Weed Science

www.cambridge.org/wsc

Use of open-source object detection algorithms to detect Palmer amaranth (*Amaranthus palmeri*) in soybean

Isaac H. Barnhart[1], Sarah Lancaster[2], Douglas Goodin[3], Jess Spotanski[4] and J. Anita Dille[5]

[1]Graduate Research Assistant, Department of Agronomy, Kansas State University, Manhattan, KS, USA; [2]Assistant Professor, Extension Weed Specialist, Department of Agronomy, Kansas State University, Manhattan, KS, USA; [3]Professor, Department of Geography and Geospatial Sciences, Kansas State University, Manhattan, KS, USA; [4]Owner/Manager, Midwest Research Incorporated, York, NE, USA and [5]Professor, Department of Agronomy, Kansas State University, Manhattan, KS, USA

Research Article

Cite this article: Barnhart IH, Lancaster S, Goodin D, Spotanski J, Dille JA (2022) Use of open-source object detection algorithms to detect Palmer amaranth (*Amaranthus palmeri*) in soybean. Weed Sci. **70**: 648–662. doi: 10.1017/wsc.2022.53

Received: 7 June 2022
Revised: 30 August 2022
Accepted: 9 September 2022
First published online: 19 September 2022

Associate Editor:
Ramon G. Leon, North Carolina State University

Keywords:
Artificial intelligence; TensorFlow; YOLOv5; Faster R-CNN; Single Shot Detector; site-specific weed management.

Author for correspondence:
J. Anita Dille, Kansas State University, Agronomy Department, 1712 Claflin Road, Manhattan, KS 66506-5501.
Email: dieleman@ksu.edu

Abstract

Site-specific weed management using open-source object detection algorithms could accurately detect weeds in cropping systems. We investigated the use of object detection algorithms to detect Palmer amaranth (*Amaranthus palmeri* S. Watson) in soybean [*Glycine max* (L.) Merr.]. The objectives were to (1) develop an annotated image database of *A. palmeri* and soybean to fine-tune object detection algorithms, (2) compare effectiveness of multiple open-source algorithms in detecting *A. palmeri*, and (3) evaluate the relationship between *A. palmeri* growth features and *A. palmeri* detection ability. Soybean field sites were established in Manhattan, KS, and Gypsum, KS, with natural populations of *A. palmeri*. A total of 1,108 and 392 images were taken aerially and at ground level, respectively, between May 27 and July 27, 2021. After image annotation, a total of 4,492 images were selected. Annotated images were used to fine-tune open-source faster regional convolutional (Faster R-CNN) and single-shot detector (SSD) algorithms using a Resnet backbone, as well as the "You Only Look Once" (YOLO) series algorithms. Results demonstrated that YOLO v. 5 achieved the highest mean average precision score of 0.77. For both *A. palmeri* and soybean detections within this algorithm, the highest F1 score was 0.72 when using a confidence threshold of 0.298. A lower confidence threshold of 0.15 increased the likelihood of species detection, but also increased the likelihood of false-positive detections. The trained YOLOv5 data set was used to identify *A. palmeri* in a data set paired with measured growth features. Linear regression models predicted that as *A. palmeri* densities increased and as *A. palmeri* height increased, precision, recall, and F1 scores of algorithms would decrease. We conclude that open-source algorithms such as YOLOv5 show great potential in detecting *A. palmeri* in soybean-cropping systems.

Introduction

Site-specific weed management (SSWM) involves adapting weed management strategies to match weed variation within a given field (Fernández-Quintanilla et al. 2018). In agriculture, weeds often grow in distinct patches rather than uniformly across a field (Maxwell and Luschei 2005); as a result, broadcast herbicide applications often treat areas of the field where no weeds are present. In theory, using SSWM could result in increased herbicide savings, decreased herbicide expenses, and decreased environmental contamination (Arsenijevic et al. 2021; Barroso et al. 2004; dos Santos Ferreira et al. 2019). An additional benefit is that SSWM could allow for the economical application of multiple herbicide mechanisms of action (MOAs) so that more expensive chemistries could be applied only where needed. Not only would this be less expensive for the farmer, but applications with diversified MOAs help to slow the development of herbicide-resistant weeds (Evans et al. 2015). In modern agriculture, successful weed control can be very difficult due to increases in herbicide-resistant weed cases, rising costs of herbicides, and shortages of crop protection products brought on by economic consequences of the COVID-19 pandemic (Dayan 2021; Mordor Intelligence 2022). As such, strategies such as SSWM that aim to reduce input quantities and costs could potentially benefit farmers and contribute to the sustainability of cropping systems around the world (Bongiovanni and Lowenberg-Deboer 2004).

The key component to SSWM involves the accurate detection of weed positions within a given field, but the development of a robust and accurate detection system for field conditions remains a challenge (Gao et al. 2020). One of the ways this challenge is being addressed is by applying artificial intelligence using convolutional neural networks (CNNs). Convolutional neural networks are a type of deep neural network that excel at pattern recognition and can

© The Author(s), 2022. Published by Cambridge University Press on behalf of the Weed Science Society of America. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

be utilized in a variety of tasks ranging from image analysis to audio file analysis (Albawi et al. 2017). The most common use of CNNs in the agricultural sector involves image analysis; CNNs analyze the textural, spectral, and spatial features of images and can extract features unseen by the human eye (Albawi et al. 2017; Sapkota et al. 2020). Fruit counting, weed detection, disease detection, and grain yield estimation are ways that CNNs have been used in agriculture (Biffi et al. 2021; Hussain et al. 2020, 2021; Sivakumar et al. 2020; Yang et al. 2019).

Two approaches to documenting and treating weeds in the field are real-time in situ weed detection and herbicide application and scouting and developing weed maps to guide SSWM (Cardina et al. 1997; Somerville et al. 2020). In situ weed detection involves recognizing weeds in real time and can lead to plants being treated in a timelier manner. Platforms that have been developed to detect weeds in situ include "smart" sprayers, autonomous weeding robots, and unmanned aerial spraying vehicles (Sivakumar et al. 2020), most of which use some type of CNN technology. CNNs have been shown to be accurate in tasks such as segmentation (dividing images into regions based on pixel similarities), image classification (assigning a label to an image based on the objects present), and object detection (identifying objects within an image) (Biffi et al. 2021; Sivakumar et al. 2020; Stanford 2022; ThinkAutomation 2022). Object detection CNNs are typically at the forefront of in situ weed detection, as there is often greater value in detecting and localizing agricultural pests as opposed to assigning labels to images with pests located in them (Chen et al. 2021).

In recent years, open-source object detection algorithms have become available, such as those from the TensorFlow Object Detection API (Huang et al. 2017), the "You Only Look Once" (YOLO) algorithm series (first introduced by Redmon et al. 2016), and the Detectron algorithm series (Lin et al. 2018). These object detectors have been used to implement a variety of computer vision tasks, including cancer cell detection (Al Zorgani et al. 2022), facial recognition (Mattman and Zhang 2019), underwater fish detection (Xu and Matzner 2018), and projects related to the development of self-driving vehicles (Kulkarni et al. 2018). Open-source algorithms are typically pretrained on very large data sets, such as the Microsoft COCO (Common Objects in Context) data set (Lin et al. 2014). Through utilizing a process called transfer learning, pretrained algorithm parameters can be fine-tuned to detect custom objects. Transfer learning involves using information learned from one object detection algorithm and applying this information to identify different, yet related, objects (Ghazi et al. 2017). This eliminates the need to train algorithms from scratch, which is a very computationally expensive and time-consuming process (Ruder 2021). Open-source algorithms fine-tuned to identify agricultural crops and weeds have been used in a variety of studies, including late-season species detection in soybean [*Glycine max* (L.) Merr.] of Palmer amaranth (*Amaranthus palmeri* S. Watson), waterhemp (*Amaranthus tuberculatus* (Moq.) Sauer], common lambsquarters (*Chenopodium album* L.), velvetleaf (*Abutilon theophrasti* Medik.), and *Setaria* spp. (Sivakumar et al. 2020); detection of wild radish (*Raphanus raphanistrum* L.) and capeweed [*Arctotheca calendula* (L.) Levyns] in barley (*Hordeum vulgare* L.) (Thanh Le et al. 2021); and weed detection in a variety of crops, including lettuce (*Lactuca sativa* L.) (Osorio et al. 2020), carrots (*Daucus carota* L. var. *sativus* Hoffm.) (Ying et al. 2021), corn (*Zea mays* L.) (Ahmad et al. 2021), and onions (*Allium cepa* L.) (Parico and Ahamed 2020).

For this study, we focused on detecting *A. palmeri* in soybean-cropping systems using open-source object detection algorithms. *Amaranthus palmeri* has been designated the most problematic weed in the United States (WSSA 2016), and it can reduce soybean yields by as much as 68% (Klingaman and Oliver 1994; Kumar et al. 2021). Therefore, controlling this weed is very important for United States soybean producers. Large numbers of training images are necessary to train object detection algorithms to identify custom objects (Pokhrel 2020); however, nonproprietary image databases of *A. palmeri* are often unavailable. In addition, even though algorithms have been previously trained on *A. palmeri* in the midwestern United States (Sivakumar et al. 2020), we did not find many studies that investigated the relationship of model evaluation metrics as influenced by *A. palmeri* growth features, including canopy diameter, plant height, percent ground cover, and weed density. Understanding the relationship between algorithm evaluation metrics and *A. palmeri* growth features could benefit precision weed applications. For example, future databases could focus on collecting images of *A. palmeri* plants with growth features best detected by the algorithm. In addition, farmers and agricultural professionals could gain a better understanding of which field conditions would benefit the most from deploying these algorithms. For instance, *A. palmeri* infestations with large plants and high populations may not be the best environments to use this technology for site-specific applications.

We hypothesized that as weed diameter and height increase, object detection algorithms will be better able to identify *A. palmeri* plants; however, as *A. palmeri* density and ground cover increases, ability to identify will decrease. Object detection algorithms can have difficulty both detecting small objects (Li et al. 2017) and detecting all object occurrences if objects are present in high densities in an image (Sun et al. 2022). The specific objectives of this study were (1) to develop an annotated image database of *A. palmeri* and soybean with multiple weed densities and soybean row spacings that can be used to fine-tune object detection algorithms, (2) compare multiple open-source algorithms' effectiveness in detecting *A. palmeri*, and (3) evaluate the relationship between *A. palmeri* growth features (diameter, height, density, and ground cover) and *A. palmeri* detection ability.

