2950 Niles Road, St. Joseph, MI 49085-9659, USA
269.429.0300 fax 269.429.3852 hq@asabe.org www.asabe.org

An ASABE Meeting Presentation
DOI: https://doi.org/10.13031/aim.202100742
Paper Number: 2100742

Leveraging transfer learning in ArcGIS Pro to detect "doubles" in a sunflower field

Rai. N [a], Flores. P [a]

[a] Department of Agricultural and Biosystems Engineering, North Dakota State University, Fargo, ND 58102.

Written for presentation at the
2021 Annual International Meeting
ASABE Virtual and On Demand
July 11–14, 2021

ABSTRACT.

The occurrence of doubles or multiple plant cluster in a sunflower (Helianthus annuus L.) field is undesirable since they can contribute to disease spread leading to yield reduction. A common approach to identify doubles in a sunflower field is by in-person field scouting, which can be arduous and time-consuming task. Therefore, a method was proposed to use unmanned aerial systems (UASs) imagery coupled with transfer learning in ArcGIS Pro to automate the process of identifying and mapping the occurrence of sunflower doubles across the field. The imagery was collected from a sunflower field, around a week after germination using a Phantom 4 Pro flown at an altitude of 40ft above ground level (AGL). The orthomosaic imagery was manipulated in Python where the exported image chips were trained on 4 architectures, namely, Faster R-CNN, RetinaNet, YOLOv3, and SSD. These 4 architectures were further fine-tuned to detect doubles in the sunflower field. Results showed that Faster R-CNN outran the rest of the models to detect sunflower doubles across the whole orthomosaic imagery. The average precision accuracy of Faster R-CNN was 99.6% with an average F1-score of 0.95, precision of 0.91, and recall of 0.96. On the other hand, RetinaNet and YOLOv3 yielded an average precision score of 93.8% and 87.8%, respectively. Based on these scores, we recommend Faster R-CNN to detect doubles on sunflower fields. For our future work, we are working to automate sunflower stand counts while adding another class for weed detection across the field as well.

Keywords.

Sunflower doubles, Transfer learning, UASs, Double detection

Introduction

North Dakota's sunflower *(Helianthus annuus L.)* production for 2020 was around 2.81 billion pounds, up 44% from 2019. The estimated average yield in 2020 was 1,761 pounds per acre (National Sunflower Association, 2020). Given the vast volume of production, it is desirable to develop solutions that would allow farmers and researchers to gain more insight about the crop in the field. Along those lines, the use of unmanned aerial systems (UASs) imagery coupled with computer vision holds a promising future. One topic of special interest for sunflower growers is the occurrence of doubles (two or more plants in a cluster) across the field. Doubles in a sunflower field increase the risk of disease spread on the plant in a cluster. It also creates competition among plant cluster, resulting in smaller heads thereby producing smaller seeds. A common method to identify doubles in a large field is accomplished by field scouting. Currently, manual field scouting is becoming obsolete due to advancements in the area of high-throughput crop phenotyping.

To address the task of sunflower doubles detection using UASs imagery, we have leveraged transfer learning in ArcGIS Pro (ESRI, Redland, CA), a software that has the ability to preserve the geospatial information of orthomosaic imagery and handles a hefty amount of vector data called shapefiles. The task of drawing shapefiles over the object of interest is achieved through manual digitization, which demands time and is considered an inept way of extracting information from the objects of interest (Holman et al., 2016). The main bottleneck to detect sunflower doubles in ArcGIS Pro is manual digitization. Therefore, to address the problem of doubles detection by eliminating the arduous manual digitization method, we propose a state-of-the-art solution to this problem by using a method in deep learning called transfer learning. Transfer learning aims at transferring knowledge from source domain to the target learner therefore there should be a feature commonality between the two domains (Qi et al., 2020). Because of this feature, a model can be trained on a small dataset thereby eliminating the need for arduous data-labelling efforts (Nowakowski et al., 2021). The application of transfer learning in ArcGIS Pro has achieved land cover classification, road extraction, human settlements classification, tree point classification (ESRI). Since, a plethora of tasks have been achieved using transfer learning in ArcGIS Pro, we grab the opportunity to integrate this method in agricultural domain by extending the application of ArcGIS Pro to accomplish the task of sunflower detection.

