Crop and Weed Detection in Sunflower and Sugarbeet fields Using Single Shot Detectors

Diaa Addeen Abuhani
Department of Computer Science and Engineering
American University of Sharjah
Sharjah, UAE
b00086137@aus.edu

Maya Haj Hussain
Department of Computer Science and Engineering
American University of Sharjah
Sharjah, UAE
g00085636@aus.edu

Jowaria Khan
Department of Computer Science and Engineering
American University of Sharjah
Sharjah, UAE
g00084343@aus.edu

Mohamed ElMohandes
Department of Computer Science and Engineering
American University of Sharjah
Sharjah, UAE
b00083108@aus.edu

Imran Zualkernan
Department of Computer Science and Engineering
American University of Sharjah
Sharjah, UAE
izualkernan@aus.edu

2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS) | 979-8-3503-4647-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/COINS57856.2023.10189257

***Abstract*— Weed detection plays a critical role in improving agricultural production. Distinguishing crops from weeds is vital for achieving precise spraying for weeds without polluting the ecological environment as a whole. Many computer vision methods have been proposed to achieve reliable weed detection with relatively high speed. However, the lack of publicly available datasets hinders the efforts to improve the work in this field. In this paper, a manually labeled dataset for sugar beets and sunflowers was developed. Different state-of-the-art single-short object detection architectures and methods were trained on the two datasets. A comparison of the various methods to detect weeds is presented. The primary result is that the You Look Only Once (YOLO) family of architectures were better at weed detection for these data sets than alternative architectures like RatinaNet, EfficientDet and Detection Transformer (DETR). One key challenge was the detection of smaller weeds.**

Keywords—Weed Detection, Precision Farming, UGVs, Single Shot Detectors, YOLO

I. Introduction

Weed detection and management plays a crucial role towards ensuring the growth of healthy crops and plants. According to Food and Agriculture Organization (FAO), weeds account for 15 to 30 percent reduction in grain yield loss [1]. Weed damage varies based on its type, crop growth stage, and the duration of competition between weeds and crops. Traditionally, the task of removing weeds is manually carried out by farmers through field inspections followed by weed uprootals or field-wide agrochemical application. Given the time-consuming, labor-intensive, and wasteful nature of this work, in conjunction with the increased demand for crop production, many have turned to more automated solutions based in computer vision and deep learning. These solutions have focused on accurately classifying and more precisely detecting weeds amongst various crops in agricultural fields. With the automation of the weed detection task, manual field inspection can potentially be partially eliminated. In addition, the task of weed removal can become less daunting and more accurate. In addition, selective agrochemicals application can lead to reduced costs.

The contributions of this paper are as follows:

- Provide manually labeled datasets of bounding box-annotated images of crops and weeds obtained using an Unmanned ground vehicle (UGV) from two different crop fields, namely, sunflower and sugar beet.
- Compare state-of-the-art detection models on the manually labelled datasets

The rest of the paper is organized as follows. Selected previous work in using neural networks to detect weeds is presented first. The materials and methods section then provide an overview of the datasets used, discusses the evaluation metrics, and introduces the various single-stage detection approaches that can be utilized for weed detection. The methodology is discussed next. Finally, the results are discussed followed by a discussion and analysis of the obtained outputs.

II. Previous Work

Previous work in weed detection includes semantic segmentation of weeds and crops, and targeted object detection of weeds within an agricultural field. For example, Arun et al. [2] proposed a U-Net-based semantic segmentation model to classify weed and crop pixels in images from the Crop/Weed Field Image dataset (CWFID). Their work presented an efficient version of the U-Net model that employed a small convolutional neural network (CNN) and required a smaller number of parameters when compared to the original U-Net architecture. Sodjinou et al. [3] performed instance segmentation on images of crops and weeds using a method that combined a U-Net with a K-means subtractive algorithm. Fawakherji et al. [4] presented a robust BonNet architecture that could accurately segment crops and weeds in RGB and Near Infrared Spectroscopy (NIR) images belonging to multiple datasets after training on datasets of real (RGB+ NIR) and synthetic images.

