Novel Assessment of Region-Based CNNs for Detecting Monocot/Dicot Weeds in Dense Field Environments

Nima Teimouri 1,*, Rasmus Nyholm Jørgensen 1 and Ole Green 1,2

1 Agro Intelligence ApS, Agro Food Park 13, 8200 Aarhus, Denmark; rnj@agrointelli.com (R.N.J.); olg@agrointelli.com (O.G.)
2 Department of Agroecology, Aarhus University, 8000 Aarhus, Denmark
* Correspondence: nte@agrointelli.com

Abstract: Weeding operations represent an effective approach to increase crop yields. Reliable and precise weed detection is a prerequisite for achieving high-precision weed monitoring and control in precision agriculture. To develop an effective approach for detecting weeds within the red, green, and blue (RGB) images, two state-of-the-art object detection models, EfficientDet (coefficient 3) and YOLOv5m, were trained on more than 26,000 in situ labeled images with monocot/dicot classes recorded from more than 200 different fields in Denmark. The dataset was collected using a high velocity camera (HVCAM) equipped with a xenon ring flash that overrules the sunlight and minimize shadows, which enables the camera to record images with a horizontal velocity of over 50 km h⁻¹. Software-wise, a novel image processing algorithm was developed and utilized to generate synthetic images for testing the model performance on some difficult occluded images with weeds that were properly generated using the proposed algorithm. Both deep-learning networks were trained on in-situ images and then evaluated on both synthetic and new unseen in-situ images to assess their performances. The obtained average precision (AP) of both EfficientDet and YOLOv5 models on 6625 synthetic images were 64.27% and 63.23%, respectively, for the monocot class and 45.96% and 37.11% for the dicot class. These results confirmed that both deep-learning networks could detect weeds with high performance. However, it is essential to verify both the model’s robustness on in-situ images in which there is heavy occlusion with a complicated background. Therefore, 1149 in-field images were recorded in 5 different fields in Denmark and then utilized to evaluate both proposed model’s robustness. In the next step, by running both models on 1149 in-situ images, the AP of monocot/dicot for EfficientDet and YOLOv5 models obtained 27.43%/42.91% and 30.70%/51.50%, respectively. Furthermore, this paper provides information regarding challenges of monocot/dicot weed detection by releasing 1149 in situ test images with their corresponding labels (RoboWeedMap) publicly to facilitate the research in the weed detection domain within the precision agriculture field.

Keywords: deep learning; weed detection; synthetic image; computer vision

1. Introduction

With global population growth, the demand for higher productivity in farmlands has been gradually increased [1]. Weed emergence is one of the main challenges involved in increasing crop yields. Weeds grow randomly through fields and compete with crops [2]. On the other hand, non-target pesticides not only immensely contaminate the environment but also lead to biodiversity loss [3]. The studied economic losses due to weed competition in different countries emphasize that the weeds have an important impact important on increasing the yield [4]. Therefore, weed management plays a fundamental role in increasing yields and consequently increasing revenue.

Manually mapping the weed population through hectares of fields is laborious, time-consuming, and often inaccurate. Moreover, traditional weed management is an economically wasteful procedure [5]. There are three weed management approaches in state-of-the-art research: physical, biological, and chemical approaches [6]. In all methods of weed...
management, extracting information about weed types, amount, growth stage, etc., plays an important role. Gathering this information, a precise map of the farmland is generated that is productive for site-specific weed management. The site-specific weed management results in a lower and/or optimal herbicide mixture and dose that is both efficient and environmentally friendly. State-of-the-art technologies mainly focus on autonomously detecting weeds with higher precision and lower contamination [7,8].

The use of advanced computer vision and deep learning techniques has been significantly developed by improvements in graphics processing units (GPUs) [9]. GPUs allow deep learning networks to learn from a large amount of data, which is essential for obtaining high accuracy levels [10]. Furthermore, recently, computer vision and deep learning technologies show promising performances in classification and detection problems. In the agriculture domain specifically, proposed methods come with a remarkable interpretation of weeds (including type and amount) [11–13]. Also, a variety of visual characteristics have been used in weed detection [14], weed mapping [15], and classification [16] using various computer vision and deep learning methods.