In our study, we have used sunflower images to train single shot detector (SSD), RetinaNet, YOLOv3, and FasterRCNN architectures. ResNet50 has been used alongside SSD, RetinaNet, and FasterRCNN as a backbone model for feature extraction, whereas, Darknet-53 has been used alongside YOLOv3. Since ArcGIS Pro is compatible with Python (open-source) platform, all the image processing tasks which are usually done using the graphical user interface (GUI) clicks in the ArcGIS Pro environment have been automatized with a logically organized workflow script. Also, the task of detecting sunflower doubles using Python script has been automatized outside ArcGIS Pro environment. Therefore, the specific objectives of this paper were, 1) to develop a workflow script for detecting sunflower doubles, 2) compute metrics to perform detection accuracy assessment, 3) perform model comparison based on the acquired metrics.

Materials and methods

Detailed methodology of this study, including image data acquisition, exporting image chips, image meta-data format, training image chips using appropriate hyperparameters, detecting sunflower doubles, and variations of metrics has been discussed in subsequent sections.

Data acquisition and description

Sunflower plots (fig. 1a, 50 ft long by 30 ft wide) planted on May 25, 2017 with a 30-inch row spacing. A small UAS (DJI Phantom 4 Pro) was flown on June 7, 2017 from the experimental area. The phantom 4 Pro is an off-the-shelf, ready-to-fly quadcopter, fitted with a 20.1-megapixel (MP) RGB camera. Missions were flown at 40 ft above ground level (AGL). The images collected were processed using Pix4Dmapper Pro by Pix4D. The resulting orthomosaic raster had a ground resolution of around 0.1 in/pixel. That resolution allowed for clear observation of individual sunflower plant, with few exceptions. The final raster was RGB (3-bands, 1.03GB in size) orthomosaic with a spatial resolution of 22,000 x 17,000 (columns by rows of pixels) and tagged image file (*.TIF) format.

Exporting data and data preparation

Exported dataset for deep learning in ArcGIS Pro comprises of small subimages (fig. 1c) containing the feature class or objects of interest, called as, image chips. Since, we were interested to automate doubles detection using python scripting, we used a semi-automated approach to export image chips. This approach combined manual annotation of sunflower doubles (fig. 1b) using *Training Samples Manager* tool in ArcGIS Pro and *Export Training Data for Deep Learning* function in Python to export 1000 s of image chips (fig. 1c) for training purpose. This dataset was further augmented (fig. 1e, with varying field lighting conditions, and rotations) to create a massive dataset consisting 8,254 (6 GB in size) images, from which 80% of the images were used for training and 20% were used for testing purposes.

