

Automated pest detection in plant trials

Presented by Ernest Uche-Eze

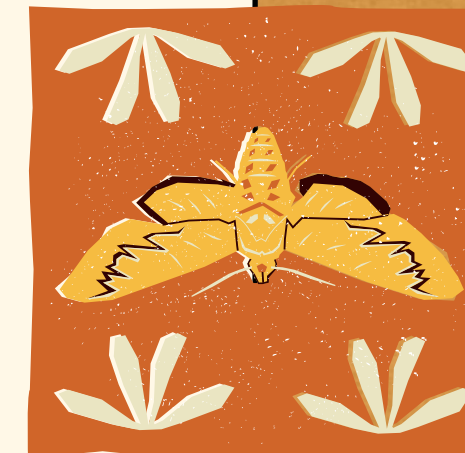
Lancaster University
e.uche-eze@lancaster.ac.uk





Overview

- We are exploring the possibility of companion planting to be applied at scale.
- This project aims to leverage AI algorithms to automate the process of detecting and tracking pests across the growing season.
- If successful, it has the capacity to replace highly inefficient manual fieldwork and reliance on pesticide use.



Data and Methods I

Data collection

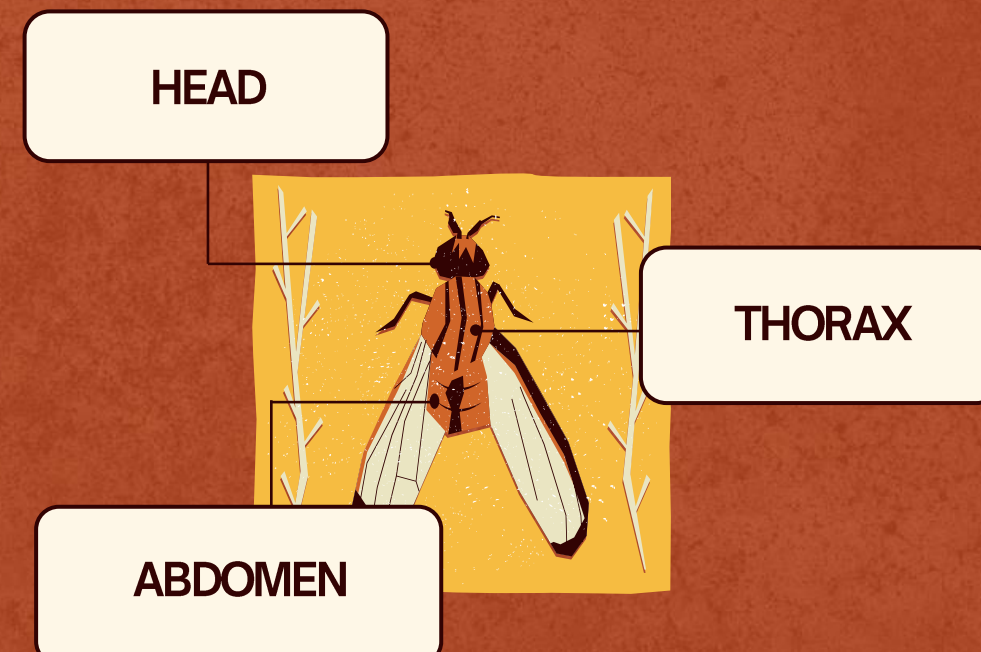
- **Zooniverse** is the world's largest online platform for citizen science.
- Volunteers gave their opinion on different images for where they thought an insect was.
- Can't use the data to train the model yet—must be cleaned and refined.



Validation

- To test our machine learning model, we must have an 'answer key'. Normally, these are the citizen-sourced classifications.
- We inspected the images to test the validity of the classifications.
- ...let's just say they were not very accurate.