Materials and Methods

Image Acquisition

To establish conditions representative of multiple *A. palmeri* densities and soybean-cropping systems, field locations were identified at the Kansas State University, Department of Agronomy Ashland Bottoms Research Farm near Manhattan, KS (39.122°N, 96.635° W) and at the Lund Research Farm near Gypsum, KS (38.797° N, 97.448°W) in 2021. At each location, 24 plots of soybeans were planted at a seeding rate of 331,000 seed ha^{-1}: 12 plots were planted at 38-cm-wide row spacing, and 12 plots were planted at 76-cm-wide row spacing; plot dimensions were 3.1-m wide and 9.1-m long. Both field sites had a naturally occurring population of *A. palmeri* that was allowed to germinate and grow with the soybeans. These field locations allowed multiple densities of *A. palmeri* to be photographed while growing among soybean in different row spacings, providing a greater diversity of field situations to be "seen" by each algorithm.

The training database was built with 1,500 images taken of *A. palmeri* only, soybean only, or both species between May 27 and July 27 (Table 1). Imagery was taken both with a TG-610

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

handheld camera (OM Digital Solutions, Hachioji-City, Tokyo, Japan), and with a DJI Inspire 1 unmanned aerial vehicle (UAV) equipped with a Zenmuse X5R RAW camera (DJI, Shenzhen, China). The TG-610 has a sensor size of 28 mm^2, whereas the Zenmuse X5R has a larger sensor size of 225 mm^2. To increase the variability of the photographed vegetation, above-ground altitudes at which the images were taken varied from 1.5 m to 8 m, often with the minimum height chosen on any given day determined by vegetation height. For example, as the plants increased in height, it became necessary to increase the UAV flight altitude to prevent propeller downdraft from collapsing the plants. To add an additional source of variability, images were also collected under a variety of lighting conditions.

Field-collected Data

To model algorithm evaluation metrics related to *A. palmeri* growth features, plant height, canopy diameter, and density were taken weekly between the middle rows of each plot from 1 to 4 wk after planting (WAP) and 1 to 5 WAP for the Manhattan and Gypsum plots, respectively. *Amaranthus palmeri* density was measured each week in a 0.25-m^2 quadrat placed at random within these rows. A total of four height and diameter measurements were recorded from random plants within the quadrats. Within this study, *A. palmeri* height was defined as the tallest measurable structure, and diameter was defined as the widest portion of the plant within the top 20 cm. In plots with a total of fewer than four *A. palmeri* plants observed, height and diameter measurements corresponded to the total number of observable plants. Data were taken on *A. palmeri* plants after the formation of the first true leaf and any growth stage afterward, as cotyledons proved to be too difficult to annotate on the image data set. On these same dates, five photos were taken within the middle of each plot with the handheld camera, approximately 1.5 m above the canopy. Five photos were taken to provide a representative sample of the plot, as plots were 9.1 m in length. These images were kept separate from the training data set and were used to evaluate algorithm performance within each plot.

Image Processing and Data Annotation

Raw image outputs from the handheld and the Zenmuse cameras produced images with dimensions that were too large and would exceed processor memory capacity. To begin, every input image was cropped to dimensions of 2,880 × 2,880 pixels to remove the "fish-eye" effect that often accompanies aerial imagery (Gurtner et al. 2007). Next, these images were tiled into smaller dimensions of 1,024 × 1,024 pixels using Python 3.9.7 (Python Software Foundation 2022) and the Pillow module (Clark 2022). This allowed images large enough to retain features necessary for labeling, but small enough so as to not exhaust processor memory during training. Each input image was tiled into 20 new images of 1,024 × 1,024 pixels for a total of 30,000 images, which allowed for more images to be added to the training database. Images in which no plant features were visible or those of poor quality were simply discarded and not labeled; in all, 4,492 images were selected for labeling.

Images were labeled using the annotation tool LabelImg (Tzutalin 2015), which allows users to draw rectangular bounding boxes around objects within imagery and assign classes to each box. In many cases, the presence of multiple classes of objects can lead to better detection results due to the presence of multiple feature gradients (Oza and Patel 2019). Therefore, we chose to annotate two classes for this study: *A. palmeri* and soybean. Using the same methods described by Sivakumar et al. (2020), bounding boxes were drawn over both individual and patches of *A. palmeri* and soybean. In the event of irregularly shaped plants or patches of plants, multiple bounding boxes were drawn to encompass the entirety of the plant features. Labeling partial sections of irregularly shaped plants has been shown to be beneficial to object detectors (Sharpe et al. 2018, 2020a; Zhuang et al. 2022), so these irregular features were not ignored. In any given image, both plant species could be present, so they were labeled accordingly. *Amaranthus palmeri* plants were labeled at all growth stages ranging from the formation of the first true leaf through inflorescence, and soybeans were labeled from the VE-VC stage through the R2 stage (Fehr et al. 1971) (Figure 1). *Amaranthus palmeri* plants visible within each labeled image ranged from no plants (only soybean labeled) to roughly 115 labeled specimens. Throughout the image data set, soybean growth stage was homogeneous on each plant date, but *A. palmeri* could have multiple growth stages visible due to its ability to emerge throughout the growing season (Jha and Norsworthy 2009; Shyam et al. 2021). Figure 2 illustrates the image labeling process.

The images that were selected for labeling contained a total of 10,494 and 10,312 *A. palmeri* and soybean annotations across all growth stages, respectively. The data set was then divided into 90% training and 10% test images, used to train and evaluate the algorithms, respectively. The training and test data sets consisted of 4,042 and 450 images, respectively. The aforementioned images taken over the plots for analysis of *A. palmeri* growth features and algorithm evaluation metrics were not included in the training or testing data sets but were kept separate for further analysis.

Table 1. Dates, number of images, platform used, and height above ground for image collection at Manhattan and Gypsum, KS, field locations in 2021.

Date	Images	Platform[a]	Height above ground	Image dimensions[b]
Manhattan				
	no.		m	
May 27, 2021	35	UAV	1.5	A
June 1, 2021	90	UAV	0.5–1.5	A
June 9, 2021	17	UAV	2.8	A
June 14, 2021	44	UAV	2.0–3.0	A
June 15, 2021	14	HH	2	B
June 17, 2021	72	HH	1.2–1.5	B
June 22, 2021	150	HH	1.2–1.5	B
July 1, 2021	125	HH	1.2–1.8	B
July 8, 2021	80	HH	1.2–2.3	B
July 9, 2021	33	UAV	3.0–8.0	A
July 19, 2021	80	HH	2.3	B
July 19, 2021	61	UAV	7.0–8.0	A
July 26, 2021	40	HH	2.3	B
Gypsum				
June 21, 2021	80	HH	1.2–1.5	B
June 29, 2021	135	HH	1.2–1.5	B
July 5, 2021	80	HH	1.2–1.8	B
July 12, 2021	90	HH	1.2–1.5	B
July 12, 2021	68	UAV	6.0–7.0	A
July 20, 2021	80	HH	1.8	B
July 27, 2021	82	HH	2.3	B
July 27, 2021	44	UAV	7.0–8.0	A

[a]HH, handheld Olympus TG-510 digital camera; UAV, unmanned aerial vehicle with Zenmuse camera.
[b]Total pixel dimensions per image: A = 4,608 × 3,456; B = 4,288 × 3,216.

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

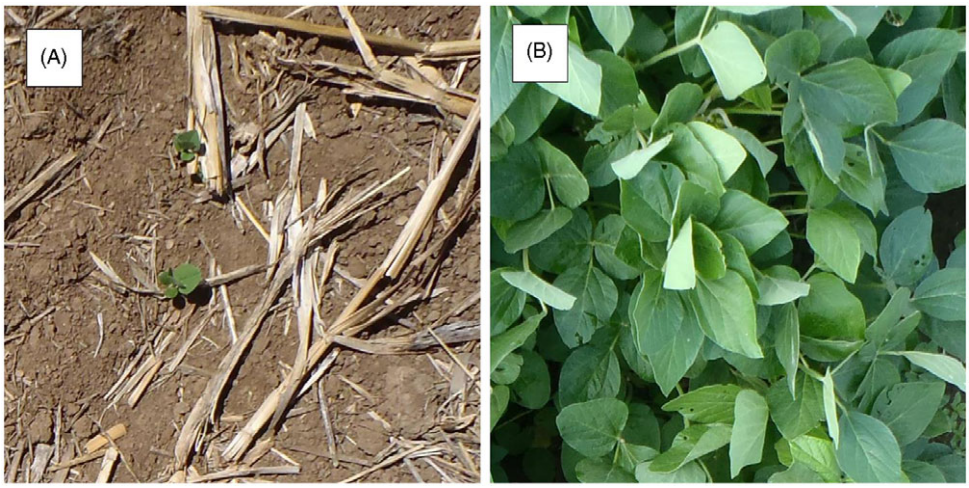


Figure 1. Examples of images collected for soybean in the VE-VC (A) and R2 (B) growth stages.

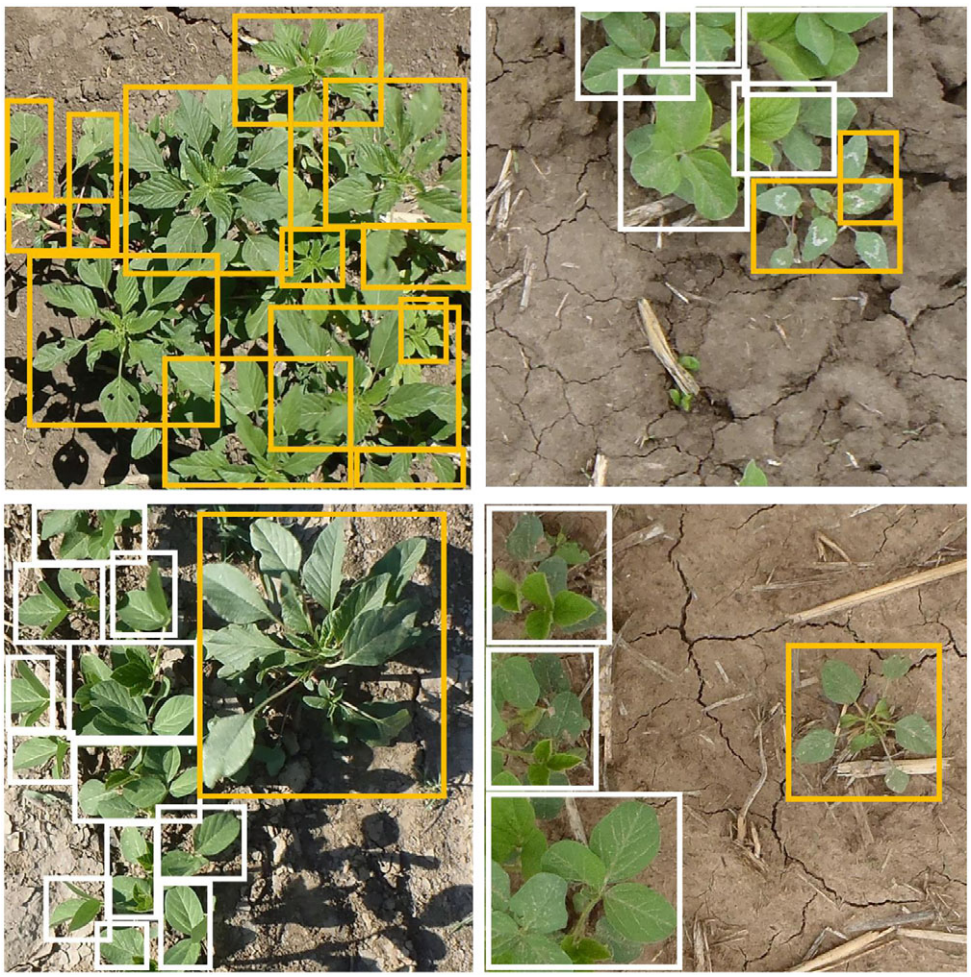


Figure 2. Illustration of the annotation process. *Amaranthus palmeri* and soybean plants are labeled in this figure with orange and white boxes, respectively. Bounding boxes overlap with neighboring bounding boxes when plant features are irregular. In cases where a single bounding box could not encompass a plant without including a plant of another species, multiple irregular bounding boxes were drawn on a single specimen.