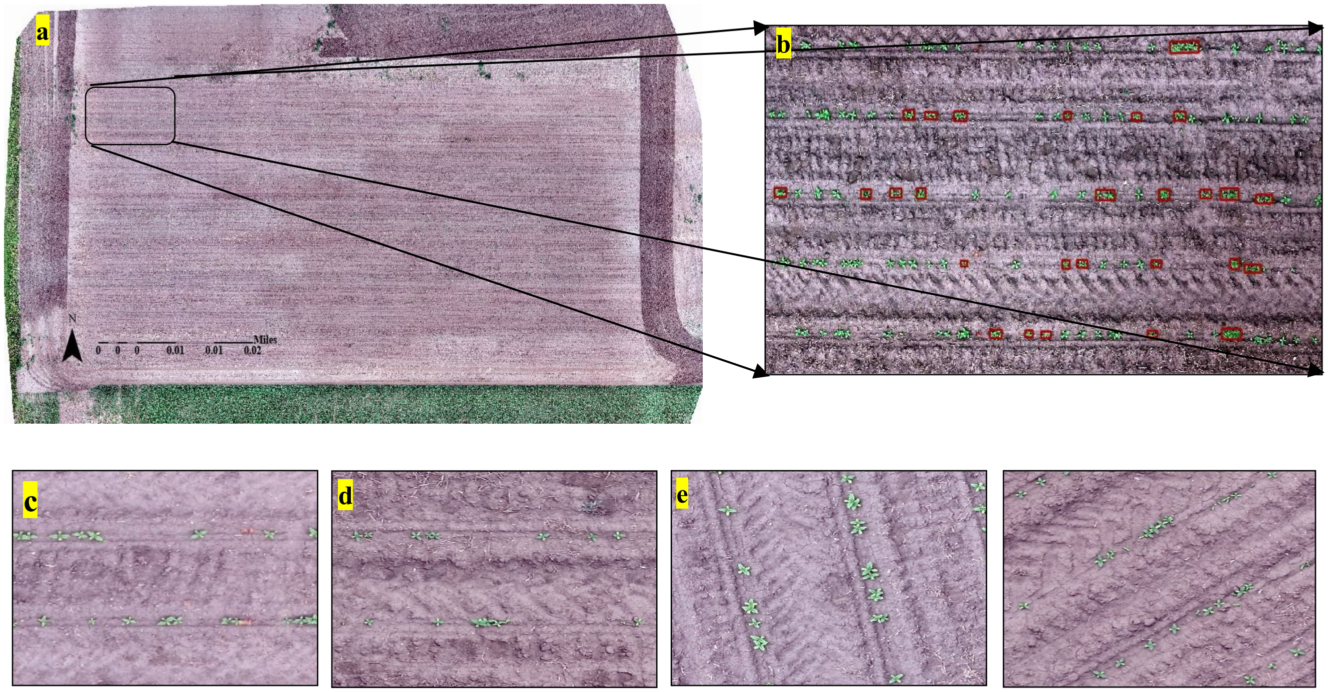

Figure 1. Represents image outputs using the function export training data for deep learning on Python platform. (a) original orthomosaic, (b) bounding box using hand annotation created using Training Samples Manager (red rectangle - doubles), c) image chips obtained for training purpose, (e) augmented sunflower crop images at varying angles.

Training dataset and hyperparameter tuning

Training a massive dataset for deep learning applications is a computationally hefty task. But when trained with adequate amount of data and proper fine tuning of hyperparameters leads to better detection accuracy. For our work, four CNN architectures were chosen, namely, single short detector (SSD), You Look Only Once (YOLOv3), RetinaNet, and Faster Regions with Convolutional Neural Network (FasterRCNN). ResNet-50 was selected as a backbone model with SSD, FasterRCNN, and RetinaNet to serve as a feature extractor for our input dataset, whereas, DarkNet-53 was chosen with YOLOv3 because of only available backbone model to be used alongside YOLOV3 in ArcGIS Pro.

These four architectures were chosen mainly for three reasons; 1) they are the best object detection models available within the latest ArcGIS Pro (v 2.7), and, 2) based on past successful applications in plant detection applications in agricultural domain (Giuffrida et al., 2018, Mosley et al., 2020., Miao et al., 2021, Rabbi et al., 2020, Sungchan et al., 2020), and 3) to choose best model for sunflower doubles detection. The final augmented dataset was trained on a Nvidia RTX2080 Super (8GB) GPU with a parallel processing factor of 50% resulting in usage of half GPU cores (3072 in total). Training on the GPU was executed in order to perform high-throughput computing to deliver low latency for training and detecting tasks.

