Selcuk Journal of Agriculture and Food Sciences

http://sjafs.selcuk.edu.tr/sjafs/index

Research Article

SJAFS
(2022) 36 (Special Issue),53-58
e-ISSN: 2458-8377
DOI:10.15316/SJAFS.2022.078

Comparison of Plant Detection Performance of CNN-based Single-stage and Two-stage Models for Precision Agriculture

Recai ÖZCAN[1,*], Kemal TÜTÜNCÜ[2], Murat KARACA[3]

[1]Selçuk University, Bozkır Vocational School, Department of Electricity and Energy, Konya, Türkiye
[2]Selçuk University, Faculty of Technology, Department of Electrical Electronics Engineering, Konya, Türkiye
[3]Selçuk University, Faculty of Agriculture, Department of Plant Protection, Konya, Türkiye

ARTICLE INFO

Article history:
Received date: 02.12.2022
Accepted date: 18.12.2022

Keywords:
Precision Agriculture
Plant Detection
SSD
Faster R-CNN
Performance Evaluation

ABSTRACT

The fact that arable land is not increasing in proportion to the ever-increasing population will increase the need for food in the coming years. For this reason, it is necessary to increase the yield of crops to make optimum use of arable land. One of the most important reasons for the decrease in yield and quality of crops is weeds. Herbicides are generally preferred for weed management. Due to deficiencies in herbicide application methods, only 0.015-6% of herbicides reach their target. The use of herbicides, which is an important part of the agricultural system, is an issue that needs to be emphasized, considering the risk of residue and environmental damage. In parallel with the rapid development of electronic and computer technologies, artificial intelligence applications have had the opportunity to develop. In this context, the use of artificial intelligence for plant detection in the subsystems of herbicide application machines will contribute to the development of precision agriculture techniques. In this study, the plant detection performances of single-stage and two-stage Convolutional Neural Network (CNN)-based deep learning (DL) models are evaluated. In this context, a dataset was created by taking images of *Zea mays*, *Rhaponticum repens* (L.) Hidalgo, and *Chenopodium album* L. plants in agricultural lands in Konya. With this dataset, the training of the models was carried out by the transfer learning method. The evaluation metrics of the trained models were calculated using the error matrix. In addition, training time and prediction time were used as quantitative metrics in the evaluation of the models. The plant detection performance, training time, and prediction time of the models were 85%, 8 h, 1.21 s for SSD MobileNet v2 and 99%, 22 h, 2.32 s for Faster R-CNN Inception v2, respectively. According to these results, Faster R-CNN Inception v2 is outperform in terms of accuracy. However, in cases where training time and prediction time are important, the SSD MobileNet v2 model can be trained with more data to increase its accuracy.

1. Introduction

The most preferred method for weed management is the use of herbicides (Ali et al., 2015). It is also observed that herbicides are used in more than 90% of the total area planted for maize in European Union countries (Vasileiadis et al., 2015). Although weeds do not cover the entire soil surface, the most common use of herbicides is to apply them over the entire area. It is obvious that if herbicide use is not optimized, various environmental and economic risks are evident (Pérez-Ortiz et al., 2016; Zheng et al., 2017). Traditional methods of weed control are limited in terms of time, cost, and errors. These limitations can be overcome with the use of computer vision systems. In this respect, the use of herbicides in weed control can achieve better results if they target only weeds and are applied selectively according to a specific weed class (dos Santos Ferreira et al., 2017). Variable rate herbicide application is also recommended when it comes to herbicide use in precision agriculture practices (Asad et al., 2020; Bàrberi, 2002). The most important step to realizing weed control with an automated system within the framework of precision agriculture is to detect weeds correctly (Liu and Bruch, 2020).

In this study, the performance evaluation of Single Shot Multibox Detector (SSD), a two-stage detector that

* Corresponding author email: *recaiozcan@selcuk.edu.tr*

can be used in the subsystems of robotic machines for automatic weed control, and Faster R-CNN, a two-stage detector, was carried out with a dataset created with plant images taken from Konya plain agricultural lands.

The study is organized as follows. The second section of the paper describes the two-stage and single-stage detectors. The third section describes the dataset, object labeling, model training, performance metrics, and experimental results. In the last section, the results are evaluated, and future work is presented.

