challenge" prompts competitors to create a model to count the number of two different types of bollworms in images. This need spurs from the 2017 bollworm infestation in India that resulted in major cotton farming losses. After this infestation, a company developed a mobile app in which farmers could upload pictures of their bollworm traps and receive advice on how much pesticides to use based on their level of infestation. While the company's app has been somewhat successful, they want to discover if more accurate counts can be found through modeling - hence their participation in sharing this prompt as a Zindi data competition. Considering the background of this study, the goal is to create an efficient and accurate model to count the number of bollworms present in a given image. Successful counting of bollworms can greatly benefit the cotton farming community in India.

Data Source & Description

The data is provided through Zindi's competition site and consists of two csv files (a training and a testing dataset) and an image dataset. Each row of the training dataset contains an image ID and the number of each type of bollworm contained in the image. The testing data provides only the IDs of the images to be analyzed. Finally, the image dataset contains all of the training and testing images. The final results should be in the format of image IDs combined with the number of each type of bollworm found in each image.

Methods

We plan on using a neural network (built using either PyTorch or Keras) to generate labeled bounding boxes that will identify the different types of bollworms. We will utilize neural network architectures such as Yolo, R-CNN, and RetinaNet to form the backbone to our object detection algorithm. We will train the network and perform hyperparameter tuning using a grid search to find the best model that doesn't overfit the data. To further prevent overfitting of the data, we will use cross validation during the training and tuning process. Finally, an MSE loss function will be used to compare the counted number of bollworms to the actual number of bollworms.

One of two methods will be attempted as the data is further investigated. First, two neural networks could be trained on the same data. The first model would attempt to identify one type of bollworm while the second network would identify the other type of bollworm. The results of the two models would be combined together and the number of bounding boxes calculated and returned. One issue that may arise with this method is the possibility of overlap between the two sets of results. The second method would be to train a neural network to identify both sets of bollworms in one pass. If it is possible to configure the neural network to identify the differences between the bollworms and the results, then this option may prove to be a simpler and more efficient methodology.

Expected Results & Outcomes

The result of the neural network should be a set of bounding boxes outlining each bollworm. Given either of the two proposed methods above, the bounding boxes identifying the bollworms would identify not only the presence of a bollworm but also the type. From there, the boxes could be counted for each bollworm type and the total number of each bollworm type returned to the loss function. Thus, the outcome would be a count of each type of bollworm in each image.

References

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv*. Retrieved from https://pjreddie.com/darknet/yolo/

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988). Retrieved from https://paperswithcode.com/method/retinanet

Getting Started with R-CNN, Fast R-CNN, and Faster R-CNN. MathWorks. (n.d.). Retrieved October 6, 2022, from https://www.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html