An organized flowchart (fig. 2) was employed when training our dataset. To fine tune our model for detection task, a saved model (with saved weights) were used as a pretrained models against all the 4 architectures. This step led to increase in average precision accuracy of each model. An approach employed was not to train the models until a given epoch. This was done because maximum epoch sometime leads to overfit the neural networks resulting in increase in generalization error thereby making the model less useful for detecting objects on an unseen dataset. On the contrary, too little training would underfit the model. With this in mind, an early stopping criterion was added as a parameter to stop the training process as the validation loss or the generalization error increases (table 1). Batch size was set to 16 as the GPU size was not large enough to train our dataset in larger batches. After our models were trained, 3 crucial files were stored on the local drive. These files were, deep learning model package (*. dlpk), ESRI Model Definition file (*. emd), and a model characteristics folder (*. html) consisting of learning curves and average precision score.

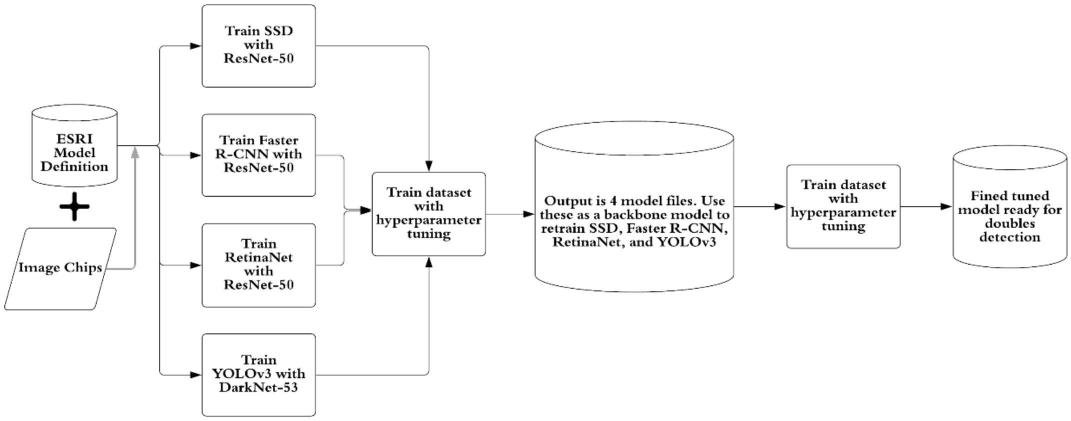

Figure 2. Flowchart represents method employed to train image chips

Table 1. Training time for sunflower crop dataset

Models	Given Epoch	Final Epoch	Learning Rate	Training Time (hr: min: sec)
SSD + ResNet50	100	12	0.001	00:58:19
SSD + Pretrained Model	50	9	0.001	3:51:26
RetinaNet + ResNet50	100	8	0.001	1:44:24
RetinaNet + Pretrained Model	50	8	0.001	2:16:28
FasterRCNN + ResNet50	100	8	0.001	3:48:32
FasterRCNN + Pretrained Model	50	12	0.001	6:58:15
YOLOv3 + DarkNet-53	100	40	0.001	12:33:13
YOLOv3 + Pretrained Model	50	50	0.001	13:13:43

Detecting sunflower doubles

The imagery was categorized into 2 parts for detecting doubles. The first part comprised of clipped raster patches of sunflower doubles (fig. 5a&b), while the second part was a clipped raster involving weeds (fig. 5c) and pebbles (fig. 5d). A raster with a patch of weeds was included to verify that our model was not detecting weeds as doubles. In general, these categories comprised of area not included in the training dataset (or unseen dataset). Ground truth data was generated by manually digitizing doubles using the Training Samples Manager tool in ArcGIS Pro. This step was done in order to compute metrics to determine model accuracy for detection task. The trained deep learning model was deployed on the ground truth data to detect sunflower doubles. Two arguments, namely, confidence threshold and nonmaximum suppression were tweaked to optimize the task of detecting sunflower doubles. We set different confidence threshold values to adjust the sensitivity of our model. Setting up the threshold values were dependent on the confidence accuracy with which doubles were detected by our models. If a high threshold value was given, shapefiles for all the objects above that value was automatically drawn. Therefore, a lower threshold value resulted in optimum detection of doubles in the imagery. On the other hand, nomaximum suppression was set to none because it included duplicate objects (doubles) during object detection task.