2. Materials and Methods

State-of-the-art DL algorithms in object detection generally consist of two main parts: object detectors and backbone architectures. In an object detection network, backbone architectures extract features from input images. The quality of the extracted features directly affects the performance of the algorithm. Object detectors, on the other hand, perform classification based on the extracted features and identify bounding boxes. Object detectors can be categorized into two main categories as two-stage and one-stage according to the object detection method. In the first stage, two-stage detectors use a region proposal network (RPN) to identify regions that are likely to contain objects. In the second stage, classification and object location are determined. In single-stage detectors, these processes are performed at one time (Zaidi et al., 2022). This provides an advantage in terms of speed (Jiao et al., 2019). The basic architectures of two-stage and single-stage detectors are shown in Figure 1.

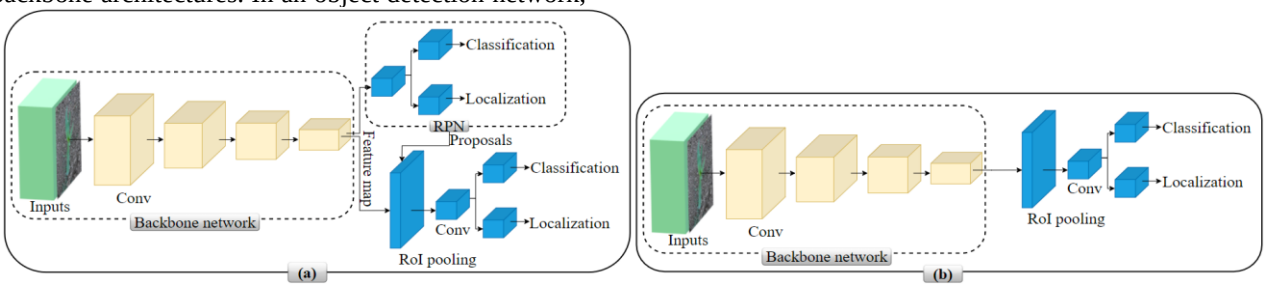

Figure 1
Basic structure of detectors (a) Two-stage detectors, (b) Single-stage detectors (Jiao et al., 2019)

2.1. Faster R-CNN Inception v2

In the CNN algorithm, creating boxes of different sizes for objects of different sizes and detecting the region proposal each time by shifting the boxes on the image would require too much computer power and too much processing. For this reason, different R-CNN algorithms have been developed. Within CNN algorithms, the region suggestion that is likely to be an object in an image is used in R-CNN algorithms.

In R-CNN, region proposals are created by the selective search in which the same class value is assigned to the pixels in each closed area separated from each other in shape or color in an image that is likely to contain objects in the images to be region proposals. The marked regions are brought to the same size and passed through the CNN one by one to check whether there is an object in that region and if so, the detected object is assigned to a class. The region boundaries are defined by linear regression based on the relationship between the dependent variable and independent variables, and the classification is based on a supervised learning method, namely the Support Vector Machine (SVM), which is based on a model created using known data with known results, and is based on the determination of decision boundaries or, in other words, hyperplanes to optimally separate the data belonging to the classes from each other (Girshick et al., 2014). The training and prediction times of R-CNN are very long as all the region proposals are passed through the CNN one by one.

Fast R-CNN, which has the same structure as R-CNN but uses a different technique to speed it up, has been developed. In Fast R-CNN, the entire image is directly passed through the CNN without identifying any region in the image and a high-resolution feature map is extracted. From this feature map, a region proposal is generated by the selective search. Again, as in R-CNN, the marked regions are resized to the same size and given directly to the Fully Connected Layer (FCL) this time. In this way, each region is not passed through the neural network separately as in R-CNN. In this case, excessive time loss is avoided. Classification is then performed, and the boundaries of the detected object are determined. The boundaries are determined by linear regression, and the classification is performed using the SoftMax function, which is used for multiple classification problems and produces outputs between 0 and 1, indicating the probability that each given input belongs to a class (Girshick et al., 2015). Fast R-CNN works quite fast in the training phase. However, it spends most of the time in the testing phase making region proposals.