Algorithm Selection

Three open-source algorithm architectures were used in this study: faster regional convolutional neural network (Faster R-CNN), single-shot detector (SSD), and two YOLO algorithms. Faster R-CNN algorithms are a modern development of what are called regional-based CNNs, first proposed by Girshick et al. (2014). This approach was revolutionary, in that it was one of the first large-scale successful approaches to addressing the task of object

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

Table 2. Training information and hyperparameters used in this study.

Architecture[a]	Backbone model	Batch size	Training interval	Initial learning rate	Learning rate policy
Faster R-CNN	ResNet-50	2	100,000 steps	0.01	Cosine decay
Faster R-CNN	ResNet-101	2	100,000 steps	0.0001	Cosine decay
Faster R-CNN	ResNet-152	1	100,000 steps	0.0001	Cosine decay
Faster R-CNN	Inception ResNet-V2	1	182,500 steps	0.001	Cosine decay
SSD	ResNet-50	4	100,000 steps	0.001	Cosine decay
SSD	ResNet-152	1	100,000 steps	0.0001	Cosine decay
YOLOv4	CSPDarknet53	1	6,000 iterations	0.001	Cosine decay
YOLOv5	CSPDarknet53	4	41 epochs	0.01	Cosine decay

[a]Faster R-CNN, faster regional convolutional neural network; SSD, single-shot detector; YOLO, "You Only Look Once."

localization and detection (Balasubramanian 2021). As input images are fed into the algorithm, areas of interest based on groups of pixels are extracted from the image and fed into the neural network (Oinar 2021). The architecture has been updated with the development of Fast R-CNN in 2015 (Girshick 2015) and finally with Faster R-CNN in 2016 (Ren et al. 2017), with each version being faster in detection speed than the previous. Faster R-CNN is known as a two-stage object detector, in that it first extracts regions of interest where it is likely that the objects will be and then classifies these regions of interest (Du et al. 2020). Consequently, Faster R-CNN is known to perform better in terms of detection accuracy but has slower detection speeds (Sivakumar et al. 2020).

SSD and YOLO algorithms were proposed by Liu et al. (2016) and Redmon et al. (2016), respectively. Considered single-stage object detectors, they are generally faster and less computationally expensive than Faster R-CNN algorithms (Liu et al. 2016), allowing for faster detection and suitable for real-time detection applications. Instead of extracting regions of interest as R-CNN algorithms do, they accomplish object localization and classification in one forward pass of the neural network (Forson 2017). As in Sivakumar et al. (2020), Faster R-CNN was chosen for this project due to its detection performance, whereas SSD and YOLO algorithms were chosen due to their inference speeds.

The backbone models refer to the specific neural networks behind the architectures and allow for feature extraction from the input image (Shimao 2019). These networks are interchangeable, with multiple networks able to be used as backbone models (Li et al. 2020). For this study, the Faster R-CNN architecture backed with the ResNet (He et al. 2016) network was chosen with multiple layers, including ResNet-50, ResNet-101, ResNet-152, and Inception ResNet-V2 (Szegedy et al. 2017). Additionally, ResNet-50 and ResNet-152 were also chosen as backbone models for the SSD architecture.

For the YOLO algorithms, YOLOv4 (Bochkovskiy et al. 2020) and YOLOv5 (Jocher et al. 2020) were used, both running on the Cross Stage Partial (CSP) Darknet53 (Bochkovskiy et al. 2020) network. YOLOv5 also implements a Path Aggregation Network, allowing for both increased propagation of lower-level features and improvements in using localization signals (Carlos and Ulson 2021). This allows for an increase in accuracy when localizing an object (Carlos and Ulson 2021). Additionally, the YOLOv5 algorithm consists of four releases: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x (ultralytics 2022b). YOLOv5x was chosen for this experiment, as it is considered the most accurate object detector of the four (Carlos and Ulson 2021). Both YOLO algorithms were obtained from their respective GitHub repositories (Alexey 2022; Jocher 2022a). YOLO algorithms down-sample input images by a factor of 32 when training (Hui 2018), so input images with width and height dimensions divisible by 32 are necessary. Our input image dimensions of 1,024 × 1,024 pixels fit this criterion. All Faster R-CNN and SSD algorithms were obtained from the TensorFlow 2 Detection Model Zoo (TensorFlow 2021); the respective algorithms with input dimension requirements of 1,024 × 1,024 pixels were chosen. All YOLO and TensorFlow models selected were pretrained, thus eliminating the need to train from scratch (Ruder 2021).

Training

All algorithms were trained on a virtual Ubuntu 18.04 computer available on Paperspace, a virtual machine learning platform (Paperspace Cloud Computing, https://www.paperspace.com). The computer was equipped with an Intel® Xeon® E5-2623 v4 processor (Intel Technologies, Santa Clara, CA, USA) equipped with 16 CPU cores and 60 GB of RAM. To increase training speed, training was done utilizing a NVIDIA P6000 Graphics Processing Unit (GPU) with 24 GB of RAM (NVIDIA, Santa Clara, CA, USA).

For all algorithms, the default training hyperparameters were accepted, except for the learning rates for both the TensorFlow algorithms and batch sizes for all algorithms. As algorithm loss was monitored during training, learning rate had to be lowered below default settings for most Faster R-CNN and SSD algorithms due to an issue with exploding gradients. When doing so, we also lowered the warm-up learning rate to a value below the learning rate to avoid errors during training. As input images were large, batch sizes were reduced to prevent exhausting the GPU's memory capacity. Larger batch sizes were possible with smaller algorithms (i.e., Faster R-CNN ResNet-52 and Faster R-CNN ResNet-101), but batch sizes had to be reduced for larger algorithms (i.e., Faster R-CNN Inception ResNet-V2) to avoid the ResourceExhaustedError (TensorFlow 2017) indicating that the GPU was out of memory. Because batch size has been said to not be a significant factor in affecting algorithm performance (Ghazi et al. 2017), we did not expect this to affect the outcome of our algorithms. Algorithm training information is presented in Table 2.

During the image annotation process, all annotations were saved in Pascal VOC format. Pascal VOC format is compatible with TensorFlow algorithms, but not with YOLO algorithms. Therefore, before training the YOLO algorithms, copies of the annotations were saved in a separate folder and converted to YOLO format using Python 3.9.7. The script that was used can be obtained on the Convert PascalVOC Annotations to YOLO GitHub website (vdalv 2017).

All algorithms were trained for the default 100,000 steps, except for the Faster R-CNN Inception Resnet-V2, YOLOv4, and YOLOv5 algorithms. The default number of training steps for

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

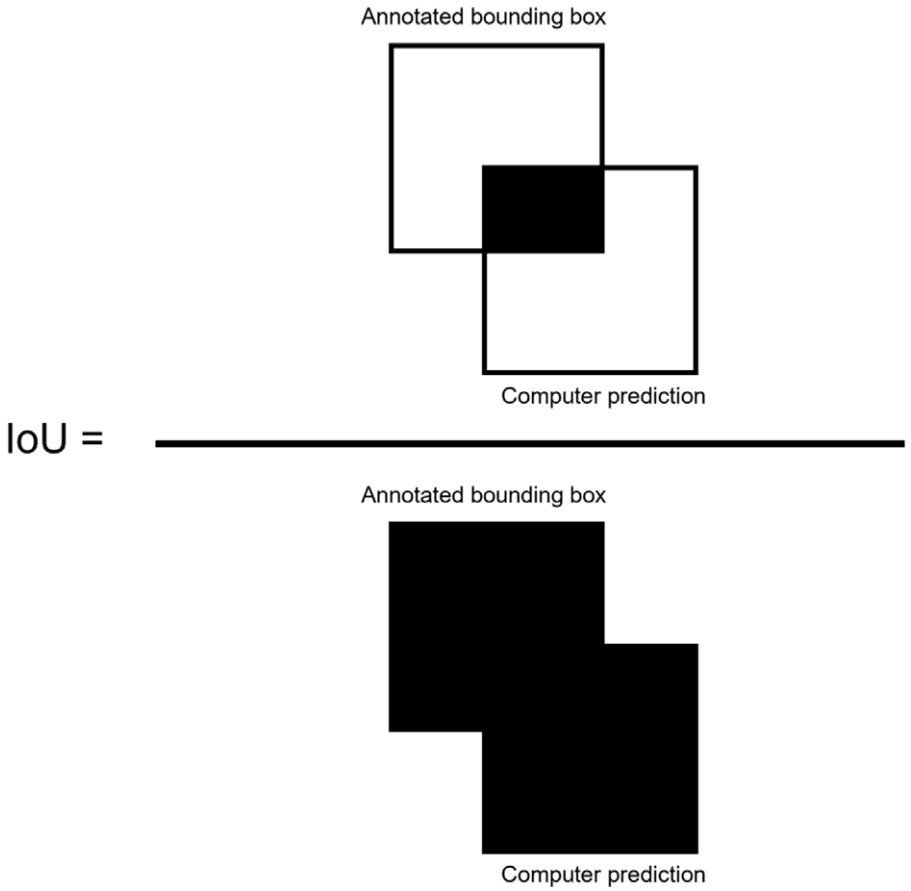


Figure 3. Intersection over union (IoU) equation, defined as the overlap between the ground truth annotation and the computer prediction bounding box, divided by the total area of the two bounding boxes. IoU overlaps greater than 0.5 were considered true-positive predictions, whereas overlaps less than 0.5 were considered false-positive predictions.