There are three main methods of object detection in computer vision: semantic segmentation, instance detection, and bounding-box detection. Semantic segmentation particularly shows outstanding performances in predicting the biomass composition due to classifying images pixel-wise [17]. However, the pixel-wise classification is required for highly accurate weed control methods such as physical management. In fields with heavy occlusion, counting the number of weeds is also challenging [18]. Therefore, instance segmentation is utilized in fine weed detection applications such as electrification-type weeding to also extract the number of weeds in each square meter [19]. In weed detection using bounding boxes, weed locations, and species are estimated [20]. Furthermore, weed’s growth stages are predicted based on the size of bounding boxes [21]. However, the performance of the object detection networks highly depends on the extracted features from different backbone layers [22]. Detecting weeds using a bounding box is studied in this paper since it is the core information base for integrated weed management (IWM) optimization.

Since convolutional neural networks specifically show outstanding performances in the computer vision domain, there are high tendencies in state-of-the-art papers to detect and classify weeds using deep learning algorithms [9,23]. While deep learning technology highly depends on a great number of images, data labeling is a time-consuming procedure [24]. On the other hand, class-imbalanced datasets bias the network toward the class with the majority of instances that lead to more uncertainty for the model [25]. Synthetically generated data improves the performance of the model by increasing labeled data, makes a balance between the number of images in different classes, and leads to a model with better generalization on real in-field images.

Synthetic data with high similarities to real in-field images are a good alternative or supplement for real in-field data without labels. General proposed researches based on synthetic data are categorized into three groups with respect to how the synthetic data is generated: cut-and-paste method, graphical modeling, and generative adversarial networks (GAN) [26]. In [27], the selected samples for generating the data are randomly rotated, scaled, and saturated before inserting into the soil background. The Gaussian kernel was adopted to generate the shadow and make the illusion of depth in the simulated images. The cut-and-paste method is a simple but effective method for generating synthetic data [28]. In the second category, graphical modeling (such as the L-system) is utilized to generate synthetic images [29]. In the third category, two networks are utilized to form the synthetic data: one as a generator and the other as a discriminator to remove poor samples [30]. However, GAN has considerably low accuracy compared with the two previous methods.

Many of the proposed plant detection algorithms work for cases wherein plants were well displayed and there was no occlusion [2], and one of the noticeable challenges in the agricultural dataset is plant occlusion [31,32]. Plants and weeds have either partial or heavy occlusion which is necessary to be considered in synthetic data generation [33].
Otherwise, the model is not robust enough to work fine on the real in-field images. Many of the proposed approaches have been evaluated on the images without occlusion or only low level of occlusion. Therefore, it is still challenging to assess the performance of plant detection algorithms on the images that plants are totally occluded as these cases are close to the in-field situation.

Therefore, the objective of this research was to determine the applicability of region-based CNN in detecting monocot/dicot weeds within in-field environments and highly occluded images which are necessary for the assessment of weeds development. In order to reach this target, two deep learning object detection networks were employed to identify the weeds and recognize them from the present dense crops within the images. It is also important to mention that the focus of this study was to develop the entire chain from recording in-situ and generating simulated images plus employing two deep learning object detection networks, EfficientDet (coefficient 3) and YOLOv5, to detect and recognize weeds within highly occluded images.

This paper is organized as follows. The dataset section is presented in Section 2. The synthetic data generation is discussed in Section 3.1. The supervised learning and neural networks are presented in Sections 3.2 and 3.3, respectively. The two deep-learning techniques, EfficientDet and YOLOv5, are described in detail in Sections 3.3.1 and 3.3.2. Furthermore, the evaluation procedure and the used metrics are discussed in Section 3.4. Finally, both the quantitative analysis and the two test scenarios are defined in Section 4.

2. Dataset

The dataset comprises images from two sources: handheld consumer cameras and images from a driving vehicle. The images from the handheld cameras allow one to cherry-pick locations in fields with interesting weeds, while the camera on the vehicle allows one to cover large areas and create unbiased measurements. The camera on the vehicle is a high-speed camera that was built by Aarhus University. This camera allows for automatically collecting images at high speed while traversing fields on an All-Terrain Vehicle (ATV) Figure 1A. The camera is equipped with a xenon ring flash that overrules the sunlight and minimize shadows, which enables the camera to record images with a horizontal velocity of over 50 km h\(^{-1}\). To trigger the camera, an RTK GNNS receiver was connected to an embedded Linux computer (Nvidia TX1), which triggers the camera when the distance since the last triggering exceeded a certain distance. In some fields, the distance was set to 10 m. In other fields, it was set to 5 m. Denser sampling increases the likelihood of covering the fields’ variation, but it also increases the amount of data and required processing time. Both images from the consumer cameras and the images from the ATV-mounted camera were taken vertically towards the ground with a ground sampling distance of 3 to 8 px mm\(^{-1}\), which ensures that weeds smaller than one centimeter can still be visually identified.