Faster R-CNN was developed to reduce the time spent on region recommendation, which is the disadvantage of Fast R-CNN and to make the model work even faster. Faster R-CNN (Ren et al., 2015) gains speed by making the region suggestion within the network instead of getting region suggestions with Selective Search. After applying CNN to the input image, the feature map is extracted. Then, unlike Fast R-CNN, it makes region proposals through RPN. After RPN identifies the regions, the rest of the operations are

the same as Fast R-CNN. After the identified regions are resized, they are given to the FCL, and classification is performed. In this case, it is necessary to train both the RPN and the CNN.

The RPN has two tasks: to decide whether there are objects in each proposal region and to determine the box size of proposals. CNN, on the other hand, has two tasks: to perform classification in the region and determine the boundaries of the object after classification.

Inception v2 (Szegedy et al. 2016) is one of the widely used and highly accurate CNN architectures. This architecture is designed to avoid the complexity of CNNs by reducing the depth of the network.

2.2. SSD MobileNet v2

It was mentioned above that in the faster R-CNN, regions in the image that are likely to be objects are first identified and then classified using FCL. In SSD, on the other hand, these two are done at one time. An image is taken as input and passed through the CNN to obtain feature maps of different sizes. In all feature maps, bounding rectangles are obtained with the help of a 3x3 convolutional filter. For each rectangle, both boundaries and classifications are determined simultaneously. Since these rectangles are present in every activation map, they can recognize both small objects and large objects. During training, the correct boundaries are compared with the predicted boundaries. The best rectangles predicted above 50% are labeled as positive.

MobileNet v2 (Sandler et al., 2018) is designed to improve computational cost and accuracy by adding jump links between the convolution layers in MobileNet v1 (Howard et al., 2017), a lightweight network backbone that uses in-depth convolutions developed for embedded and mobile vision applications.

2.3. Experimental Design and Statistical Analysis

2.3.1. Dataset

A data set was created from the images obtained by using mobile phones and cameras in Konya and its surrounding districts. A total of 1500 images were selected from this data set by taking 500 images each from the images of *Zea mays*, *Rhaponticum repens* (L.) Hidalgo, and *Chenopodium album* L. plants, which are given in Table 1 below and whose sample images are shown in Figure 2. These selected images were resized to 640 pixels on the short side so that the aspect ratios would not be distorted for fast training. Of the total 500 images of each type, 80% were used for training and 20% for testing. Thus, 1200 images were used for training and 300 images were used for testing.

Table 1
Created dataset

Species	Scientific Names	EPPO Code	Training Data		Total
			Train (%80)	Validation (%20)	
1	*Zea Mays*	ZEAMX	400	100	
2	*Chenopodium album* L.	CHEAL	400	100	1500
3	*Rhaponticum repens* (L.) Hidalgo	CENRE	400	100	

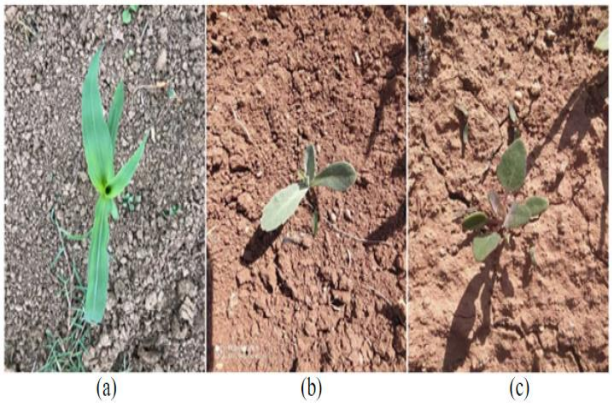

Figure 2
Example dataset images; (a) *Zea Mays*, (b) *Rhaponticum repens* (L.) Hidalgo, (c) *Chenopodium album* L.

2.3.2. Labeling

For labeling processes, xml files were created by labeling all images using the open-source LabelImage (Tzutalin, 2015) tool, an example of which is shown in Figure 3.