Metrics to compute detection accuracy

To compute the accuracy of our models, we calculated a common metrics (table 2) which correlated the predicted bounding-box with the ground truth data. This metric number was the result of an average precision (AP) scores based on 3 variations, 1) True Positive (TP, fig. 3c) was when the model correctly predicted that there was a double, 2) False Positive (FP, fig. 3d) was when the model incorrectly predicted that there was a double, and lastly, 3) False Negative (FN, fig. 3e) was when the model incorrectly predicted that there was no double, and 4) True Negative (TN) was when the model correctly predicted that there was no double (either soil background, weeds, other objects). Since our objective was to detect doubles, we discarded the last variation. Further, to combine our metrics based on 3 variations for computing detection accuracy of our models, a measurement based on Jaccard Index, called as, intersection over union (IoU) (fig. 3a) was calculated along with, precision in Eq. (1), recall in Eq. (2), and F1 score in Eq. (3). IoU is a very common metrics evaluator specifically used to measure the accuracy of an object detection model. In the figure below, the formula with the schematic shows how this method is useful in evaluating the accuracy of our model.

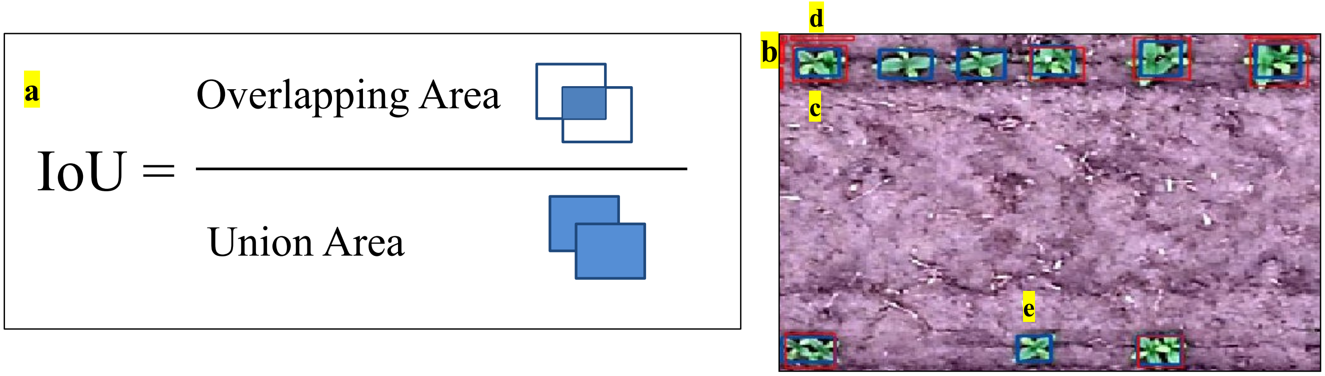

Fig 3. Schematic for metrics evaluation . (a) Intersection over Union or Jaccard Index, (b) Image patch with depiection of metrics variation, (c) TP (blue rectangle - ground truth, red rectangle – predicted box), (d) FP (only red rectangle – predicted box), (e) FN (only blue rectangle – ground truth)

$$\text{Precision (P)} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP) + False Positive (FP)}} \tag{1}$$

$$\text{Recall (R)} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP) + False Negative (FN)}} \tag{2}$$

$$\text{F1 score} = \frac{\text{2 x Precision (P) x Recall (R)}}{\text{Precision (P) + Recall (R)}} \tag{3}$$

Results and discussions

This section reports about the evaluation metrics which compares ground truth (human-vision) with computer vision approach. A variation of metrics has been generated to compute detection accuracy. We evaluate model performance, compare average precision accuracy between the models, and finally recommend which model should be used for detecting sunflower doubles in ArcGIS Pro. Furthermore, we also add training and validation loss graphs which has been categorized as underfit, overfit or a good fit model.