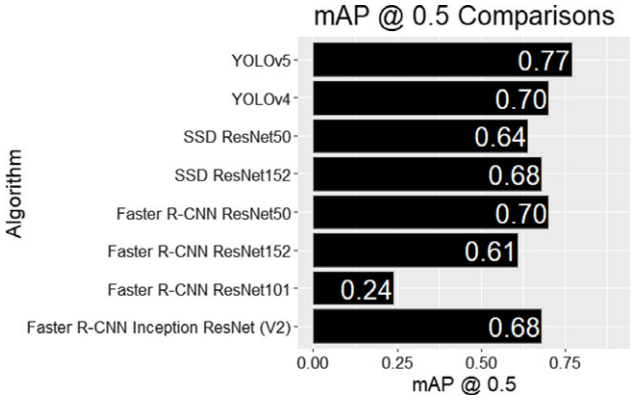


Figure 4. Mean average precision (mAP) results of each model after training. YOLOv5 was considered the best-performing algorithm of each tested model with a mAP of 0.77.

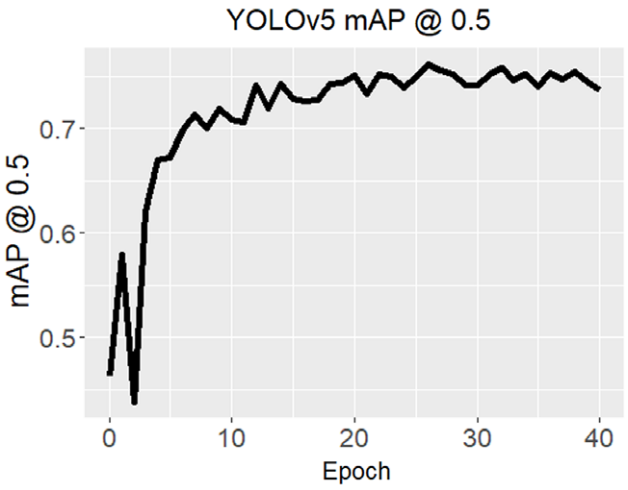


Figure 5. Change in mean average precision (mAP) @ 0.5 over each epoch during training. mAP was reported after the completion of each epoch. Training was terminated after visual inspection of curve and when mAP @ 0.5 curve was seen to "plateau."

the Faster R-CNN Inception ResNet-V2 algorithm is 200,000, but upon monitoring the loss, it was determined that no further increases in algorithm training were being made, and training was terminated early. YOLOv4 training involves iterations, thereby defining a batch size before training and an iteration as complete when the algorithm has processed the number of images specified in the batch size. Finally, YOLOv5 output metrics were reported after each completed epoch, which is defined as one iteration through the entire training data set (Brownlee 2018). Upon viewing an output of the evaluation metrics for each epoch, we terminated algorithm training after 41 epochs, and the best weights were automatically saved for analysis.

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

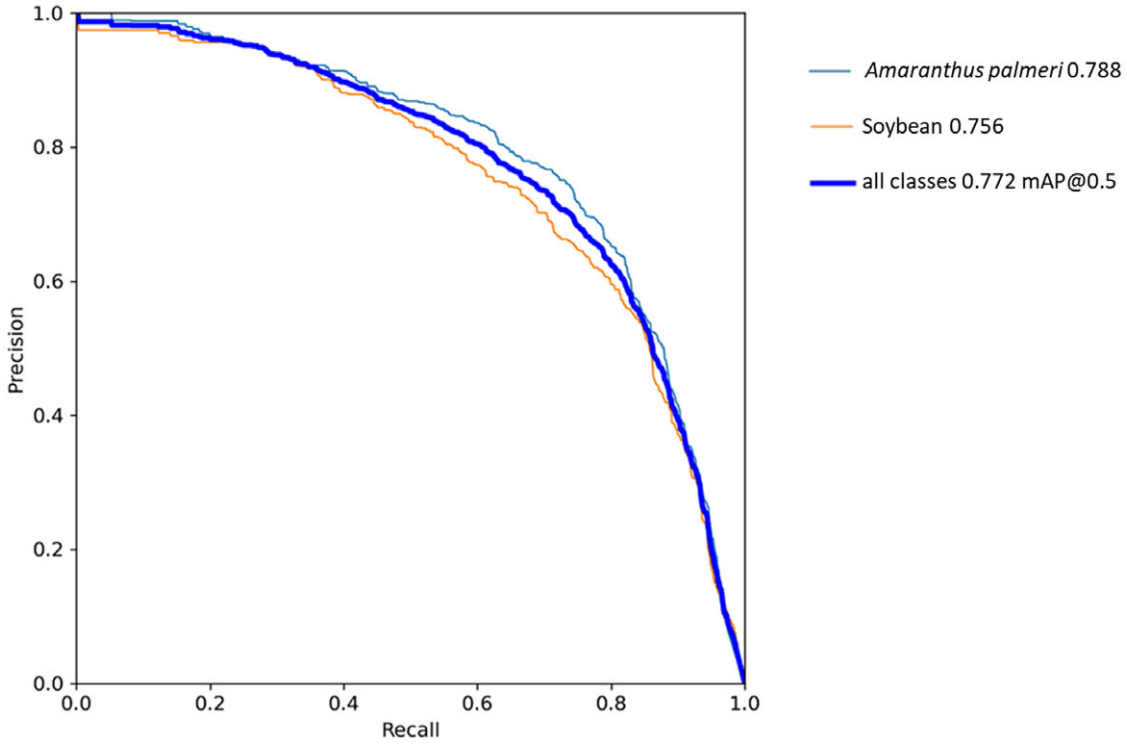


Figure 6. Precision–recall curve for YOLOv5. *Amaranthus palmeri* achieved a slightly higher average precision (AP) (0.788) than soybean. Solid blue line represents mean average precision (mAP) computed on the test data set. The AP for each class and the mAP for the overall algorithm were representative of the area of the graph under each respective curve.

Image augmentation is an important aspect of model training, as it allows for a more comprehensive set of images to be passed through the algorithm and reduce overfitting (Shorten and Khoshgoftaar 2019). Each of the algorithms contained code to automatically augment images during algorithm training, according to each algorithm's default settings. The Faster R-CNN architecture augmentations included random horizontal flips, hue adjustments, contrast adjustments, saturation adjustments, and random image cropping. The SSD architectures used random horizontal flips and random image cropping augmentations. For the YOLO algorithms, the YOLOv4 augmentations included random image saturation, exposure, and hue adjustments (Alexey 2020). Finally, the YOLOv5 model used random mosaicking. This process involves combining an input image with three random images from the training data set. The new mosaic is then passed through the algorithm for training (Jocher 2022b).

Algorithm Evaluation and Statistical Analysis

To measure overall performance of the algorithms, the metrics of precision, recall, and F1 score were computed for the test data set (Shung 2018). Using an intersect over union (IoU) threshold of 0.5 (50%) between the annotated objects and the predicted bounding boxes (Henderson and Ferrari 2017), true-positive (TP) and false-positive (FP) detections are determined. IoU is defined as the overlap between the ground truth bounding boxes drawn during annotation and the predicted bounding box determined by the computer (Jin et al. 2022), divided by the total area of each bounding box (Figure 3). IoU values greater than or equal to 0.5 were considered TP, and values less than 0.5 were considered FP predictions (Henderson and Ferrari 2017).

Precision is the ratio between the number of TP predictions and the total number of positive predictions, with the lowest value being 0 and the highest value being 1 (Hussain et al. 2021). Precision is reduced when an algorithm makes many incorrectly positive, or FP, predictions or a low number of TP predictions but is increased by larger numbers of correct predictions and low FP detections (Gad 2021). Precision was computed with the following equation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad [1]$$

Recall, also referred to as the TP rate (Hussain et al. 2021), is a measure of how well a given algorithm identifies TP predictions (Huilgol 2020). Also ranging from 0 to 1, a higher recall indicates better TP predictions. Recall was computed as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad [2]$$

where FN denotes false-negative detections.

The F1 score is the harmonic mean between precision and recall (Zhong et al. 2019), with the best score being 1 and the worst being 0 (Hussain et al. 2021). It was calculated as:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}} \quad [3]$$

The average precision (AP) for each class is determined by graphing a precision–recall curve for each test image for both classes and computing the area beneath each curve (Henderson and Ferrari 2017). The AP is then used to find the mean AP (mAP) of the

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press



Figure 7. Image annotation of soybean at the R2 growth stage. As soybean populations were much higher than *Amaranthus palmeri* populations, there was a high level of soybean overlap. Therefore, it was necessary to include multiple soybean plants in each image. However, *A. palmeri* plants typically did not have as much overlap, and in most cases, it was much easier to identify and label individual plants.

algorithm, which was calculated from an IoU threshold of 0.5 in this study using the following equation (Jin et al. 2022):

$$\text{mAP} = \frac{\sum_{i=1}^{N} \text{AP}i}{N} \quad [4]$$

where N corresponds to the total number of object classes. Values for mAP range from 0 to 1, with higher values corresponding to larger areas beneath the curve. For this study, the algorithm with the largest mAP was selected to analyze the model evaluation metrics for the photos taken above the individual plots.

For each measurement date, an average for each plot was determined for all field-collected data (*A. palmeri* height, diameter, and density). *Amaranthus palmeri* coverage was computed by multiplying the average canopy area by the average density per plot, assuming a circular shape:

$$\text{T} = \pi r^2 n \quad [5]$$

where T is the average total *A. palmeri* coverage (m^2), r is the average *A. palmeri* radius (m), and n is the average weed density (plants m^{-2}).

Each test image taken over the plots was passed through the best-performing algorithm, and precision, recall, and F1 scores were generated. Each evaluation metric was averaged within its respective plot. Statistical analysis was done using R v. 4.1.2 (R Core Team 2021). Regression models (see "Results and Discussion" section) were used to test whether field-measured data (*A. palmeri* density, height, and coverage) significantly predicted algorithm evaluation metrics. Data were combined over all collection dates and locations for this analysis, as regression assumptions were checked visually for each location and were determined to meet all assumptions (data not shown) (Osborne and Waters 2002). Best regression models were selected based on the Akaike information criterion (AIC) values, such that the model with the lowest AIC value was selected for each evaluation metric. AIC values and weights (indicating the total predicative power among all tested models) were found using the AICcmodavg package in R (Mazerolle 2020). Density and coverage were found to be highly collinear (data not shown), so these variables were never included in the same model together.