Object detection methods including two-stage detector models and newer one-shot detector models have also been used to detect weeds [5]. These models can locate and classify objects within an image, where bounding boxes are used to identify the located objects. Two-stage detectors generally performed better than one-stage detectors. However, such models are more computationally expensive

979-8-3503-4647-3/23/$31.00 ©2023 IEEE

and have higher inference times. This is because an input image goes through two stages, the first of which is a region proposal network (RPN) that outputs candidate regions where objects of interest are present, and the second stage is a classifier that classifies the regions and outputs associated confidence scores. One-shot detectors, on the other hand, can detect objects and predict their classes in one single feed-forward pass without going through a region proposal network. The You Only Look Once (YOLO) class of architectures is well known for doing such tasks [6]. However, weed detection remains an obstacle due to the limitations of object size and interleaving branches. Recently, Dandekar et al. [7] used the YOLO v3 object detection model to detect crops and weeds in images obtained from the Sesame crop and Weed Dataset. Similarly, Czymmek et al. [8] used the YOLO v1 model to detect various weed species in images collected from carrot fields. Finally, Doddamani et al. [9] used the YOLO v5 model for the detection of early-stage weeds in images acquired from an Egyptian star cluster flower field. Table I summarizes weed detection results using YOLO architecture. As the table shows, YOLO's performance varies across data sets and can reach an accuracy as high as 96%.

TABLE I. **PREVIOUS WORK ON WEED DETECTION USING YOLO**

Work	Dataset	Arch.	Best results
Dandekar et al [7]	Sesame crop and Weed Dataset	YOLOv3	96.00% accuracy
Czymmek et al [8]	Carrot fields weeds	YOLOv1	62.2 % avg IoU
Doddamani et al [9]	an Egyptian star cluster flower field.	YOLOv5	~15% mAP

III. MATERIALS AND METHODS

A. Sugar Beets Dataset

The original sugar beets dataset was originally collected through a UGV robot operating on a sugar beet farm near Bonn in Germany over a three months period using a remote sensing JAI camera [10]. Figure 1 shows two sample images from this dataset. The dataset provides RGB images of a sugar beet field along with the segmented mask annotations of these images automatically generated using a U-Net architecture. Upon inspection it was revealed that the masks provided based using this approach were not labeled correctly and contained many errors. Consequently, 1600 images from this data set were labelled manually with a split of 1000 training images, 300 validation images, and 300 testing images. The labeled dataset can be accessed publicly on the following link: *https://github.com/Diaa340/Weed-Detection.git*

Figure 1: Sugarbeet Dataset samples

B. Sunflower Dataset

The original sunflower dataset contains 500 images collected through a UGV robot from a farm in Jesi, Italy, over a period of one month [11]. Figure 2 shows two samples from the sunflower data set. Like the sugar beet dataset, the mask annotations provided with the data set had inaccurate labeling and overlapped segments. Hence, this data was also manually labelled again and can be accessed publicly through the link provided earlier. The labelled data has 300 training images, 50 validation images, and 150 testing images.

Figure 2: Sunflower Dataset samples

C. Evaluation Metrics

Intersection over Union (IoU) is used to assess the accuracy of a detection algorithm's output bounding boxes around an object of interest in an image when compared to ground truth boxes. The perfect score is 1.00 which represents a complete overlap.

$$\text{IoU} = \frac{Area\ of\ overlap}{Area\ of\ intersection} \quad (1)$$

Mean Average Precision (mAP) is used to assess the overall quality of object detection models. This metric requires finding a model's average AP across its classes. The calculation of AP requires calculating a model's Precision and Recall, followed by drawing its precision-recall curve, and finally, finding the area under the curve.

$$\text{AP} = \int_0^1 P(R)dR \qquad \text{mAP} = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (2)$$

mAP can be calculated using various IoU thresholds. For example, using an IoU threshold of 0.7 means that if the IoU of an inferred bounding box with the ground truth is more than 0.7, then assigned label is considered. This paper uses two different IoU thresholds. A baseline IoU threshold of 0.50 is denoted as mAP@.5. Another common metric is to calculate mAP over a range of 0.5 to 0.95 IoU and provide the average. This metric is denoted as mAP@[0.5,0.95] and provides a more stable measure.