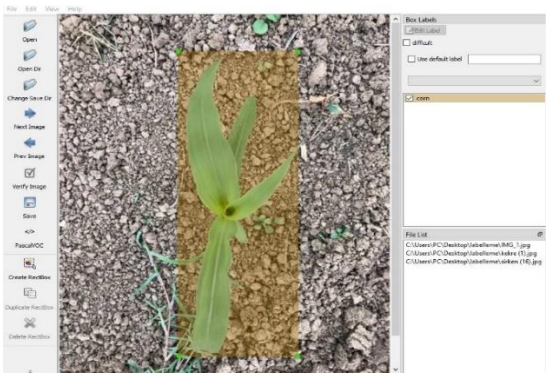

Figure 3
Example labeling for *Zea Mays*

2.3.3. Training Processes

Since building and training, a model from scratch would require a lot of time and powerful computers, two models were created using transfer learning, a method in which a pre-trained model for one task is redesigned for a second task. Tensorflow, an open-source DL library, was used in both models. The preferred models were pre-trained with the Tensorflow Object Detection API (Tensorflow, 2021) using the Common Objects in Context (COCO) dataset consisting of 91 object types

and 2.5 million labeled data. We retrained these models by transfer learning using our dataset and object detection was performed. Thus, we were able to observe the successful results by training on a low-equipped computer with little data. For transfer learning using pre-trained models:

- The coefficients of the convolution layers were fixed,
- The number of output neurons activated by the activation function in the output layer was changed to match the desired number of classes,
- The network coefficients of the FCL between the output layer and the convolution layers were retrained with new data with random initial values.

The training times of the Faster R-CNN and SSD models trained for 10,000 steps using the generated dataset and Intel Core i5 3230M 2.60 GHz processor are shown in Table 2 and the training graphs are shown in Figure 4.

Table 2
Training times of the models

Model	Training steps	Training time (h)
Faster R-CNN Inception v2	10.000	22
SSD MobileNet v2	10.000	8

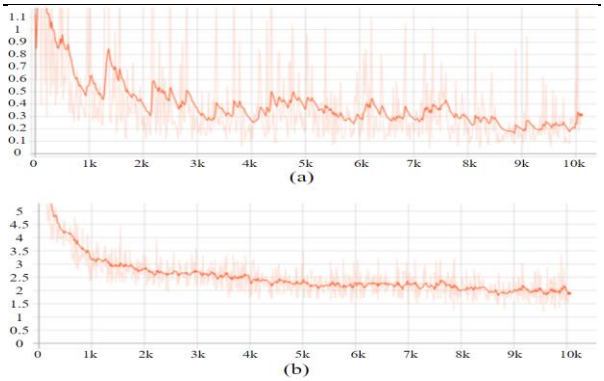

Figure 4
Total loss graphs of the models according to iteration; (a) Faster R-CNN Inception v2, (b) SSD MobileNet v2

2.3.4. Performance Metrics

The confusion matrix allows us to find the correlation between model performance and test results. This matrix provides information about the correct or incorrect classification of positive and negative samples. Table 3 shows a two-class confusion matrix.

Table 3
Two-class confusion matrix

	Prediction	
	Positive	Negative
Positive	TP (True Positive)	FN (False Negative) (Type-2 error)
Negative	FP (False Positive) (Type-1 error)	TN (True Negative)

The elements of the confusion matrix can be defined as follows:

- TP: The model correctly predicted the positive class as a positive class.
- FP: The model incorrectly predicted the negative class as a positive class.
- FN: The model incorrectly predicted the positive class as a negative class.
- TN: The model correctly predicted the negative class as a negative class.

The three-class confusion matrix is given in Table 4 and the calculation of the matrix elements is given in Table 5.

Table 4
Three-class confusion matrix

		Prediction		
		C_1	C_2	C_3
Actual	C_1	T_1	F_{12}	F_{13}
	C_2	F_{21}	T_2	F_{23}
	C_3	F_{31}	F_{32}	T_3

Table 5
TP, TN, FP, and FN calculation for a three-class confusion matrix

Class	TP	TN	FP	FN
C_1	$TP_1=T_1$	$TN_1=T_2+T_3+F_{23}+F_{32}$	$FP_1=F_{21}+F_{31}$	$FN_1=F_{12}+F_{13}$
C_2	$TP_2=T_2$	$TN_2=T_1+T_3+F_{13}+F_{31}$	$FP_2=F_{12}+F_{32}$	$FN_2=F_{21}+F_{23}$
C_3	$TP_3=T_3$	$TN_3=T_1+T_2+F_{12}+F_{21}$	$FP_3=F_{13}+F_{23}$	$FN_3=F_{31}+F_{32}$

Statistical calculations can be made using the TP, FP, FN, and TN values in the confusion matrix (Martínez et al., 2022). In this study, the precision, recall, accuracy, and F1 score in Table 6 were determined as the performance evaluations of the models.