Results and Discussion

Algorithm Comparison

The results of algorithm training are presented in Figure 4. After training, it was shown that YOLOv5 achieved the highest overall

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

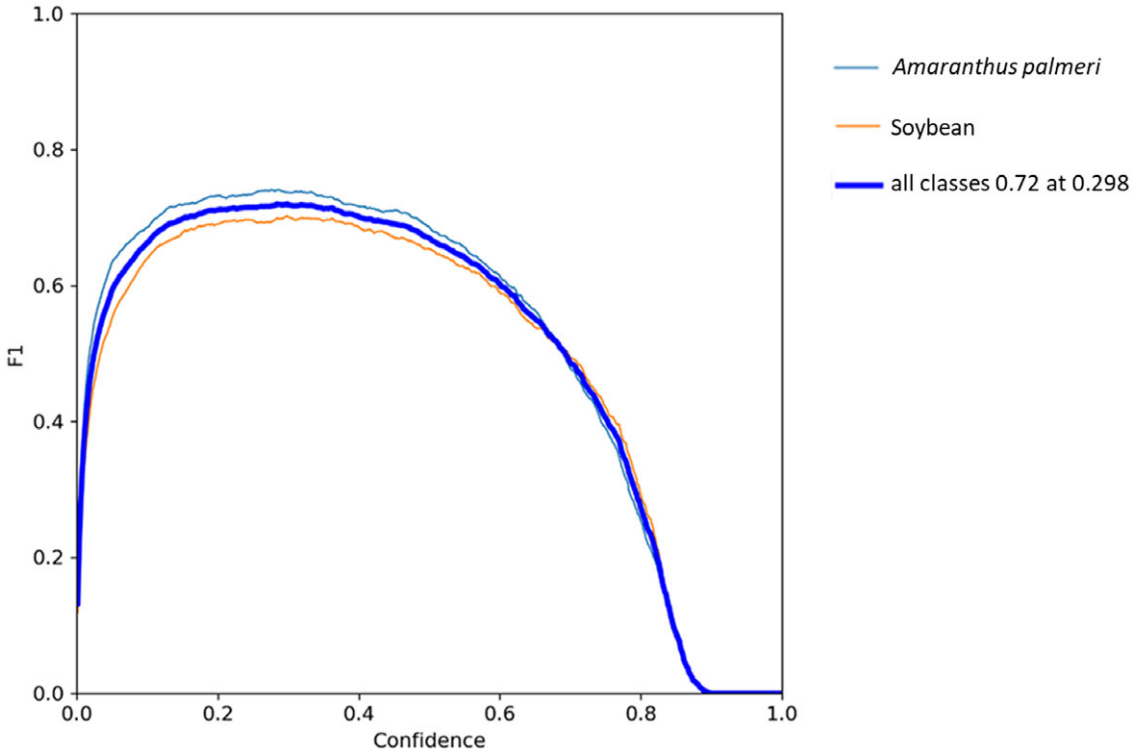


Figure 8. F1 scores for YOLOv5 indicating the harmonic mean between precision and recall scores. Data indicated that detection results for both species would be best at a confidence threshold of 0.298.

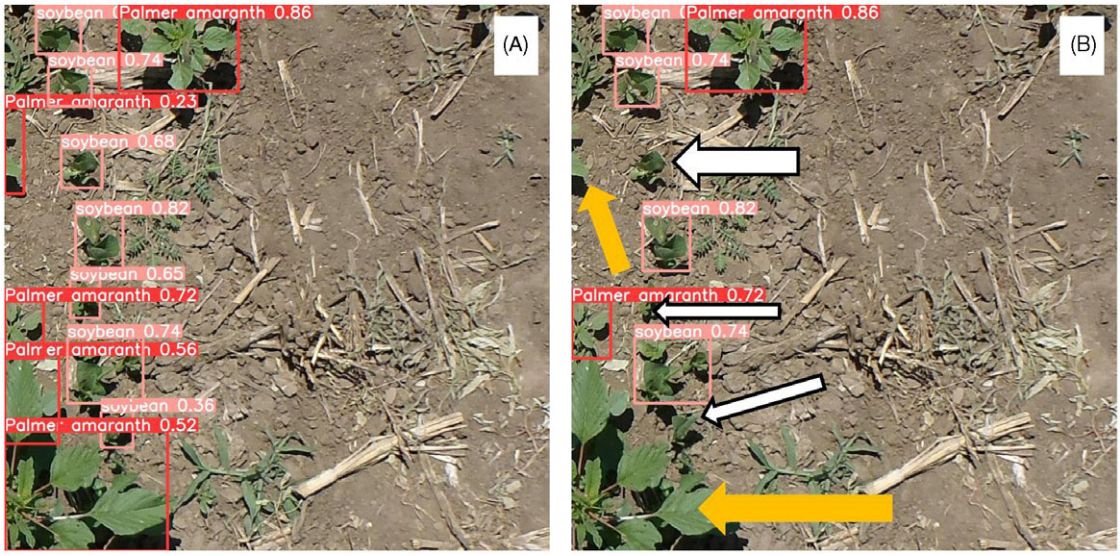


Figure 9. YOLOv5 detection results for *Amaranthus palmeri* and soybean using confidence thresholds of 0.15 (A) and 0.70 (B). The likelihood of false-negative (FN) detections increases as confidence thresholds increase, as can be seen in B. Objects assigned a confidence interval of less than 0.70 are not detected in B. FN A. palmeri and soybean detections in B are indicated by the orange and white arrows, respectively.

mAP value of 0.77. YOLOv4 and Faster R-CNN ResNet50 both achieved acceptable results with mAP values of 0.70, followed by SSD ResNet152 and Faster R-CNN Inception ResNet(v2) with values of 0.68. In most cases, the mAP values for the single-stage detectors (YOLO and SSD) on this data set were equal or superior to those of the two-stage detectors. Faster R-CNN models are generally considered accurate object detectors (Sivakumar et al. 2020), but they can be sensitive to background noise and often have difficulty detecting small objects (Amin and Galasso 2017; Roh and Lee 2017). Our test data set contained images of multiple *A. palmeri* and soybean growth stages, including very small plants of both species. This could explain why the single-stage object detectors often outperformed the Faster R-CNN models in this study. Additionally, we were not surprised to see the higher

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

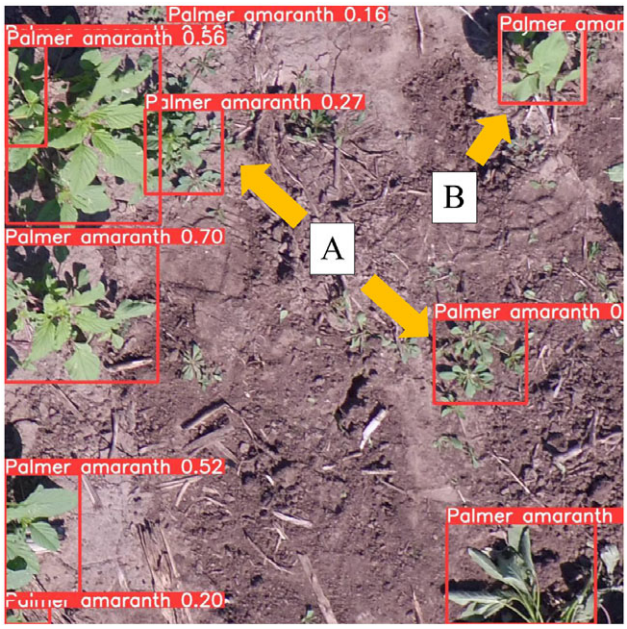


Figure 10. Detection results for YOLOv5 with a confidence interval of 0.15. False-positive detections of Mollugo verticillata and Abutilon theophrasti as *Amaranthus palmeri* are denoted by arrows pointing from "A" and "B," respectively.

performances of YOLOv4 and YOLOv5 algorithms, as previous versions of YOLO have been reported to detect weeds faster than Faster R-CNN algorithms and with greater accuracy than SSD algorithms (Ahmad et al. 2021). Given these results, YOLOv5 was selected for further analysis.

The YOLOv5 algorithm in this study was trained for 41 epochs (Figure 5), taking approximately 15 h to complete. Training was monitored based on the changes in mAP per epoch. Algorithm training was terminated after mAP values were seen to "plateau," thus indicating no further meaningful gains in algorithm performance were expected. During training, YOLOv5 saved the best-performing weights, which were used to compute all further algorithm evaluations. In the precision–recall curve for YOLOv5 (Figure 6), the AP of *A. palmeri* (0.788) is greater than that of soybean (0.756), indicating that detection was slightly better for *A. palmeri* than for soybean. This could be related to differences in the way that the species were annotated. As soybean increased in size, larger bounding boxes were drawn over multiple plants, as there was a high level of overlap between individual plants (Figure 7). Separating out the individual plants would have been both difficult and time-consuming. On the other hand, although there were some overlaps observed with *A. palmeri*, these were much less pronounced and allowed for more individual weed plants to be annotated. Therefore, individual *A. palmeri* plants were presumably easier for the YOLOv5 algorithm to identify.

When using YOLOv5 for detection, users can specify a confidence threshold as an input parameter to the detection script (ultralytics 2022a). This confidence threshold acts to limit the number of FP scores displayed in the final prediction (Wenkel et al. 2021). In our algorithm, lower confidence thresholds increased the likelihood of detecting either an *A. palmeri* or soybean plant, but the FP detection rates increased as a result. Figure 8 illustrates the F1 scores calculated on the test image data set. Interpretation of Figure 8 indicates that at a confidence threshold of 0.298, the highest F1 score (0.72) was achieved for both classes. This indicates that at this threshold, both precision and recall will be optimized for best detection results. As this confidence threshold is a recommended value, users still have the option to set the threshold to a value of their choosing.

There were differences between detections using lower (0.15) and higher (0.70) confidence thresholds (Figure 9). With multiple objects present in this image, YOLOv5 confidence in object detections ranged from low (0.36) to high (0.86); values are displayed immediately following the class prediction on each box. If this algorithm were to be used by a precision ground sprayer or a similar platform, we propose that using a lower confidence threshold would result in more *A. palmeri* plants being identified and treated. Consequently, the likelihood of soybean being incorrectly detected as *A. palmeri* would increase, but in soybean-cropping systems with herbicide-tolerant traits, this would not result in crop damage, assuming all current labels for such applications were followed. An increase in *A. palmeri* FP detections would likely lead to more herbicide being applied to the field. Interestingly, *A. palmeri* FP detections were not just limited to soybean; other broadleaf weeds such as carpetweed (*Mollugo verticillata* L.) and *A. theophrasti* were sometimes detected as *A. palmeri* with lower confidence intervals (Figure 10). This suggests that some weeds with features similar to *A. palmeri* plants would be sprayed if confidence intervals were lowered upon deployment. Regardless, further research is needed to determine which threshold would be optimal to reduce the volume of herbicides applied, while still achieving acceptable weed control.