Model performance

Based on the table below (table 3), fine-tuned model when used as a pretrained model alongside Faster R-CNN resulted in a highest average precision accuracy value of 99.6%. A combination of ResNet-50 with RetinaNet also resulted in an average precision accuracy of 84.6%. YOLOv3 was able to fetch 81% overall precision accuracy. SSD on the other hand, performed poorly with an overall precision accuracy of 23%, therefore we discarded this model to perform doubles detection. Table 4, reports the average metrics calculated on 5 ground truth data. As mentioned earlier, the ground truth data consisted of other objects such as, weeds, pebbles, and ground control points.

Table 3. Average precision accuracy for 4 models

Models	Class	Average precision accuracy
Faster R-CNN	doubles	99.6%
RetinaNet	doubles	93.6%
YOLOv3	doubles	87.8%
SSD	doubles	22.54%

Table 4. Double detection metrics for 4 models

Models	F1 score	Precision	Recall
Faster R-CNN	0.95	0.91	0.96
RetinaNet	0.93	0.92	0.96
YOLOv3	0.90	0.86	0.89
SSD	0.47	0.52	0.45

Assessing model performance based on learning curves

For the training and validation loss graph, we provide 4 learning curves which can be used as a diagnostic tool to check learning and generalization behavior of the developed models. Learning curve consists of training curve which gives an idea about how well the model is learning, whereas validation curve describes about how well the model is generalizing on the unseen dataset. These 4 learning curves are extracted from the fine-tuned models deployed on the imagery dataset for detection purpose. Instead of setting up early stopping process, the trained models showed variations in learning curves based on which we divided them into underfit, overfit, and good fit models. Underfitting occurs when the model fails to learn from a given dataset resulting in a flat learning curve. Overfitting occurs when the model learns noise and random fluctuations from the dataset resulting in increase and decrease validation losses. A good fit model on the other hand, is identified with a stable training and validation loss. The curve keeps on decreasing where it ultimately becomes stable with a small gap with the validation loss. From the graphs extracted from the fine-tuned models after training the dataset, we observed that Faster R-CNN (fig. 4a) and YOLOv3 (fig. 4d) comes under the category of a good fit curve. This is the result that these models delivered a high accuracy when detecting sunflower doubles on the imagery. RetinaNet (fig. 4c) can be categorized as an overfit model where minimal fluctuations could be seen in the validation loss. The learning curve for SSD (fig. 4b) was an underfit curve where the training data was not sufficient enough to provide information for the model to learn. This situation is termed as unrepresentative training dataset where a huge gap between the learning curve and validation curve can be seen. Based on the accuracy obtained for SSD and after diagnosing model's performance using learning curve, we do not recommend SSD model to detect or count sunflower stands. We recommend resampling sunflower plants dataset based on k-fold cross validation method where the model is trained or tested k-times.