D. Models and architectures

Single Shot Detectors (SSD) use a fully convolutional approach where the neural network can find all objects within an image at one shot or a single forward pass using convolution nets. Single-shot algorithms are known for their higher accuracy and faster inference time. This paper explored four types of detectors for weed detection as described below.

1) You Only Look Once (YOLO)

YOLO architecture consists of a CNN where the backbone extracts the main features of an image, followed by a neck for feature map collection, and lastly, an object detection head, which is the final layer in the model. This paper explored using two state-of-the-art YOLO series models including YOLOv7 [12] and YOLOv8 [13]. The first model introduced an extended efficient layer aggregation network (E-ELAN) backbone, which is designed to improve accuracy and speed. YOLOv8, on the other hand, proposed the concatenation of features to reduce the number of parameters in the model and its overall size.

2) EfficientDet

EfficientDet [14] is an object detection model that uses an EfficientNet backbone pre-trained on the ImageNet dataset. The EfficientDet model was built with the objective of creating an object detection model that provides high accuracy while being lightweight and requiring few computational resources. The model introduces a method of compound scaling in which the backbone, feature network, and the detection head of the model are scaled up to optimize the model's accuracy and efficiency. The model also uses a Bidirectional Feature Pyramid Network (BiFPN) for multi-scale feature fusion. This network uses cross-scale connections to improve the feature model's efficiency and fast normalized fusion, which is a weighted fusion method used to enhance the discriminative power of the network's input feature representations.

3) RetinaNet

RetinaNet [15] is a one-stage object detector designed to deal with the problem of background-foreground class imbalance in the training of object detection models. To address this issue, RetinaNet utilizes a focal loss function. RetinaNet relies on a ResNet backbone. Similar to EfficientDet, it uses a feature pyramid network (FPN) to detect objects at multiple scales and produce representations proportional to the size of the feature maps. The model uses a classification subnet to predict the presence of objects in anchor boxes in each of the input image's spatial positions and a box regression subnet to predict the offsets of the generated bounding boxes relative to the model's anchor boxes.

4) Detection Transformer (DETR)

DETR [16] uses a CNN backbone to learn the 2D representations of an input image before applying positional encodings on the learned representations and feeding it to a transformer encoder-decoder, which uses self-attention layers that allow the model to capture and learn the relationships between different regions of an input image. Lastly, the output of the transformer is fed into a feed-forward network made out of a 3-layer perceptron, where the final bounding box predictions, as well as class label predictions, are made. In general, DETR is known to perform comparatively close to other object detectors. Nonetheless, detecting small objects poses an obstacle to transformer-based architectures in general.

IV. Training

Table II shows the models, their backbones and the hyperparameters used to train the various models on the newly labelled data sets. As the table shows, E-LAN backbones were used for both YOLO models while DETR used an encoder-decoder architecture. Similarly, EfficientDet used EfficientNet while RetinaNet used the classical ResNet architecture. The models were trained until the loss did not change for many episodes resulting in different episodes.