The Anatomy of a Fly

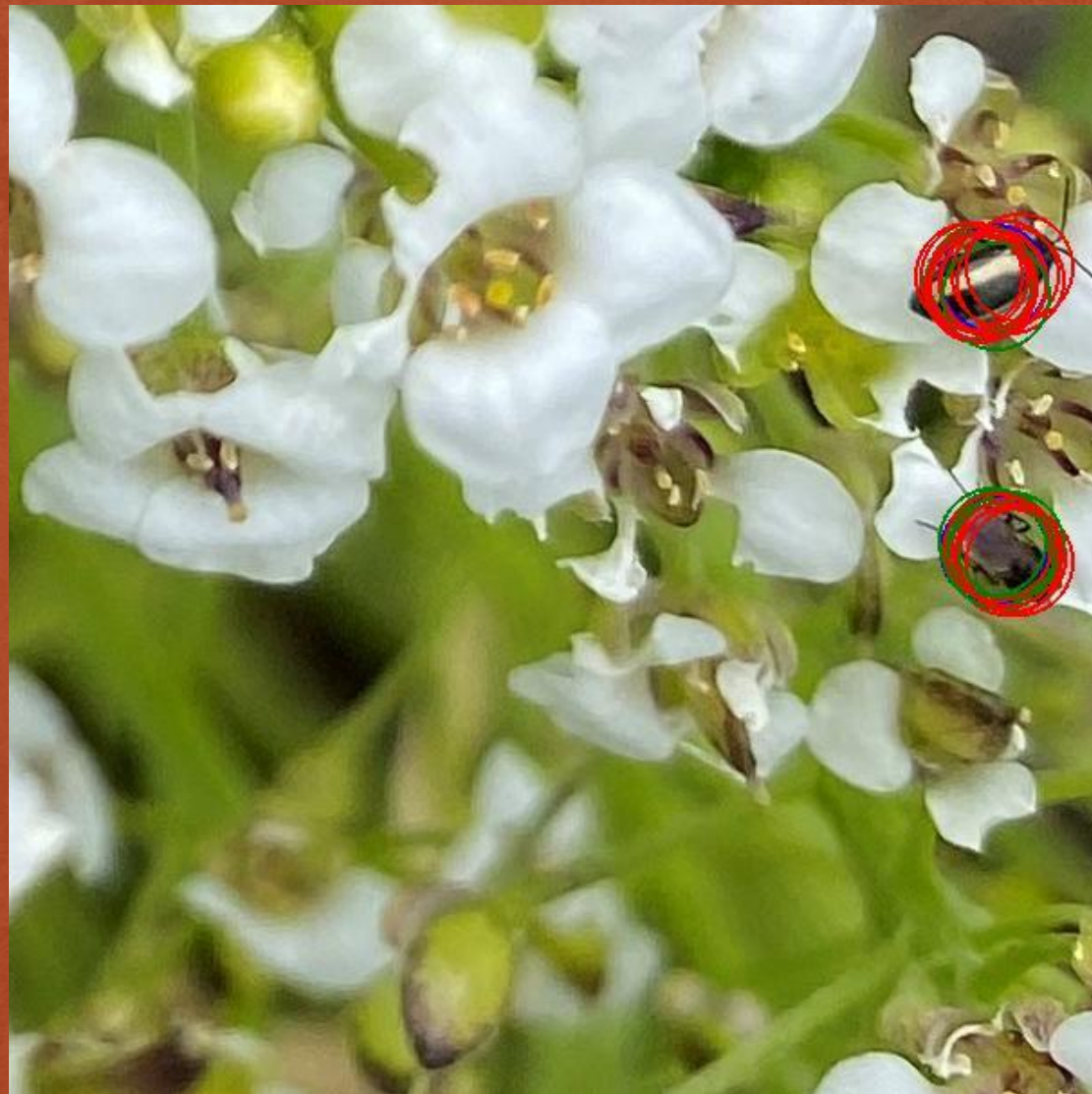


Data and Methods II

Data pipeline

- We are working towards establishing a ‘gold’ dataset to train and test the machine learning model.

1,111 INSECT classifications detected in **18,000+**



Citizen-sourced data output from Zooniverse

01

- All classifications sorted by date
- Difficult to look for and group classifications that belong to the same sub image ID



Python Script A

02

- Outputs a more readable version and removes unimportant metadata
- Outputs a version sorted and grouped by sub image ID



Python Script B

03

- Uses classification position data to plot visual selections on the sub images
- Uses two parameters, min. participants and consensus, to output a dataset



‘Gold’ Dataset

04

- The dataset was our initial impression of what would be optimal to train and test our machine learning model



Results

Classification model

- To address the class imbalance issue, we used a technique in machine learning known as **augmentation**.
- The model's algorithm was continuously optimised until we arrived at what we believed was the best given the data and time I had.
- The model had an accuracy of about 85%.