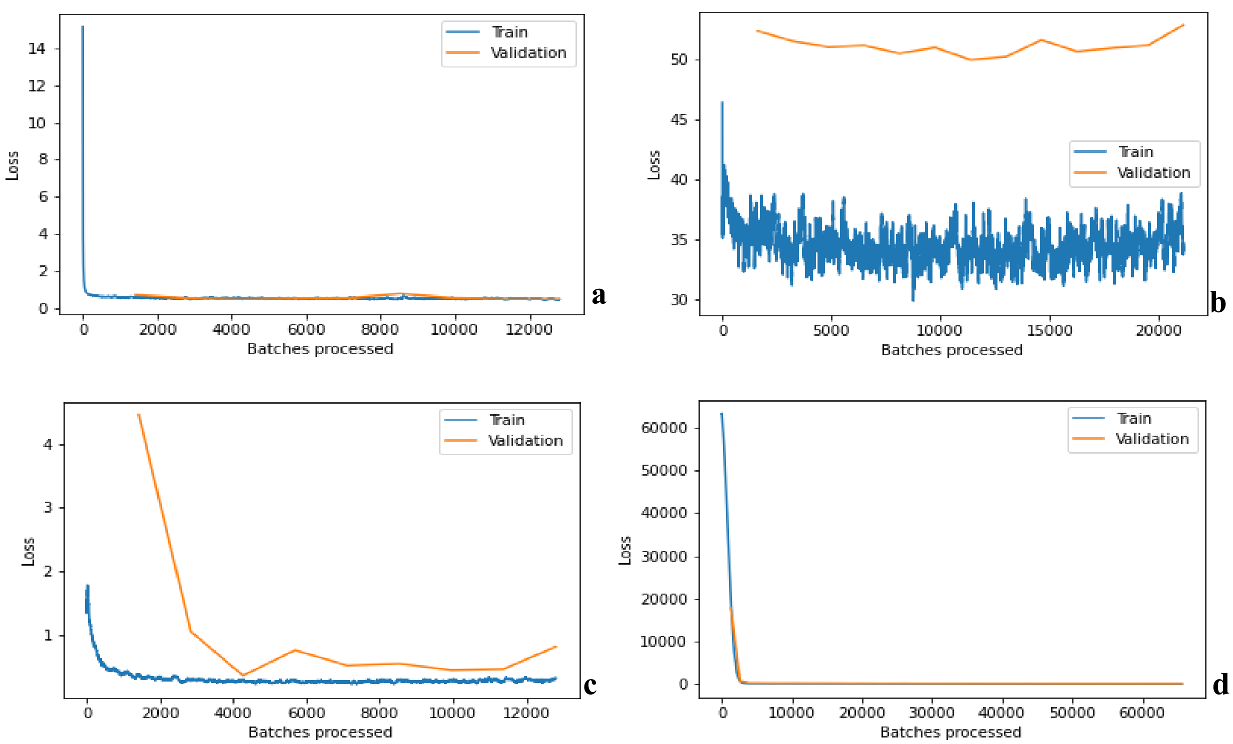

Figure 4. Learning curves for 4 architectures; (a) Faster R-CNN, (b) SSD, (c) RetinaNet, and (d) YOLOv3