TABLE II. **Models and Hyperparameters**

Model	Backbone	Epi.	Optm.	Learning rate	Btc. size
YOLOv8	E-ELAN	25	SGD	0.01	16
YOLOv7	E-ELAN	55	Adam	0.01	8
DETR	CNN+ encoder-decoder transformer	20	Adam W	1e-4	2
EfficientDet	EfficentNet + BiFPN	30	Adam W	5e-3	8
RetinaNet	ResNet + FPN	30	Adam	1e-5	8

V. Results and Discussion

This section presents the findings of this study on the sugarbeet dataset and the sunflower dataset, followed by a discussion of the limitations. For comparison purpose, as Table III shows, the state-of-art segmentation methods on weed data sets, the accuracy varies between 70% to 95%. However, for the two original data sets of sunflowers and sugar beet modified for this paper, mAP@[0.5, 0.95] of 23% and 49% for weeds were achieved. It is worth mentioning that the original data labelling provided was found not to be reliable upon inspection as indicated earlier due to the way it was generated. Therefore, these results are as reliable as the mask labelling.

TABLE III. Previous Work on Weed Detection

Work	Dataset	Arch.	Best results
Arun et. al. [1]	CWFID	Reduced U-Net	95.52% overall accuracy
Sodjino et al. [2]	Maryland and leaf counting	U-Net	89.9% weed detection accuracy
Fawakherji et al. [3]	Sunflower and Sugarbeet	Bonnet	mAP@[0.5, 0.95] Sunflower: 86% on crops and 23% on weeds Sugar beet: 82% on crops and 49% on weeds

A. Sugarbeet Dataset

Figure 3 shows an example of a YOLO model detecting weeds and crops.

Figure 3: Sugarbeet Dataset Inference

The results obtained from the manually labelled sugar beet dataset showed that both YOLO models clearly outperformed other models. YOLOv7 achieved a mAP score of 81.6% on both classes with a 0.50 IoU threshold. Crops were detected with a 91.1% mAP, while weeds were detected at a lower rate of 72.2%. On the other hand, YOLOv8 slightly outperformed YOLOv7 in terms of mAP@[0.5,0.95] with a 57.6% mAP overall in comparison to 54.9% mAP overall for YOLOv7. EfficientDet outscored both DETR and RetinaNet. Nonetheless, does not perform as well as the advanced YOLO architectures.

Table IV summarizes the results of all models tested on this dataset. It is worth mentioning that all models managed achieved higher mAP scores on detecting crops than weeds. This can perhaps be explained by the fact that weeds are usually smaller in size and have a less obvious shapes than crops, which tend to be bigger and more uniformly shaped. In addition, weeds are usually present at ground level which makes it difficult to distinguish weeds from the background. This can also help explain the high false positives. This can explain the high false positive rates between weeds/crops and the background as seen in Figure 4.

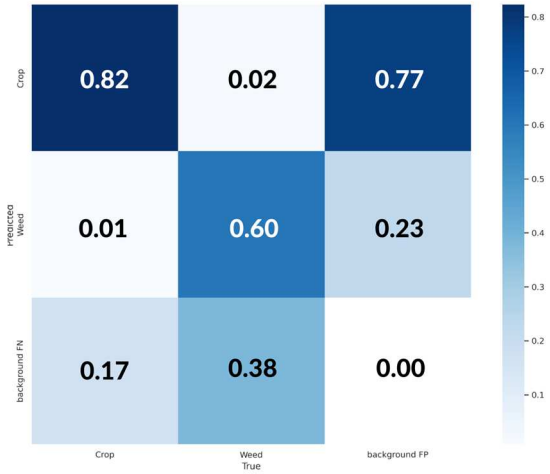

Figure 4: Sugarbeet Dataset confusion matrix (YOLOv7)

B. Sunflower Dataset

Figure 5 shows an example of a weed and crop being detected using YOLO on the Sunflower dataset.