Table 6
Performance metrics

Metrics	Formula	Description
Precision	TP / (TP + FP)	Accuracy rate of positive predictions
Recall	TP / (TP + FN)	Accuracy rate of true positives
Accuracy	(TP + TN) / (TP + TN + FP + FN)	Gives the success of the model
F1 Score	(2*TP) / (2*TP + FP + FN)	It is the harmonic mean of the precision and recall values. Includes all error costs.

3. Results

Training and model tests were performed using the same computer. A total of 75 images were created by taking 25 images from each class that the model had not seen before, and the test data were created by resizing the short side of the image to 640 pixels so that the aspect ratio would not be distorted as it was done when creating the training and test data.

For both Faster R-CNN Inception v2 and SSD MobileNet v2 models, the threshold value was chosen as 0.5. That is, 50% similarity and above is considered a correct prediction. Also, predictions above the bounding box threshold (IoU) of 0.5 were considered valid. The same test data was given as input to both models and the error matrix for the three classes in Table 7 was created.

Table 7
Confusion matrices of models

	Prediction					
	Faster R-CNN Inception v2			SSD MobileNet v2		
	ZEAMX	CHEAL	CENRE	ZEAMX	CHEAL	CENRE
ZEAMX	25	0	0	23	2	0
CHEAL	0	25	0	0	25	0
CENRE	0	1	24	0	15	10

The calculations in Table 5 were made and the statistical results in Table 8 were obtained by species.

Table 8
Performance of models by species

Metrics	Faster R-CNN Inception v2			SSD MobileNet v2		
	ZEAMX	CHEAL	CENRE	ZEAMX	CHEAL	CENRE
Precision	1	0,96	1	1	0,6	1
Recall	1	1	0,96	0,92	1	0,4
Accuracy	1	0,99	0,99	0,97	0,77	0,8
F1 Score	1	0,98	0,98	0,96	0,75	0,57

The average of the results for the final evaluation is shown in Table 9.

Table 9
Overall performance of the models

Metrics	Faster R-CNN Inception v2	SSD MobileNet v2
Precision	0,99	0,87
Recall	0,99	0,77
Accuracy	0,99	0,85
F1 Score	0,99	0,76

Examples of the prediction results of the models on the test dataset images are given in Figure 5.

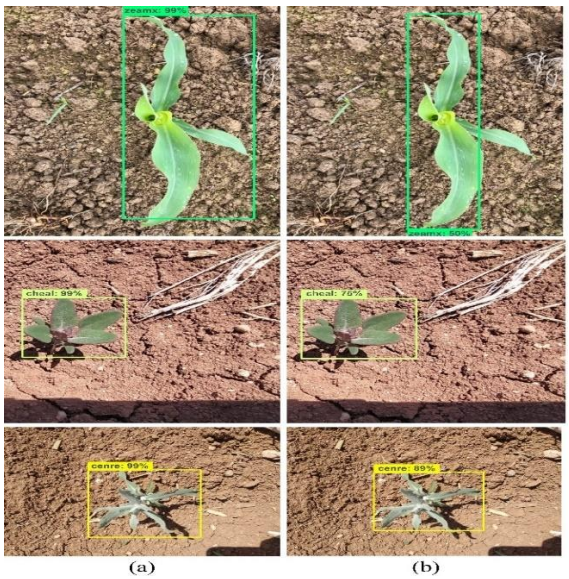

Figure 5
Predictions of the models; (a) Faster R-CNN Inception v2, (b) SSD MobileNet v2

Prediction time was also used as a quantitative metric in model evaluation. The prediction times of the models per image are given in Table 10.