Illumination was varied from field to field and throughout a day with changing sunlight creating extremely dynamic range scenes with bright reflectance and dark shadows. The target weed classes were varied as they are being captured in situ with unknown size, scale, orientation. In total, around 26,000 training images from 200 fields in Denmark, 1149 in-situ images collected from 10 different fields in Denmark as the test set (Figure 1B), and 6625 weed free images to be used for synthetic data generation pipeline, were prepared. A sample of an image acquired using the ATV-mounted camera is shown in Figure 2.
Figure 1. (A) The recording setup used HVCAM to collect both the training and test images; (B) the in-situ images collected from 10 different fields across Denmark as test set.

Figure 2. The acquired images in situ.
3. Methodology
3.1. Synthetic Data Generation

In this work, the real field data were utilized to train the deep learning model and in order to evaluate the capability and robustness of the trained model. Two scenarios were employed, namely (1) generating synthetic images and (2) gathering and annotating in-situ images. The key point in generating synthetic images is to make the generated images resemble a real in-field environment, where the images represent natural growing weeds and crops (Figure 2). Therefore, we collected some weed free images (approved by the experts) as backgrounds and several manually cut-out monocot/dicot images obtained from the in-field images as objects. It is important to mention that all of the collected images for generating synthetic samples were not used in the training procedure and the models had not seen these images. In the next step, once objects and backgrounds were successfully gathered, an approach called cut-and-paste was implemented. The cut-and-paste method is simple, fast, and effective. The application of the proposed approach relies on first the quality of the collected objects and background images and second pasting the objects into the appropriate locations in the background image. Furthermore, this method is rapid and automatic to generate synthetic images for the tasks of object detection and classification as the network recognizes the position and label of the pasted objects within the images. Therefore, we collected approximately 8051 cut-out monocot/dicot weed images as the objects, 1059 background images containing only soil, and 1362 images including both soil and crop (weed free images). In the data generation stage, various scenarios including different illumination conditions, soil and crop types were considered to ensure the generated images cover a wide range of the in-field scenarios. The algorithm structure is presented in the following:

1. A set of transformations such as rotation, zooming in and out, and blurring methods were applied on the weed images that prepared them for the next step.
2. If the random-chosen background contains several pixels belonging to the crops, the proposed segmentation algorithm will be active and find crop pixels in the image.
3. A random number between 1 and 100 refers to the number of the needed weeds for each background image will be chosen and according to the selected number, a list of the weeds is picked out from all available weed images. The selected number represented the level of occlusion in the generated synthetic images. The larger the number, the heavier occlusion within the generated synthetic images.
4. The developed automatic image processing algorithm chooses a random x and y coordinates to place the weed image on the randomly chosen background image.
5. The overlap between crop and weed pixels is calculated and if the overlap is less than 10 percent of the weed size, the coordinate will be approved and the selected weed paste on the background, otherwise the overlap region of the weed is eliminated, and its bounding box is updated based on the remaining part of the weed which has no overlap with the crop pixels.
6. These steps are repeated until all selected weeds will be properly pasted within the background image and finally the synthetic image is generated.

Noteworthy, the segmentation algorithm has been only developed to find pixels belonging to various crops in the background images. The segmentation algorithm first transferred the RGB background images to HSI color space and then the Otsu thresholding approach was applied on the H component of the transferred images in order to extract the crop pixels from the rest of the image. By extracting the crop pixels in the background images, we can find the proper locations for placing the segmented weeds on backgrounds by following the stages defined above. A few of the generated images have been illustrated in Figure 3. As it is shown in this figure, the generated images cover different illuminations, quality levels, and visual appearances which is fundamental for assessing the network robustness.
Therefore, in this work, we used the field images to train the network and followed two scenarios to evaluate the trained model’s performance. These two scenarios were: (1) the generated synthetic images (Figure 3) and (2) the annotated in-field images with various in-field environments (Figure 4). One key point must be noted that we tried to only gather and annotate the in-field images that looked so different from the training images in terms of the visual appearance and in many cases the field environments in the images were significantly complicated due to the lack of the appropriate illumination or the heavy occlusion within the collected images.
3.2. Supervised Learning