Modeling YOLOv5 Evaluation Metrics

Only data for the *A. palmeri* class were used to model *A. palmeri* physical characteristics in relation to YOLOv5 evaluation metrics. For all regression models analyzed, the model P-values were significant, indicating that all ground-measured variables affected the evaluation metrics (Table 3). However, it was determined that model 5, which included the main effects and interaction effect of *A. palmeri* density and height, was the model that best fit the data. For precision, recall, and the F1 score, the model carried 72%, 86%, and 91% of the weights from the models that were compared, respectively. Therefore, this model was selected to describe the relationship between model evaluation metrics for density and height.

For all evaluation metrics analyzed with model 5, the interaction coefficient for density and height was significant ($P = 0.049$, 0.016, and 0.010 for precision, recall, and F1 score, respectively) (Table 4). As smaller weeds are generally more susceptible to control (Kieloch and Domaradzki 2011), we chose to model our results with four heights representing *A. palmeri* plants at early growth stages. The results of these predictions by the YOLOv5 algorithm indicated that smaller, younger *A. palmeri* plants growing at lower densities were detected better than taller plants (Figure 11). The results were as anticipated for density, but the prediction suggesting that YOLOv5 detection ability was greater for smaller plants was surprising, as species identification is often easier on larger plants with distinctive features. This may have been because there were a greater number of *A. palmeri* annotations of younger, smaller plants as opposed to larger, more mature plants. Additionally, larger plants have canopies that overlap with each other, making it difficult to distinguish and label large individual plants. Regardless, these predictions were encouraging, as algorithms that can detect smaller weeds are of more practical use, because weeds can be controlled much more easily

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

Table 3. Regression models used to evaluate the effect of *Amaranthus palmeri* morphological parameters on model evaluation metrics and Akaike information criterion (AIC) used for model selection to detect *A. palmeri*. Bold type indicates that model 5 best fit the data.

		Precision			Recall			F1 score		
Model[a]		AIC	AIC Wt[b]	P-value	AIC	AIC Wt[b]	P-value	AIC	AIC Wt[b]	P-value
$y_{em} = \beta_0 + \beta_1 x_1$	1	−259.20	0	<0.0001	−304.63	0	<0.0001	−288.34	0	<0.0001
$y_{em} = \beta_0 + \beta_2 x_2$	2	−302.64	0	<0.0001	−247.97	0	<0.0001	−292.20	0	<0.0001
$y_{em} = \beta_0 + \beta_3 x_3$	3	−263.21	0	<0.0001	−268.37	0	<0.0001	−278.33	0	<0.0001
$y_{em} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$	4	−337	0.28	<0.0001	−331.98	0.12	<0.0001	−349.93	0.09	<0.0001
$y_{em} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_4 x_1 x_2$	**5**	**−338.9**	**0.72**	**<0.0001**	**−335.9**	**0.86**	**<0.0001**	**−354.57**	**0.91**	**<0.0001**
$y_{em} = \beta_0 + \beta_2 x_2 + \beta_3 x_3$	6	−315.60	0	<0.0001	−275.43	0	<0.0001	−311.39	0	<0.0001
$y_{em} = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \beta_5 x_2 x_3$	7	−316.5	0	<0.0001	−274.68	0	<0.0001	−312.58	0	<0.0001
$y_{em} = \beta_0 + \beta_1 x_1 + \beta_6 x_1^2$	8	−260.90	0	<0.0001	−327.38	0.01	<0.0001	−296.14	0	<0.0001
$y_{em} = \beta_0 + \beta_2 x_2 + \beta_7 x_2^2$	9	−301.82	0	<0.0001	−246.06	0	0.0001	−291.60	0	<0.0001
$y_{em} = \beta_0 + \beta_3 x_3 + \beta_8 x_3^2$	10	−286.87	0	<0.0001	−283.37	0	<0.0001	−298.05	0	<0.0001
$y_{em} = \beta_0 + \beta_9 \log(x_1)$	11	−256.02	0	<0.0001	−285.07	0	<0.0001	−278.61	0	<0.0001
$y_{em} = \beta_0 + \beta_{10} \log(x_2)$	12	−256.03	0	<0.0001	−241.07	0	0.0009	−264.51	0	<0.0001
$y_{em} = \beta_0 + \beta_{11} \log(x_3)$	13	−262.75	0	<0.0001	−269.19	0	<0.0001	−281.14	0	<0.0001

[a] y_{em} = evaluation metric (precision, recall, F1 score), x_1 denotes density (plants m^{-2}), x_2 denotes height (cm), and x_3 denotes coverage (m^2). β_0, intercept; β_1, coefficient for *A. palmeri* density (plants m^{-2}); β_2, coefficient for height (cm); β_3, coefficient for coverage (m^2); β_4, interaction coefficient for density and height; β_5, interaction coefficient for height and coverage; β_6, coefficient for the square of density; β_7, coefficient for the square of height; β_8, coefficient for the square of coverage; β_9, coefficient for the log of density; β_{10}, coefficient for the log of height; β_{11}, coefficient for the log of coverage.
[b] AIC weight, indicating the total predictive power among all tested models.

Table 4. Linear regression results (model 5) for *Amaranthus palmeri* density (plants m^{-2}) and height (cm) regressed against model evaluation metrics.

	Parameter estimates[a]					
Evaluation metric	Density	Height	Density × height	R^2	RMSE[b]	P-value
Precision	-8.6×10^{-4}***	-3.4×10^{-3}***	1.1×10^{-5}*	0.42	0.10	<0.0001
Recall	-1.3×10^{-3}***	-2.2×10^{-3}***	1.4×10^{-5}*	0.43	0.10	<0.0001
F1 score	-1.1×10^{-3}***	-3.0×10^{-3}***	1.4×10^{-5}*	0.44	0.10	<0.0001

[a] Significant at: *P < 0.05; ***P < 0.001.
[b] RMSE, root mean-square error.

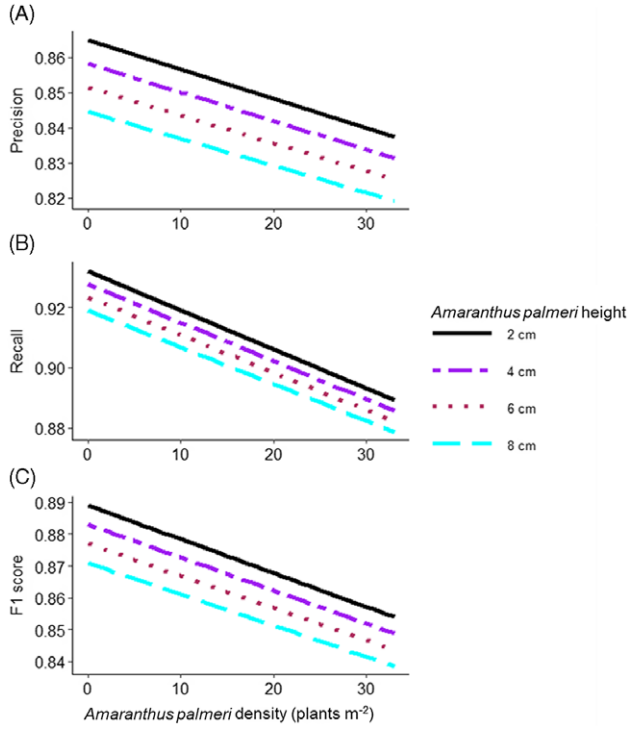


Figure 11. YOLOv5 precision (A), recall (B), and F1 score (C) changes as a function of *Amaranthus palmeri* density (plants m^{-2}).

when they are younger and smaller in size (Naghashzadeh and Beyranvand 2015).

In relation to weed density affecting algorithm performance, our study conflicts with Yu et al. (2020), in that they found that images with higher weed densities generally led to better algorithm detection results than those with lower weed densities. However, the study by Yu et al. (2020) utilizes image classification rather than object detection. Rather than localizing the weeds within the image, the entire input image was classified with the weed species that were visible. Image classification for weed detection comes with some disadvantages, however, as the location of individual weeds was not provided, and multiple weed species within an image were not able to be detected (Ahmad et al. 2021). With object detection algorithms, object localization within the image allows for weeds to be located and controlled where they occur. However, we hypothesize that this application is best suited for postemergence applications (in the case of using precision herbicide application technology) where weed density is relatively low. Fields with very high weed densities would likely not benefit from a site-specific herbicide application, as the volume of herbicide needed for control would likely not be statistically different from a whole-field broadcast application. Further research is needed to determine the optimum weed density beyond which precision weed control has no economic or environmental benefits.

The overall precision, recall, and F1 scores computed for the 450 images in the test data set were 0.71, 0.70, and 0.71, respectively (data not shown). A precision of 0.71 indicates that the YOLOv5

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

algorithm was 71% accurate in successfully predicting *A. palmeri* and soybeans. Likewise, a recall of 0.70 indicates that the algorithm correctly predicted 70% of the plants belonging to either class (Jin et al. 2022). These results were lower than previously reported by other YOLO weed detectors (Jin et al. 2022; Zhuang et al. 2022). However, our "test" data set consisted of images randomly selected from the large-input database and had a variety of *A. palmeri* growth stages and population densities. When evaluating YOLOv5 evaluation metrics on images taken within the plots with lower *A. palmeri* densities and shorter plant heights, the algorithm precision, recall, and F1 scores greatly improved. Based on the regression model fit to the data, *A. palmeri* plants 2-cm tall and growing at a density of 1 plant m^{-2} would be detected with precision, recall, and F1 scores of 0.87, 0.93, and 0.89, respectively.

YOLO algorithms have been used previously for weed detection. Ahmad et al. (2021) achieved an overall mAP score of 0.543 when using YOLOv3 (Redmon and Farhadi 2018) to detect redroot pigweed (*Amaranthus retroflexus* L.), giant ragweed (*Ambrosia trifida* L.), common cocklebur (*Xanthium strumarium* L.), and green foxtail [*Setaria viridis* (L.) P. Beauv.]. Hussain et al. (2020) developed an in situ sprayer using both YOLOv3 and YOLOv3-tiny (Adarsh et al. 2020) as backbone algorithms to detect *C. album*, achieving mAP scores of 0.932 and 0.782, respectively. Sharpe et al. (2020b) achieved good detection results when training the YOLOv3 algorithm to identify general classes of grasses, broadleaves, and sedge species; further, they found that including multiple classes (as opposed to a single class) in their algorithm increased precision, recall, and F1 metrics. Hu et al. (2021) used YOLOv3 and YOLOv4 to detect 12 different weed species common to rice (*Oryza sativa* L.) and found that YOLOv4 achieved a mAP score that was 0.116 higher than YOLOv3. Our best mAP score was slightly lower than some of these YOLO weed detectors; however, it must be mentioned that this data set was collected with multiple cameras covering a variety of *A. palmeri* densities and growth stages. Data sets such as those collected by Jin et al. (2022) and Zhuang et al. (2022) consisted of a handheld camera taking multiple images at a consistent height. In this study, we collected imagery ranging from 1.5 m to 8 m above ground level. While many data sets collected consist of "ideal" specimens including plants grown in greenhouses or photos of individual plants, our data set was based on what a field sprayer or application UAV would observe in the field. As a result, during the labeling process, several overlapping bounding boxes had to be drawn, and it was impossible for each image to contain labels for individual plants.