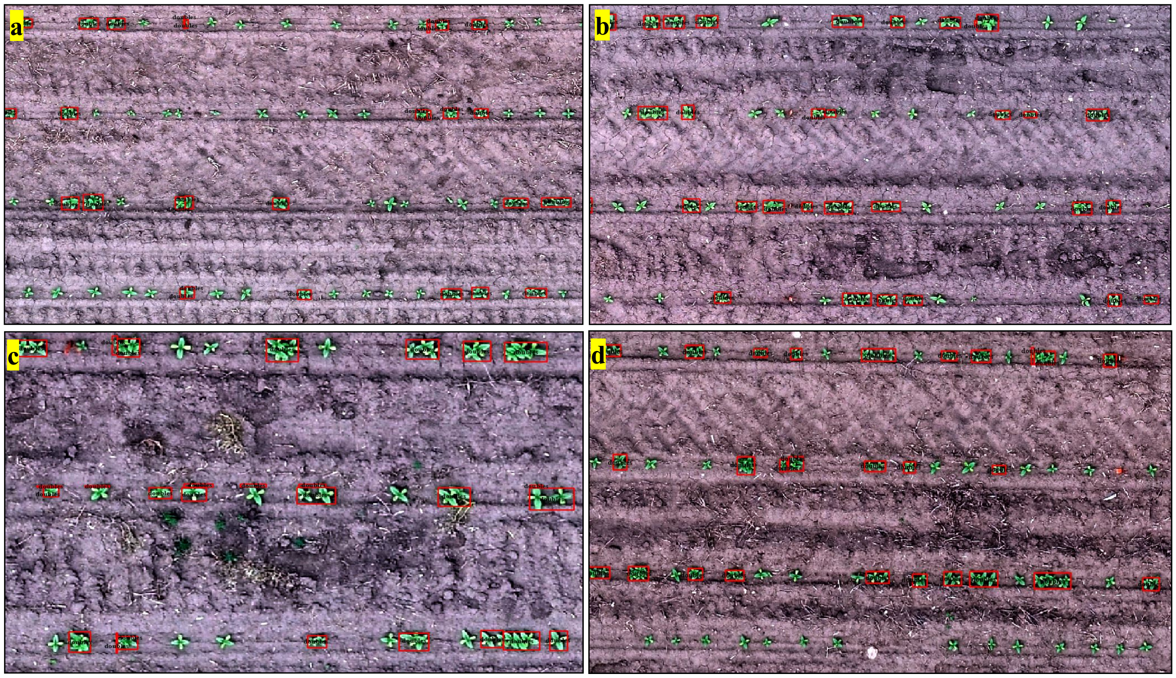

Fig 5. Snips of detected sunflower doubles. (a) Ground truth data (red – detected doubles), (b) ground truth data, (c) ground truth data with weeds, (d) ground truth with pebbles

Conclusions

This work has presented a workflow to detect sunflower doubles in ArcGIS Pro. An effective combination of computer vision and transfer learning has been leveraged to address the task of detecting and visualizing doubles in a high resolution UASs imagery. Automation of this task using Python platform has been effectively implemented eliminating the need to open the software using GUI.

While there were previous studies that addressed doubles detection, to the best of our knowledge, no work has been reported that has combined transfer learning in ArcGIS Pro specifically for detecting sunflower doubles thereby addressing the issue of automating shapefile generation for individual crops. The work adds to the growing corpus of research showing that the models we have developed can be utilized by other users to train their dataset and deploy on their high-resolution imagery to detect sunflower doubles. Collectively, our results appear consistent with the average accuracy obtained when Faster R-CNN model was trained on our dataset.

Future Work

As a part of our future workflow, we are working on adding a separate class for singles to the training dataset. Our plan is to fully automatize the task of sunflower stand counts and doubles detection in ArcGIS Pro. Besides, we are also working on adding a feature that would estimate crop spacing. We also reckon to add weeds as a separate class for detection, which can be used as the basis to create weed control prescription maps.

References

ESRI. (2020). Where Deep Learning Meets GIS. Retrieved from: https://www.esri.com/about/newsroom/arcwatch/where-deep-learning-meets-gis/

F. Zhuang. (2021). A Comprehensive Survey on Transfer Learning, in *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43-76, doi: 10.1109/JPROC.2020.3004555.

Holman, F.H.; Riche, A.B.; Michalski, A.; Castle, M.; Wooster, M.J.; Hawkesford, M.J. (2016). High Throughput Field Phenotyping of Wheat Plant Height and Growth Rate in Field Plot Trials Using UAV Based Remote Sensing. *Remote Sens*, *8*, 1031. https://doi.org/10.3390/rs8121031.

Nowakowski. A, Mrziglod. J, Spiller. D, Bonifacio. R, Ferrari. I, Mathieu. R. P, Garcia-Herranz. M, Kim. D. (2021). Crop type mapping by using transfer learning, International Journal of Applied Earth Observation and Geoinformation, Volume 98, 102313, ISSN 0303-2434, https://doi.org/10.1016/j.jag.2021.102313.

Radanović A, Miladinović D, Cvejić S, Jocković M, Jocić S. (2018). Sunflower Genetics from Ancestors to Modern Hybrids-A Review. *Genes (Basel)*. 2018; 9(11):528. doi:10.3390/genes9110528.

Wang L, Xiang L, Tang L, Jiang H. A. (2021). Convolutional Neural Network-Based Method for Corn Stand Counting in the Field. *Sensors (Basel)*. 2021; 21(2): 507. doi:10.3390/s21020507.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu. (2015). SSD: Single Shot MultiBox Detector; arXiv:1512.02325.