Table 10
Prediction times of models

Model	Prediction time per image (sec)
Faster R-CNN Inception v2	2,32
SSD MobileNet v2	1,21

4. Discussion

In this study, CNN-based single-stage and two-stage models that can be used in the subsystems of robotic herbicide application machines were evaluated. In this context, a data set was created by capturing images of *Zea mays*, and *Rhaponticum repens* (L.) Hidalgo, and *Chenopodium album* L. plants in Konya plain agricultural lands. Using this dataset, SSD MobileNet v2, a single-stage detector, and Faster R-CNN Inception v2, a two-stage detector, were trained. Models were given images they had not seen before, and performance evaluations were made. The accuracy and F1 score scores of the two-stage detector were found to be 0.99. The accuracy and F1 score of the single-stage detector were 0.85 and 0.76, respectively. In terms of time, the two-stage detector has a training time of 22 h and a prediction time of 2.32 s, while the single-stage detector has a training time of 8 h and a prediction time of 1.21 s. According to these results, the performance of the two-stage detector is high. However, the single-stage detector performs better in terms of training and prediction time. The single-stage detector is therefore preferable where time is of the essence. In addition, training with more data and/or using data augmentation techniques can improve the performance of the single-stage detector to the desired level. In this study, promising results were obtained by evaluating a dataset containing a few images. In our future studies, we plan to investigate high-performance methods for plant detection for precision agriculture applications.

5. References

Ali A, Streibig JC, Christensen S, Andreasen C (2015). Image-based thresholds for weeds in maize fields. *Weed Research* 55(1): 26-33.
https://doi.org/10.1111/wre.12109

Asad MH, Bais A (2020). Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network. *Information Processing in Agriculture* 7(4): 535-545.
https://doi.org/10.1016/j.inpa.2019.12.002

Bàrberi PAOLO (2002). Weed management in organic agriculture: are we addressing the right issues? *Weed Research* 42(3): 177-193.
https://doi.org/10.1046/j.1365-3180.2002.00277.x

dos Santos Ferreira A, Freitas DM, Silva GG, Pistori H, Folhes MT (2017). Weed detection in soybean crops using ConvNets. *Computers and Electronics in Agriculture* 143: 314-324.
https://doi.org/10.1016/j.compag.2017.10.027

Girshick R, Donahue J, Darrell T, Malik J (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, June 2014, pp. 580-587.

Girshick R (2015). Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision*, December 2015, Piscataway, New Jersey, United States, pp. 1440-1448.

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

Jiao L, Zhang F, Liu F, Yang S, Li L, Feng Z, Qu R (2019). A survey of deep learning-based object detection. *IEEE access* 7: 128837-128868.

Liu B, Bruch R (2020). Weed Detection for Selective Spraying: a Review. *Current Robotics Reports* 1(1): 19-26.
https://doi.org/10.1007/s43154-020-00001-w

Martínez SS, Gila DM, Beyaz A, Ortega JG, García JG (2018). A computer vision approach based on endocarp features for the identification of olive cultivars. *Computers and Electronics in Agriculture* 154: 341-346.
https://doi.org/10.1016/j.compag.2018.09.017

Pérez-Ortiz M, Peña JM, Gutiérrez PA, Torres-Sánchez J, Hervás-Martínez C, López-Granados F (2016). Selecting patterns and features for between- and within- crop-row weed mapping using UAV-imagery. *Expert Systems with Applications* 47: 85-94.
https://doi.org/10.1016/j.eswa.2015.10.043

Ren S, He K, Girshick R, Sun J (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28.

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, June 2018, pp. 4510-4520.

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016). Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, June 2016, pp. 2818-2826.

Tensorflow (2021), TensorFlow 2 Detection Model Zoo, https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md (Access date: 4 May 2022).

Tzutalin. LabelImg. Git code (2015).
https://github.com/heartexlabs/labelImg (Access date: 18 Sep 2022).

Vasileiadis VP, Otto S, Van Dijk W, Urek G, Leskovšek R, Verschwele A, Furlan L, Sattin M (2015). On-farm evaluation of integrated weed management tools for maize production in three different agro-environments in Europe: Agronomic efficacy, herbicide use reduction, and economic sustainability. European Journal of Agronomy 63: 71-78.
https://doi.org/10.1016/j.eja.2014.12.001

Zaidi SSA, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing* 126: p. 103514.
https://doi.org/10.1016/j.dsp.2022.103514

Zheng Y, Zhu Q, Huang M, Guo Y, Qin J (2017). Maize and weed classification using color indices with support vector data description in outdoor fields. *Computers and Electronics in Agriculture* 141; 215-222.
https://doi.org/10.1016/j.compag.2017.07.028