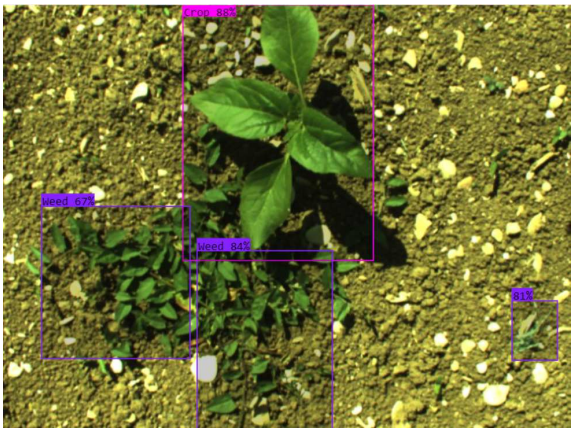

Figure 5: Sunflower Dataset Inference

The results obtained from the sunflower dataset follow a similar pattern to that obtained on the previous dataset. However, for this data set, YOLOv8 outperformed YOLOv7 for the IoU thresholds. As shown in Table V YOLOv8 had a mAP score of 77% among all classes with a 97.1% mAP score on crops and 56.8% on weeds at a threshold of 0.50.

TABLE IV. **SUGARBEET RESULTS**

Model	Class	mAP @.5	mAP @[.5,.95]
YOLOv7	All	**0.816**	0.549
	Crop	**0.911**	0.73
	Weed	**0.722**	0.367
YOLOv8	All	0.813	**0.576**
	Crop	0.910	**0.740**
	Weed	0.716	**0.413**
DETR	All	0.353	0.229
	Crop	0.564	0.336
	Weed	0.142	0.122
EfficientDet	All	0.664	0.443
	Crop	0.883	0.675
	Weed	0.445	0.221
RetinaNet	All	0.392	0.371
	Crop	0.752	0.702
	Weed	0.032	0.032

As Table V shows, DETR and EfficientDet scored relatively lower mAP scores with an overall mAP at 0.5 IoU threshold of almost 63.9% and 38.6% on the 0.95 threshold. Again, weed detection seemed to be more challenging than crop detection as all models resulted in lower detection rates of weed in comparison to crops. As stipulated earlier, this can be a result of the weed size, geometry, and high interleaving.

TABLE V. SUNFLOWER RESULTS

Model	*Class*	*mAP @.5*	*mAP @[0.5,.95]*
Yolo v7	All	0.652	0.439
	Crop	0.956	0.731
	Weed	0.348	0.147
Yolo v8	All	**0.770**	**0.517**

Model	*Class*	*mAP@.5*	*mAP @[0.5,.95]*
	Crop	**0.971**	**0.772**
	Weed	**0.568**	**0.262**
DETR	**All**	0.430	0.256
	Crop	0.639	0.386
	Weed	0.221	0.126
EfficientDet	**All**	0.437	0.266
	Crop	0.643	0.398
	Weed	0.231	0.134
RetinaNet	**All**	0.370	0.344
	Crop	0.726	0.582
	Weed	0.015	0.012

As shown in Figure 6, the sunflower dataset showed lower rates of false positive labels of background as opposed to sugar beets. This can perhaps be explained by lesser number of stones and wood in the background within the sunflower dataset in comparison to the sugar beet dataset. Presence of such features decreases the model performance especially in detecting smaller objects.

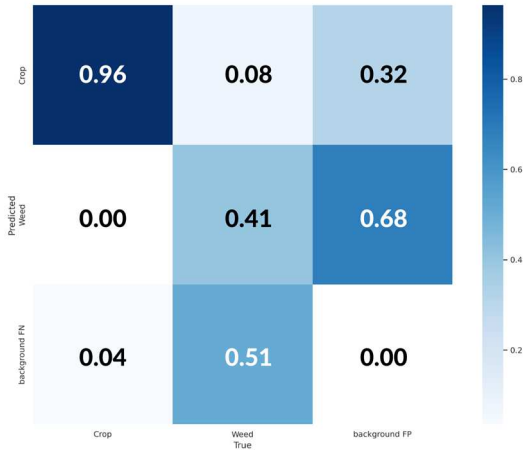

Figure 6: Sunflower Dataset confusion matrix (YOLOv7)