Supervised learning occurs once the datasets for training are completely labeled. The dataset passed in the deep learning model as input contains the images along with the corresponding labels. In the supervised learning process, the network learns how to create a mapping function from a given input to an output based on the label information. The network is trained until it can extract the underlying patterns and relationships in the training samples. It is called supervised learning since the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. This approach is popular in both classification and regression tasks. Therefore, we used the supervised learning method to train our deep learning model.
3.3. Neural Network Architectures

Neural networks (NNs) are one of the main tools used in machine learning that consist of input and output layers, as well as hidden layers with units that transform the input layer to the output. The basic idea behind NNs is to simulate lots of densely interconnected brain cells inside a computer to make it possible to learn and recognize patterns and make decisions in a human-like way.

Classical NNs are commonly ordered in multiple stacked layers with the hidden layer output of one layer forming the input of the consecutive layer. By determining both the number of hidden layers and number of neurons in each layer, the overall structure of NNs is specified. Then the expressions computed at each node of NNs are compositions of other functions which are called activation functions. The degree of complexity of such compositions depends on the depth of the hidden layer and neurons within that layer. For example, a node in the second hidden layer \((l = 2)\) performs the following computation:

\[
y^{(l)}_k = f \left( \sum_j w^{(l)}_{jk} f \left( \sum_i w^{(l-1)}_{ij} x_i + b^{(l-1)}_j \right) + b^{(l)}_j \right)
\]

where \(w\) are the weights, \(b\) are the biases, \(x_i\) represents the input and finally, \(y\) is the output of the layer \(l\). Finally, the function \(f\) in Equation (1) makes it possible for NNs to learn non-linear patterns by employing Tanh, Sigmoid, or ReLU activation functions.

Generally, the goal of NNs is to optimize the loss function with respect to the parameters over a set of inputs. Therefore, all parameters in NNs such as weights and biases are optimized during training. First, the network starts with a random guess at the parameters, then figures out which direction the loss function steep downward the most, and steps slightly in that direction and then the loss value goes down. This process is repeated over and over to give the minimum error, and that state is selected as the optimum model.

However, the performance of classical NNs, in terms of efficiency and accuracy, degrades due to the complex nature of tasks such as pre-processing, segmentation, feature extraction, feature selection, etc. In computer vision problems. Convolutional neural networks (CNNs) are a specialized category of NNs which successfully deal with image-based data. CNNs take advantage of local spatial coherence in the images, which allows them to have fewer weights as some parameters are shared. This process, taking the form of convolutions, makes them especially well-suited to extract relevant information at a low computational cost.

In this work, the locations, and the labels of the weeds within the images are required to be detected. Therefore, the state-of-the-art object detection models were employed to find both the bounding box and the class for each individual weed presence within the images. However, in this race of creating the most accurate and efficient model, the researchers recently released the EfficientDet model [34]. It achieved the highest accuracy with the fewest training epochs in object detection tasks. Also, recently, the new version of the YOLOv5 model has been released and significantly noticed by the researchers due to its performances on various datasets in computer vision and machine learning and its capability to be utilized for real-time applications [35]. Hence, both EfficientDet and YOLOv5 models were utilized to detect monocot/dicot weeds in the images.

In the next parts, we describe the deep learning models and the criteria that was used to assess the accuracy of the models.

3.3.1. EfficientDet

The EfficientDet algorithm was proposed to improve the multiple dimensioned feature fusion structure of FPN and using ideas gained from the EfficientNet model scaling method for reference, EfficientDet comprises three components [34]. As shown in Figure 5, the first component is a backbone which is a pre-trained EfficientNet by ImageNet so that a pre-trained model was used for training on the weed images. The second component is BiFPN, it performs the top-down and bottom-up feature fusion multiple times for the output
specifications of levels 3–7 in EfficientNet. The third component is the classification and
detection blocks, to classify and return the monocot/dicot weeds in the images. Modules in
part two and part three can be repeated multiple times, depending on hardware conditions.
EfficientNet-b3 is used as the backbone in order to extract features from the weed images.
Then extracted features from P3-P7 passed into BiFPN block for feature fusion. BiFPN
pursues a weighted feature fusion approach to gain semantic information of different sizes.
Therefore, by following this method, the network can detect very small monocot/dicot
weeds in the images which is crucial in recognizing weeds in the agricultural fields.

Figure 5. The EfficientDet architecture [33].