In conclusion, this research demonstrated that YOLOv5, a free and open-source object detection algorithm, can detect *A. palmeri* in soybean-cropping systems. As site-specific herbicide applications become more widespread due to the potential for herbicide savings and environmental benefits, open-source algorithms such as YOLOv5 could enable increased development and adoption of precision weed detectors. Furthermore, this research suggests that our algorithm may be better at detecting smaller as opposed to larger *A. palmeri* plants. Upon further refinement and training of the algorithm, it may be of great use to growers, as smaller weeds are much more susceptible to control than larger ones.

Future research and improvements to our model will include adding more images to the data set. We included different imagery heights in this study to create a data set that could be utilized by multiple precision agriculture platforms such as precision ground sprayers and pesticide application UAVs. In the future, construction of specialized data sets that consist of imagery for each type of platform would be collected. For instance, imagery collected to train an algorithm for a precision ground sprayer should be at a height consistent with the sensors on the sprayer itself, that is, 60 cm above the target canopy. In this experiment, we trained the object detectors to identify two species (*A. palmeri* and soybean), and a future goal is to expand the number of weed species that can be detected by the YOLOv5 model. An increase in both the number of images and number of annotated weed species in these specialized data sets would increase the mAP of the YOLOv5 algorithm and reduce errors in object detection (Linn et al. 2019). Equal distribution of annotations among species is important when collecting these images. With further improvements to the algorithm, field tests will need to be carried out to both optimize weed detection and to treat weeds in real time using a precision ground sprayer with high-resolution cameras oriented close to plant canopies.

Acknowledgments. This research received no specific grant from any funding agency or the commercial or not-for-profit sectors. No conflicts of interest have been declared. This is contribution no. 23-042-J from the Kansas Agricultural Experiment Station.

References

Adarsh P, Rathi P, Kumar M (2020) YOLO v3-Tiny: object detection and recognition using one stage improved model. Pages 687–694 *in* Proceedings from the 6th International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore, India: Institute of Electrical and Electronics Engineers

Ahmad A, Saraswat D, Aggarwal V, Etienne A, Hancock B (2021) Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems. Comput Electron Agric 184:106081

Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. Pages 1–6 *in* Proceedings of the 2017 International Conference on Engineering & Technology. Antalya, Turkey: Institute of Electrical and Electronics Engineers

Alexey AB (2020) CFG Parameters in the [Net] Section. https://github.com/AlexeyAB/darknet/wiki/CFG-Parameters-in-the-%5Bnet%5D-section. Accessed: July 19, 2022

Alexey AB (2022) darknet. https://github.com/AlexeyAB/darknet. Accessed: February 3, 2022

Al Zorgani MM, Mehmood I, Ugail H (2022) Deep YOLO-based detection of breast cancer mitotic-cells in histopathological images. Pages 335–352 *in* Su R, Zhang YD, Liu H, eds. Proceedings of 2021 International Conference on Medical Imaging and Computer-Aided Diagnosis (MICAD 2021). MICAD 2021 (Lecture Notes in Electrical Engineering, Vol. 784). Singapore: Springer

Amin S, Galasso F (2017) Geometric proposals for faster R-CNN. Pages 1–6 *in* Proceedings of the 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). Lecce, Italy: Institute of Electrical and Electronics Engineers

Arsenijevic N, de Avellar M, Butts L, Arneson NJ, Werle R (2021) Influence of sulfentrazone and metribuzin applied preemergence on soybean development and yield. Weed Technol 35:210–215

Balasubramanian R (2021) Region–based convolutional neural network (RCNN). https://medium.com/analytics-vidhya/region-based-convolutional-neural-network-rcnn-b68ada0db871. Accessed: February 2, 2022

Barroso J, Fernandez-Quintanilla C, Maxwell BD, Rew LJ (2004) Simulating the effects of weed spatial pattern and resolution of mapping and spraying on economics of site-specific management. Weed Res 44:460–468

Biffi LJ, Mitishita E, Liesenberg V, dos Santos AA, Gonçalves DN, Estrabis NV, Silva JdA, Osco LP, Ramos APM, Centeno JAS, Schimalski MB, Rufato L,

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

Neto SLR, Junior JM, Gonçalves WN (2021) ATSS deep learning-based approach to detect apple fruits. Remote Sens 13:54
Bochkovskiy A, Wang CY, Liao HYM (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv database 2004.10934. https://arxiv.org/abs/2004.10934. Accessed: February 2, 2022
Bongiovanni R, Lowenberg-Deboer J (2004) Precision agriculture and sustainability. Precis Agric 5:359–387
Brownlee J (2018) Difference between a Batch and an Epoch in a Neural Network. https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch. Accessed: February 3, 2022
Cardina J, Johnson GA, Sparrow DH (1997) The nature and consequence of weed spatial distribution. Weed Sci 45:364–373
Carlos LMJ, Ulson JAC (2021) Real time weed detection using computer vision and deep learning. Pages 1131–1137 *in* Proceedings from the 14th IEEE International Conference on Industry Applications (INDUSCON). São Paulo, Brazil: Institute of Electrical and Electronics Engineers
Chen JW, Lin WJ, Cheng HJ, Hung CL, Lin CY, Chen SP (2021) A smartphone-based application for scale pest detection using multiple-object detection methods. Electronics 10:372
Clark A (2022) Pillow 9.2.0. Code. https://pypi.org/project/Pillow. Accessed: July 20, 2022
Dayan FE (2021) Market trends of the herbicide crop protection industry and impact of COVID-19. Outlook Pest Manag 32:2–4
dos Santos Ferreira A, Freitas DM, da Silva GC, Pistori H, Folhes MT (2019) Unsupervised deep learning and semi-automatic data labeling in weed discrimination. Comput Electron Agric 165:104963
Du L, Zhang R, Wang X (2020) Overview of two-stage object detection algorithms. J Phys Conf Ser 1544:012033
Evans JA, Tranel PJ, Hager AG, Schutte B, Wu C, Chatham LA, Davis AS (2015) Managing the evolution of herbicide resistance. Pest Manag Sci 72:74–80
Fehr WR, Caviness CE, Burmood DT, Pennington JS (1971) Stage of development descriptions for soybeans, *Glycine max* (L.) Merrill. Crop Sci 11: 929–931
Fernández-Quintanilla C, Peña JM, Andújar D, Dorado J, Ribeiro A, López-Granados F (2018) Is the current state of the art of weed monitoring suitable for site-specific weed management in arable crops? Weed Res 58:259–272
Forson E (2017) Understanding SSD MultiBox—Real-Time Object Detection in Deep Learning. https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab. Accessed: February 2, 2022
Gad AF (2021) Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall. https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy. Accessed: February 3, 2022
Gao J, French AP, Pound MP, He Y, Pridmore TP, Pieters JG (2020) Deep convolutional neural networks for image-based *Convolvulus sepium* detection in sugar beet fields. Plant Methods 16:29
Ghazi MM, Yanikoglu B, Aptoula E (2017) Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235:228–235
Girshick R (2015) Fast R-CNN. Pages 1440–1448 *in* Proceedings of the IEEE International Conference on Computer Vision (ICCV). Santiago, Chile: Institute of Electrical and Electronics Engineers
Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. Pages 580–587 *in* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Columbus, OH: Institute of Electrical and Electronics Engineers
Gurtner A, Walker R, Boles W (2007) Vibration compensation for fisheye lenses in UAV Applications. Pages 218–225 *in* Proceedings from the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007). Glenelg, SA, Australia: Institute of Electrical and Electronics Engineers
He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. Pages 770–778 *in* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV: Institute of Electrical and Electronics Engineers
Henderson P, Ferrari V (2017) End-to-end training of object class detectors for mean average precision. Pages 198–213 *in* Lai SH, Lepetit V, Nishino K, Sato Y, eds. Computer Vision—ACCV 2016. ACCV 2016 (Lecture Notes in Computer Science, Vol. 10115). Cham, Switzerland: Springer
Hu D, Tian Z, Li L, Ma C (2021) Rice weed detection method on YOLOv4 convolutional neural network. Pages 41–45 *in* Proceedings from the 2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA). Xi'an, China: Institute of Electrical and Electronics Engineers
Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K (2017) Speed/accuracy trade-offs for modern convolutional object detectors. Pages 7310–7311 *in* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: Institute of Electrical and Electronics Engineers
Hui J (2018) Real-Time Object Detection with YOLO, YOLOv2 and Now YOLOv3. https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088. Accessed: July 20, 2022
Huilgol P (2020) Precision vs. Recall—An Intuitive Guide for Every Machine Learning Person. https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning. Accessed: February 3, 2022
Hussain N, Farooque AA, Schumann AW, Abbas F, Acharya B, McKenzie-Gopsill A, Barrett R, Afzaal H, Zaman QU, Cheema MJM (2021) Application of deep learning to detect lamb's quarters (*Chenopodium album* L.) in potato fields of Atlantic Canada. Comput Electron Agric 182:106040
Hussain N, Farooque AA, Schumann AW, McKenzie-Gopsill A, Esau T, Abbas F, Acharya B, Zaman Q (2020) Design and development of a smart variable rate sprayer using deep learning. Remote Sens 12:4091
Jha P, Norsworthy JK (2009) Soybean canopy and tillage effects on emergence of Palmeramaranth (*Amaranthus palmeri*) from a natural seedbank. Weed Sci 57:644–651
Jin Z, Sun Y, Che J, Bagavathiannan M, Yu J, Chen Y (2022) A novel deep learning-based method for detection of weeds in vegetables. Pest Manag Sci 78:1861–1869
Jocher G (2022a) yolov5. https://github.com/ultralytics/yolov5. Accessed: February 3, 2022
Jocher G (2022b) yolov5: data augmentation #8021. https://github.com/ultralytics/yolov5/issues/8021#issuecomment-1140326033. Accessed: July 19, 2022
Jocher G, Stoken A, Borovec J, NanoCode012, ChristopherSTAN, Changyu L, Laughing, tkianai, Hogan A, lorenzomammana, yxNONG, AlexWang1900, Diaconu L, Marc, wanghaoyang0206, *et al.* (2020) ultralytics/yolov5: v3.1—Bug Fixes and Performance Improvements. https://doi.org/10.5281/zenodo.4154370. Accessed: February 2, 2022
Kieloch R, Domaradzki K (2011) The role of the growth stage of weeds in their response to reduced herbicide doses. ACTA Agrobotanica 64:259–266
Klingaman TE, Oliver LR (1994) Palmer amaranth (*Amaranthus palmeri*) interference in soybeans (*Glycine max*). Weed Sci 42:523–527
Kulkarni R, Dhavalikar S, Bangar S (2018) Traffic light detection and recognition for self driving cars using deep learning. Pages 1–4 *in* Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). Pune, India: Institute of Electrical and Electronics Engineers
Kumar V, Liu R, Peterson DE, Stahlman PW (2021) Effective two-pass herbicide programs to control glyphosate-resistant Palmer amaranth (*Amaranthus palmeri*) in glyphosate/dicamba-resistant soybean. Weed Technol 35:128–135
Li D, Wang R, Xie C, Liu L, Zhang J, Li R, Wang F, Zhou M, Liu W (2020) A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network. Sensors 20:578
Li J, Liang X, Wei Y, Xu T, Feng J, Yan S (2017) Perceptual Generative Adversarial Networks for Small Object Detection. Pages 1222–1230 *in* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI: Institute of Electrical and Electronics Engineers