VI. Conclusion and Future Work

In this paper, two newly labeled datasets derived from the sugarbeet and sunflower datasets [10], [11] was introduced. Further, different state-of-the-art single-stage detectors, including YOLO series, EfficientDet, RetinaNet, and DETR were trained on these data sets. YOLO series was found to be superior in detection tasks, while other architectures were prone to higher error rates while detecting small objects such as weeds. Future work includes data augmentation, oversampling, and synthesizing of weeds to improve detection performance. In addition, deep learning models can be optimized and deployed on UGVs and UAVs to test inference time and computational expenses for realtime detection tasks.

VII. References

[1] "Weed management in wheat fields in the cold winter desert of Uzbekistan", Food and Agricultural Organization of the United Nations Report, February, 2023.

[2] R. A. Arun, S. Umamaheswari, and A. V. Jain, "Reduced U-Net Architecture for Classifying Crop and Weed using Pixel-wise Segmentation," in *2020 IEEE International Conference for Innovation in Technology (INOCON)*, Nov. 2020, pp. 1–6. doi: 10.1109/INOCON50539.2020.9298209.

[3] S. G. Sodjinou, V. Mohammadi, A. T. Sanda Mahama, and P. Gouton, "A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images," *Inf. Process. Agric.*, vol. 9, no. 3, pp. 355–364, Sep. 2022, doi: 10.1016/j.inpa.2021.08.003.

[4] M. Fawakherji, C. Potena, A. Pretto, D. D. Bloisi, and D. Nardi, "Multi-Spectral Image Synthesis for Crop/Weed Segmentation in Precision Farming," *Robot. Auton. Syst.*, vol. 146, p. 103861, Dec. 2021, doi: 10.1016/j.robot.2021.103861.

[5] V. Lakshmanan, M. Görner, and R. Gillard, *Practical Machine Learning for Computer Vision*. O'Reilly Media, Inc., 2021.

[6] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, Jan. 2022, doi: 10.1016/j.procs.2022.01.135.

[7] Y. Dandekar, K. Shinde, J. Gangan, S. Firdausi, and S. Bharne, "Weed Plant Detection from Agricultural Field Images using YOLOv3 Algorithm," in *2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA*, Aug. 2022, pp. 1–4. doi: 10.1109/ICCUBEA54992.2022.10011010.

[8] V. Czymmek, L. O. Harders, F. J. Knoll, and S. Hussmann, "Vision-Based Deep Learning Approach for Real-Time Detection of Weeds in Organic Farming," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, May 2019, pp. 1–5. doi: 10.1109/I2MTC.2019.8826921.

[9] P. Kumar Doddamani and G. P. Revathi, "Detection of Weed & Crop using YOLO v5 Algorithm," in *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, Oct. 2022, pp. 1–5. doi: 10.1109/MysuruCon55714.2022.9972386.

[10] "Sugar Beets 2016 – StachnissLab." https://www.ipb.uni-bonn.de/data/sugarbeets2016/ (accessed Mar. 14, 2023).

[11] "Sunflower Dataset." http://www.diag.uniroma1.it/~labrococo/fsd/sunflowerdatasets.html (accessed Mar. 14, 2023).

[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." arXiv, Jul. 06, 2022. doi: 10.48550/arXiv.2207.02696.

[13] "YOLOv8 Docs." https://docs.ultralytics.com/ (accessed Mar. 14, 2023).

[14] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection." arXiv, Jul. 27, 2020. Accessed: Mar. 14, 2023. [Online]. Available: http://arxiv.org/abs/1911.09070

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection." arXiv, Feb. 07, 2018. Accessed: Mar. 14, 2023. [Online]. Available: http://arxiv.org/abs/1708.02002

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers." arXiv, May 28, 2020. Accessed: Mar. 14, 2023. [Online]. Available: http://arxiv.org/abs/2005.12872