(none of these images were augmented)

True: Insect | Pred: Insect (0.56)



True: Insect | Pred: Insect (0.55)



True: Insect | Pred: Insect (0.83)



True: Insect | Pred: Insect (0.58)



Model optimisation process

I used a CNN called ResNet-18

1 & 2

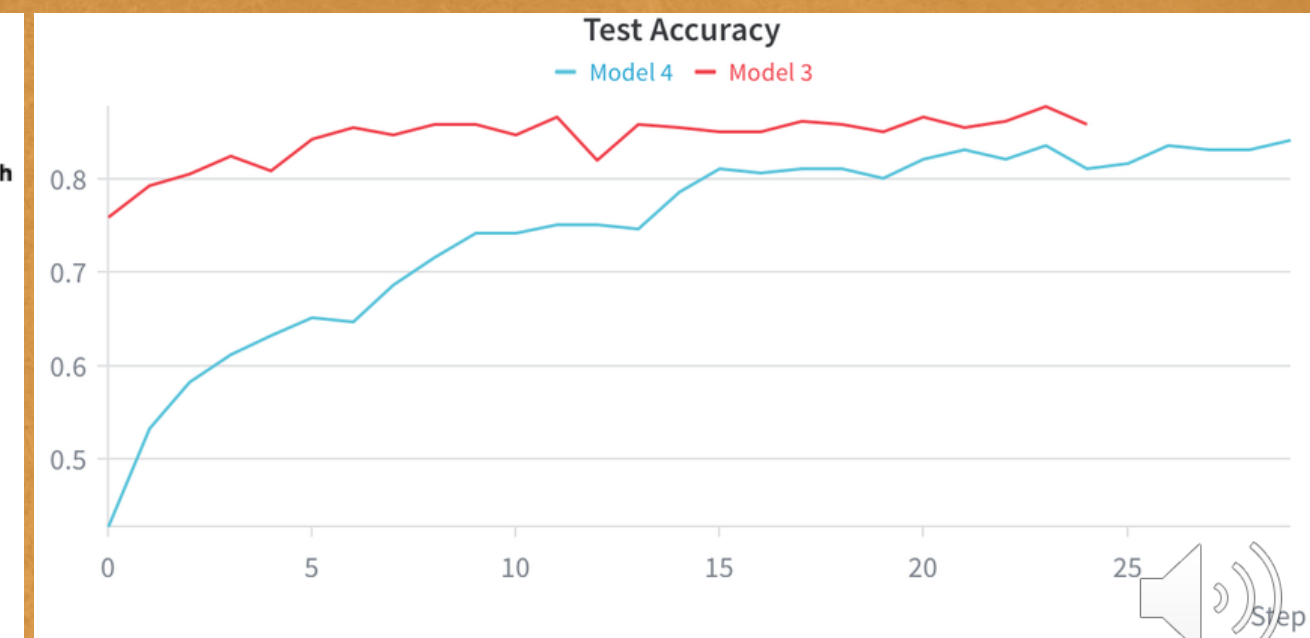
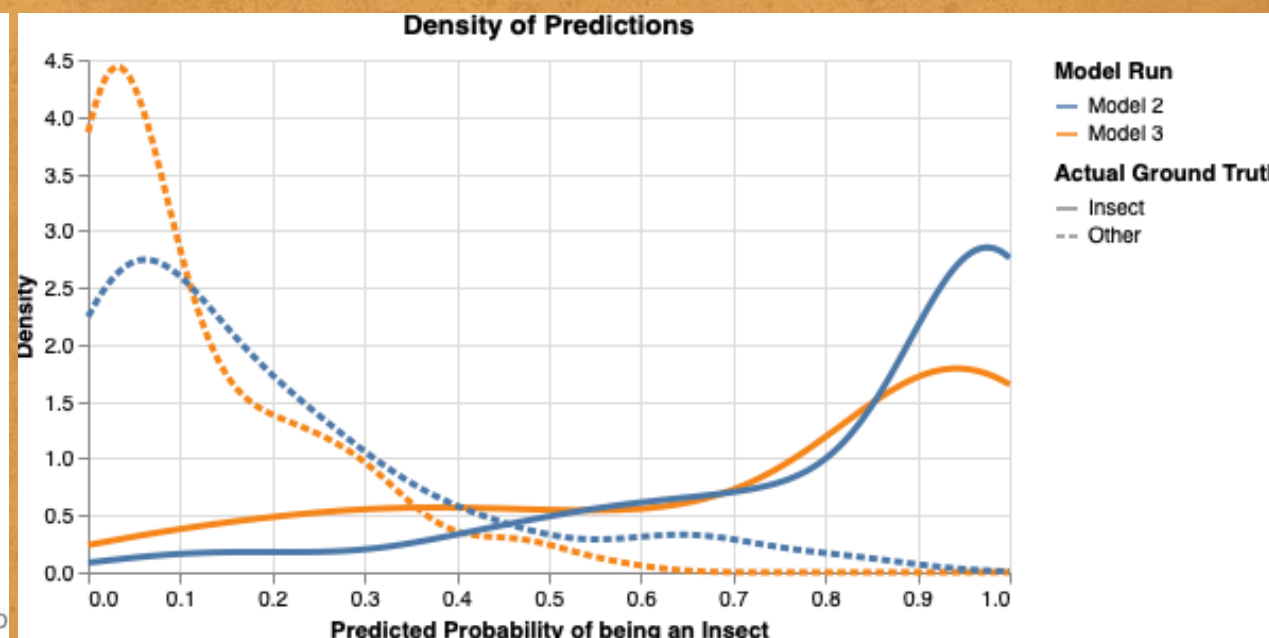
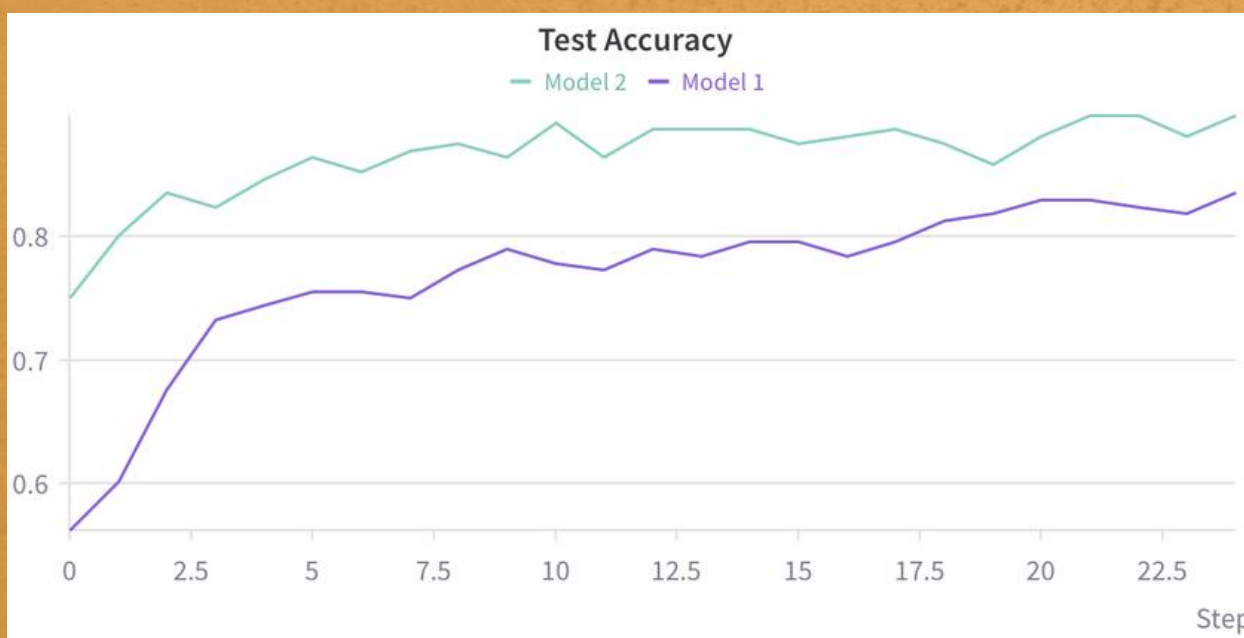
- Better accuracy
- Better test/train loss
- Good on paper, individual probabilities tell a different story

2 & 3

- Actually worse
- Severe overfitting problem

3 & 4

- Slower learning rate
- More cycles to understand features
- Might seem worse but is actually much better



Discussion

Can we automate pest detection?

Approach

- Split the images into multiple square-shaped parts.
- Filtered our dataset of images to remove bad examples.
- Generated more data by augmenting images that have insects present in them.
- Ran multiple experiments with different parameters.
- Picked the best run.

Key findings

- Augmenting images for one class because of class imbalance will have to be very limited.
- To prevent overfitting, ensure a high level of generalisation by ensuring the images are varied.
- Shortage of good examples.

Next steps

- Optimise more, consider trying a different architecture
- Get the model to identify the exact position of where it thinks the insect is
- With more data, add more classifications for the different kinds of insects—focusing on actual pests.



THANK
YOU VERY
MUCH