3.3.2. YOLO

In 2016, a new deep learning algorithm called YOLO was proposed [35]. Before
developing the new generation of object detection models in computer vision, to find objects
within images, classification models were applied to a single image multiple times, regions,
and scales. The YOLO algorithm involves a one-time application of one deep learning to the
whole image. The model splits the image into regions and then determines the probabilities
of classes for each object and the scope of objects in images. The third version of the YOLO
algorithm was published in 2018 as YOLOv3 [36]. The YOLO algorithm is one of the fastest
object detection networks, and has already been utilized in various areas such as agriculture.
Recently, the new version of the YOLO algorithm called YOLOv5 was released [37]. YOLO
algorithm utilizes convolutional networks, and this model was chosen for its robustness
and performance in diverse domains in object and pattern recognition. In this work, the
pre-trained YOLOv5m algorithm on COCO dataset was employed, COCO dataset includes
approximately 1.5 million objects of 80 categories marked out in images [38] (Figure 6).
3.4. Evaluation Metrics

In this work, the model is assessed by popular metrics in object detection. These metrics enable us to compare multiple detection systems objectively or compare them to a benchmark. Accordingly, prominent competitions such as PASCAL VOC and MSCOCO provided predefined metrics to figure out how well the object detection models perform. Therefore, for this work, the detection task was evaluated by the precision/recall curve and confusion matrix. The principal quantitative measure used was the average precision (AP). Detections are considered true or false positives based on the area of overlap (IoU) with ground truth bounding boxes. To be considered a correct detection, the area of overlap between the predicted bounding box (A) and ground truth bounding box (B) must exceed 50% by the formula:

$$\text{IoU} = \frac{\text{Area}(A \cap B)}{\text{Area}(A \cup B)}$$ (2)

The output of the object detection model is formed in three terms including the confidence scores, the bounding boxes coordinates, and the classes. Confidence score is the probability that an anchor box contains an object. It is predicted by the classifier branch in the object detection network. In order to compute precision, recall, and confusion matrix, three parameters containing true positive (TP), false positive (FP), and false negative (FN) need to be properly defined. A detection is considered a TP only if it meets three conditions: confidence score greater than threshold; the predicted class matches the class of a ground truth; and the predicted bounding box has an IoU greater than a threshold (0.5 in this work) with the ground-truth. Violation of either of the latter two conditions make a FP.
Therefore, we now are able to calculate precision, recall, and confusion matrix. Precision is defined as the number of TP divided by the sum of TP and FP:

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{3}
\]

Recall is defined as the number of TP divided by the sum of TP and FN (note that the sum is just the number of ground-truths, so there is no need to count the number of FN):

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{4}
\]

By setting the threshold for confidence score at different levels, there are different pairs of precision and recall. With recall on the x-axis and precision on the y-axis, the precision-recall curve is appropriately drawn, which represents the association between the two metrics. Afterwards, AP obtained from the precision/recall curve was employed to evaluate the performance of both object detection networks. Therefore, in this work, the AP criteria is the base of the assessment procedure to determine which model had more TP and less FN and FP.

The average precision (AP) is obtained by calculating the area under the curve (AUC) of the precision x recall curve. The precision x recall curves are often zigzag, therefore, comparing different curves (obtained from different networks) in the same plot usually is not an easy task since the curves tend to cross each other frequently. Therefore, the AP, a numerical metric, have been used to compare different networks. In practice AP is the precision averaged across all recall values between 0 and 1. In this study, we illustrated Precision x Recall curve and calculated the AP metric to compare the performance of EfficientDet and YOLOV5 networks with together.

4. Results and Discussion

In this section the steps required for data augmentation, training the deep learning networks, and assessing the outputs of the networks to detect monocot/dicot weeds are presented in the following.

4.1. Data Preprocessing and Augmentation

Images in which weeds are annotated with bounding boxes are required for calibrating the weed detection networks. However, before feeding the raw images to the networks, the image normalization approach was applied to ensure the images were prepared in the way that standard computer vision dataset provided for training the deep learning models. In order to perform image normalization, the mean and standard deviation of training samples were computed and according to the computed values the normalization function (Equation (5)) was formed:

\[
\text{NImg} = \frac{\text{Image} - \text{Mean}}{\text{Std}} \tag{5}
\]