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

Lin T, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollár P (2014) Microsoft COCO: common objects in context. Pages 740–755 *in* Proceedings of the 13th European Conference on Computer Vision, Part V. Zurich, Switzerland: European Conference on Computer Vision

Lin TY, Goyal P, Girshick R, He K, Dollár P (2018) Focal Loss for Object Detection. arXiv database 1708.02002v2. https://arxiv.org/pdf/1708.02002v2.pdf. Accessed: July 20, 2022

Linn AI, Mink R, Peteinatos GG, Gerhards R (2019) In-field classification of herbicide-resistant *Papaver rhoeas* and *Stellaria media* using an imaging sensor of the maximum quantum efficiency of photosystem II. Weed Res 59:357–366

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. Pages 21–37 *in* Proceedings of the 14th European Conference on Computer Vision, Part 1. Amsterdam: European Conference on Computer Vision.

Mattman C, Zhang Z (2019) Deep facial recognition using Tensorflow. Pages 45–51 *in* Proceedings from the 2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS). Denver, CO: Institute of Electrical and Electronics Engineers

Maxwell BD, Luschei EC (2005) Justification for site-specific weed management based on ecology and economics. Weed Sci 53:221–227

Mazerolle MJ (2020) AICcmodavg: Model Selection and Multimodel Inference Based on (Q)AIC(c). https://cran.r-project.org/package=AICcmodavg. Accessed: February 17, 2022

Mordor Intelligence (2022) Crop Protection Chemicals Market—Growth, Trends, COVID-19 Impact, and Forecast (2022–2027). https://www.mordorintelligence.com/industry-reports/global-crop-protection-chemicals-pesticides-market-industry. Accessed: January 28, 2022

Naghashzadeh M, Beyranvand AF (2015) Broad-leaved weeds in chickpea (*Cicer arietinum* L.) as affected by plant density and Lentagran herbicide application. Electron J Biol 11:90–92

Oinar C (2021) Object Detection Explained: R-CNN. https://towardsdatascience.com/object-detection-explained-r-cnn-a6c813937a76. Accessed: February 2, 2022

Osborne JW, Waters E (2002) Four assumptions of multiple regression that researchers should always test. Pract Assess Res Eval 8:1–5

Osorio K, Puerto A, Pedraza C, Jamaica D, Rodriguez L (2020) A deep learning approach for weed detection in lettuce crops using multispectral images. AgriEngineering 2:471–488

Oza P, Patel VM (2019) One-class convolutional neural network. IEEE Signal Proc Let 26:2

Parico AIB, Ahamed T (2020) An aerial weed detection system for green onion crops using the you only look once (YOLOv3) deep learning algorithm. Eng Agric Environ Food 13:42–48

Pokhrel S (2020) Collecting Data for Custom Object Detection. https://towardsdatascience.com/collecting-data-for-custom-object-detection-e7d888c1469b. Accessed: January 31, 2022

Python Software Foundation (2022) Python Language Reference. Version 3.9.7. http://www.python.org Accessed: March 3, 2022

R Core Team (2021) R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org. Accessed: February 3, 2022

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. Pages 779–788 *in* Proceedings from the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV: Institute of Electrical and Electronics Engineers

Redmon J, Farhadi A (2018) YOLOv3: An Incremental Improvement. arXiv database 1804.02767. https://arxiv.org/abs/1804.02767. Accessed: February 16, 2022

Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE T Pattern Anal 39:1137–1149

Roh MC, Lee JY (2017) Refining Faster-RCNN for accurate object detection. Pages 514–517 *in* Proceedings of the Fifteenth IAPR International Conference on Machine Vision Applications (MVA). Nagoya, Japan: Institute of Electrical and Electronics Engineers

Ruder S (2021) Recent Advances in Language Model Fine-Tuning. https://ruder.io/recent-advances-lm-fine-tuning. Accessed: January 31, 2022

Sapkota B, Singh V, Cope D, Valasek J, Bagavathiannan M (2020) Mapping and estimating weeds in cotton using unmanned aerial systems-borne imagery. AgriEngineering 2:350–366

Sharpe SM, Schumann AW, Boyd NS (2018) Detection of Carolina geranium (*Geranium carolinianum*) growing in competition with strawberry using convolutional neural networks. Weed Sci 67:239–245

Sharpe SM, Schumann AW, Boyd NS (2020a) Goosegrass detection in strawberry and tomato using a convolutional neural network. Sci Rep 10:9548

Sharpe SM, Schumann AW, Yu J, Boyd NS (2020b) Vegetation detection and discrimination within vegetable plasticulture row-middles using a convolutional neural network. Precis Agric 21:264–277

Shimao (2019) How Do Backbone and Head Architecture Work in Mask R-CNN? https://stats.stackexchange.com/questions/397767/how-do-backbone-and-head-architecture-work-in-mask-r-cnn/401107. Accessed: February 3, 2022

Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6:60

Shung KP (2018) Accuracy, Precision, Recall or F1? https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. Accessed: April 28, 2022

Shyam C, Chahal PS, Jhala AJ, Jugulam M (2021) Management of glyphosate-resistant Palmer amaranth (*Amaranthus palmeri*) in 2,4-D-, glufosinate-, and glyphosate-resistant soybean. Weed Technol 35:136–143

Sivakumar ANV, Li J, Scott S, Psota E, Jhala AJ, Luck JD, Shi Y (2020) Comparison of object detection and patch-based classification deep learning models on mid- to late-season weed detection in UAV imagery. Remote Sens 12:2136

Somerville GJ, Sønderskov M, Mathiassen SK, Metcalfe H (2020) Spatial modelling of within-field weed populations: a review. Agronomy 10:1044

Stanford (2022) Tutorial 3: Image Segmentation. ~https://ai.stanford.edu/~syyeung/cvweb/tutorial3.html. Accessed: July 19, 2022

Sun W, Dai L, Zhang X, Chang P, He X (2022) RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. Appl Intell 52:8448–8463

Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, Inception-ResNet and the impact of residual connections on learning. Pages 4278–4284 *in* Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. San Francisco, CA: Association for the Advancement of Artificial Intelligence Press

TensorFlow (2017) Resource Exhausted: OOM when Allocating Tensor with Shape [2304,384] Traceback (most recent call last): #1993. https://github.com/tensorflow/models/issues/1993#issue-244306864. Accessed: July 20, 2022

TensorFlow (2021) TensorFlow 2 Detection Model Zoo. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md. Accessed: February 2, 2022

Thanh Le VN, Truong G, Alameh K (2021) Detecting weeds from crops under complex field environments based on Faster RCNN. Pages 350–355 *in* Proceedings from the 2020 IEEE Eighth International Conference on Communications and Electronics (ICCE). Phu Quoc Island, Vietnam: Institute of Electrical and Electronics Engineers

ThinkAutomation (2022) ELI5: What Is Image Classification in Deep Learning? https://www.thinkautomation.com/eli5/eli5-what-is-image-classification-in-deep-learning. Accessed: April 28, 2022

Tzutalin (2015) LabelImg. Code https://github.com/tzutalin/labelImg. Accessed February 2, 2022

ultralytics (2022a) yolov5/detect.py. https://github.com/ultralytics/yolov5/blob/master/detect.py. Accessed: February 15, 2022

ultralytics (2022b) v6.1—TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. https://github.com/ultralytics/yolov5/releases. Accessed: July 19, 2022

vdalv (2017) Convert PascalVOC Annotations to YOLO. Code. https://gist.github.com/vdalv/321876a7076caaa771d47216f382cba5. Accessed: July 18, 2022

[WSSA] Weed Science Society of America (2016) WSSA Survey Ranks Palmer Amaranth as the Most Troublesome Weed in the U.S., Galium as the Most Troublesome in Canada. https://wssa.net/2016/04/wssa-survey-ranks-palmer-amaranth-as-the-most-troublesome-weed-in-the-u-s-galium-as-the-most-troublesome-in-canada. Accessed: January 28, 2022

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press

Wenkel S, Alhazmi K, Liiv T, Alrshoud S, Simon M (2021) Confidence score: the forgotten dimension of object detection performance evaluation. Sensors 21:4350

Xu W, Matzner S (2018) Underwater fish detection using deep learning for water power applications. Pages 313–318 *in* Proceedings of the 2018 International Conference on Computer Science and Computational Intelligence (CSCI). Las Vegas, NV: Institute of Electrical and Electronics Engineers

Yang Q, Shi L, Han J, Zha Y, Zhu P (2019) Deep convolutional neural networks for rice grain yield estimation at the ripening stage using UAV-based remotely sensed images. Field Crop Res 235:142–153

Ying B, Xu Y, Zhang S, Shi Y, Liu L (2021) Weed detection in images of carrot fields based on improved YOLO v4. Trait Signal 38:341–348

Yu J, Schumann AW, Sharpe SM, Li X, Boyd NS (2020) Detection of grassy weeds in bermudagrass with deep convolutional neural networks. Weed Sci 68:545–552

Zhong L, Hu L, Zhou H (2019) Deep learning based multi-temporal crop classification. Remote Sens Environ 221:430–443

Zhuang J, Li X, Bagavathiannan M, Jin X, Yang J, Meng W, Li T, Li L, Wang Y, Chen Y, Yu J (2022) Evaluation of different deep convolutional neural networks for detection of broadleaf weed seedlings in wheat. Pest Manag Sci 2022:521–529

https://doi.org/10.1017/wsc.2022.53 Published online by Cambridge University Press