Model Card and App Info v1.0

Revision Record

App name: Wheat Head Auto Counter
Model name: exp7_best.pt

App deployment status: Prototype
Version: 1.0
Date: 24-Jan-2023
Created by: vbookshelf

Known Issues

1- The model detects small portions of wheat heads that were cut off at the edges of the image. You may not want these cut off wheat heads included in the count. Therefore, it's important to review the predicted images and adjust the wheat head count if needed.

2- When there's an error the app freezes and the spinner just keeps turning. This is by design. In this prototype I didn't include error handling code. All error messages are clearly displayed in the console. This detailed information will help users to trace and fix errors.

-x-

Purpose

Manually counting wheat heads is tedious. This desktop app uses computer vision to automatically detect and count wheat heads on images of wheat fields.

The app takes png or jpg images as input. It analyzes each image and then displays that image with dots drawn on each detected wheat head. The output image is interactive. Clicking on a dot converts it into a bounding box. The app also displays the total number of dots on the image.

The predictions are made by a Yolov5m model. The model was fine tuned on data from the Global Wheat Head Dataset 2021.

Data Security

This is a desktop application.
- Data never leaves the user's pc or laptop.
- There's no tracking.
- The code is fully accessible and therefore auditable for malware.

Input

The app accepts images in png or jpg format. Multiple images can be submitted at the same time. Tiff images are not supported.

Output

The app outputs images with dots drawn on each detected wheat head. The output image is interactive. Clicking on a dot converts the dot into a bounding box. The app also displays a count showing the total number of dots on each image.

Global Wheat Head Dataset 2021 - Summary

- 6,512 wheat images, size 1024x1024, png format
- 275,371 total wheat head bounding boxes
- 125 images without wheat heads
- Images are labeled with one of 5 development stages:
  filling, filling-ripening, post-flowering, ripening and multiple
- Images come from 22 locations in 12 countries

Paper:
Global Wheat Head Dataset 2021: more diversity to improve the benchmarking of wheat head localization methods
https://arxiv.org/abs/2105.07660

Dataset on Zenodo:
https://zenodo.org/record/5092309#.Y7jTtuxBzUI

Dataset on Kaggle:
https://www.kaggle.com/datasets/vbookshelf/global-wheat-head-dataset-2021

Creating the train and val datasets

To create the training and validation datasets I first combined the train, val and test sets that come with the GWHD dataset. I removed a few duplicate images.

A domain is a combination of the region in the world where the images come from and the wheat development stage. I created 7 folds stratified by the “domain” column. By stratifying by “domain” the data also gets stratified by “development stage”.

I made sure that each fold contained images that did not have any wheat heads.

Total train images: 5,581 (236,204 total wheat heads)
Total val images: 931 (39,167 total wheat heads)

Validation Results

These were the validation results after fine tuning a Yolov5m model for 100 epochs. The model was trained on one fold only. The image size parameter was set to 512.

Class	Images	Labels	P	R	mAP@.5
all	929	39066	0.923	0.881	0.931
1	929	39066	0.923	0.881	0.931

How accurate were the wheat head counts?

I compared the actual count for each val image against the predicted count. This approach is intuitive and it provides quick insight into the model's performance. But this validation strategy is not ideal because there could be instances where there are false positives but the count could still be correct.

By development stage:

The count error was fairly constant across development stages.

	count_error
dev_stage	
filling - ripening	0.083442
post-flowering	0.102431
ripening	0.110694
filling	0.116109
multiple	0.120564

By domain:

The count error varied by domain. There are 47 domains. 32 domains had count errors less than 10 percent. 41 domains had count errors less than 20%. The table below shows the average count error for each domain.

	count_error	dev_stage	country	location
domain				
nau_2	0.025917	post-flowering	China	Baima
arvalis_7	0.029724	filling - ripening	France	VLB
cimmyt_2	0.038478	post-flowering	Mexico	Ciudad Obregon
arvalis_6	0.042522	filling - ripening	France	VSC
arvalis_9	0.046540	ripening	France	VLB
nmbu_1	0.050422	filling	Norway	NMBU
ksu_2	0.052823	post-flowering	US	KSU
arvalis_10	0.054295	filling	France	Mons
utokyo_2	0.056263	ripening	Japan	NARO-Tsukuba
inrae_1	0.057243	filling - ripening	France	Toulouse
usask_1	0.058004	filling - ripening	Canada	Saskatchewan
uq_7	0.058740	ripening	Australia	Gatton
arvalis_8	0.061337	filling - ripening	France	VLB
ethz_1	0.065629	filling	Switzerland	Usask
uq_4	0.066150	filling	Australia	Gatton
cimmyt_3	0.070050	post-flowering	Mexico	Ciudad Obregon
nau_3	0.071942	filling	China	Baima
uq_5	0.073255	filling - ripening	Australia	Gatton
utokyo_1	0.073611	ripening	Japan	NARO-Tsukuba
arvalis_3	0.074205	filling - ripening	France	Gréoux
ksu_1	0.075748	post-flowering	US	KSU
ksu_3	0.076271	filling	US	KSU
nmbu_2	0.076568	ripening	Norway	NMBU
arvalis_2	0.079269	filling	France	Gréoux
cimmyt_1	0.080881	post-flowering	Mexico	Ciudad Obregon
nau_1	0.085662	post-flowering	China	Baima
arvalis_1	0.086056	post-flowering	France	Gréoux
uq_6	0.087129	filling - ripening	Australia	Gatton
ksu_4	0.091174	ripening	US	KSU
rres_1	0.092106	filling - ripening	UK	Rothamsted
arvalis_12	0.097382	filling	France	Gréoux
arvalis_4	0.097825	filling	France	Gréoux
uliège-gxabt_1	0.103235	ripening	Belgium	Gembloux
uq_2	0.111111	post-flowering	Australia	Gatton
utokyo_3	0.120564	multiple	Japan	NARO-Hokkaido
arc_1	0.123616	filling	Sudan	Wad Medani
arvalis_5	0.129560	filling	France	VLB
ukyoto_1	0.137666	post-flowering	Japan	Kyoto
uq_10	0.138072	filling - ripening	Australia	Gatton
uq_3	0.142857	filling	Australia	Gatton
arvalis_11	0.186856	filling	France	VLB
uq_9	0.204269	filling - ripening	Australia	Brookstead
uq_8	0.206922	ripening	Australia	McAllister
uq_1	0.216057	post-flowering	Australia	Gatton
uq_11	0.248724	post-flowering	Australia	Gatton
terraref_1	0.283192	ripening	US	Maricopa,AZ
terraref_2	0.383447	filling	US	Maricopa,AZ

Hardware

- 1 x RTX A5000 GPU with 128 cpu's
- Trained on vast.ai

Misc Info

1- In many cases the model detected wheat heads that were cut off at the edges of images. The creators of the dataset intentionally did not provide bounding boxes for many of these cut off wheat heads. These detections show up as errors, but in reality they are not errors.

2- Having a human in the loop would be the best way to use this app. For each prediction, a person should inspect the dots and their associated bounding boxes, then adjust the count up or down to arrive at the final number of wheat heads.

3- I used data augmentation to improve accuracy and reduce overfitting.

4- The app can run on a CPU or GPU. The device is selected automatically. CPU inference time is about one second per image. A GPU would make the app faster, but it's not essential.

5- The app does not need an internet connection to run. However, an internet connection is needed during the initial setup.

Documentation

All code is available on GitHub. This includes the Jupyter notebooks that were used to train and validate the model.
https://github.com/vbookshelf/Wheat-Head-Auto-Counter

License

Contact

Email: contact -at- woza -dot- work