where, NImg is the normalized image range between 0 and 1, Image is the original RGB image, Mean and Std are the average and standard deviation obtained from all training images, respectively. Afterwards, various augmentation techniques including random contrast, blurring the images due to recording setup vibrations and changes, horizontal and vertical flips, random brightness, color variations (RGB shift), scaling, and cropping methods were carried out to enhance the quality of the training samples and subsequently to assist the model to be more generalized on the unseen data. These approaches are applied in random combinations during loading the training images for the networks. In each iteration, the data augmentation pipeline with 0.5 probability was applied to the mini-batch original images and then the augmented images were fed to the networks in learning procedure. Figure 7 illustrated the number of augmented images utilized in the learning procedure.
Figure 7. Various image augmentation techniques: (A1) original; (A2) contrast; (A3) blurring; (B1) horizontal flip; (B2) vertical flip; (B3) color to gray; (C1) brightness; (C2) RGB shift; (C3) cropping; and (D2) scaling methods.
4.2. Experimental Setup

Both models (EfficientDet and YOLOv5) were trained with the same set of hyperparameters: Adam optimizer, cross entropy loss, learning rate of $1 \times 10^{-4}$, weight decay of $1 \times 10^{-5}$, and a batch size of 12. The image resolution for both training and inference were $1600 \times 1600$ pixels, also the input dataset was split to 70% training and 30% validation set to ensure the networks were not overfitted in the training procedure.

For this study, two models were fine-tuned with around 26,000 in-field samples labeled manually by the weed experts to give accurate information to the networks at the time of learning. For training the models, a machine with two GeForce RTX 2080, 11GB GDDR6 graphic cards were used. Training metrics included precision, recall, AP, and confusion matrix. In the performance evaluation and comparison, the best model was selected by considering their confusion matrices. All codes were written in Python programming language, using OpenCV and PyTorch frameworks.

4.3. Quantitative Analysis
4.3.1. The First Test Scenario

In this study, the main objective is that the number of boxes accurately coordinate with the number of ground-truth monocot/dicot within an image. The defined metrics allow for a better insight to figure out how well the trained models can correctly recognize two weed categories and localize them in the images.

For the evaluation procedure, by following the first scenarios, 6625 synthetic test images were employed and both EfficientDet and YOLOv5 models ran on them. The experiments obtained from performing these two models on the synthetic images showed almost the same performance (Figure 8). This Figure illustrated the precision/recall curve and AP for monocot/dicot classes. The experiments represented that EfficientDet and YOLOv5 obtained the AP of 45.96% and 37.11% for detecting monocots while both models had the almost same AP at identifying dicots in the images (Figure 8). From the results given in Figures 8 and 9, it is observed that for the monocot class there are significant reductions in AP and TPR compared to dicot. A major contributor to this low performance was the high number of false positives for monocot class. A possible reason for this is the differences of visual properties between monocot and dicot classes as well as the high similarity in appearance between monocot weeds and some of the crops that caused detecting crops as monocot class within the images. Unlike monocot which only contained narrow leaves and a particular shape, dicot had more visual characteristics such as various colors and textures that made them more distinguishable and facilitated for the models to accurately detect them in the images. On the other hand, these characteristics assisted the models to correctly recognize and localize dicots in the images rather than the monocots which had almost the similar characteristics with the crops in the fields. However, both EfficientDet and YOLOv5 models could detect dicot weeds with high AP, 64.27% and 63.23%, respectively.

The normalized and non-normalized confusion matrices for both models were computed and shown in Figure 9. There are two FP groups in the confusion matrix for each weed class The first group belonged to the detections in which the IoU values were less than 0.5, and the second group were the detections in which had the IoU greater than 0.5 but with wrong predicted class. The false positive rate (FPR) of the actual monocot weeds that were wrongly detected as the dicot were only 3% and 1% for EfficientDet and YOLOv5 models, respectively; and 1% and 1% for the dicot weeds miss-classified as monocot.
were obtained 59% and 54%, respectively, which demonstrated that there were significant
weed class The first group belonged to the de
tections in which the IoU values were less
computed and shown in Figure 9. There are two FP
groups in the confusion matrix for each
mensions in which many of the monocots/dicot were
hidden among the dense crops (Figure 10). In general, the weeds contain various
sizes and orientations while in the collected dataset in this study, small to medium-sized dicot
weeds were mainly
in synthetic samples. Furthermore, the FNR in monocot for EfficientDet and
YOLOv5 were obtained 59% and 54%, respectively, which demonstrated that there were significant
undetected monocots in the predictions. The main reason for this level of error rate gained
from both models could be the difficulty of finding the monocot class with narrow leaves
in the cases where both crops and monocots appeared alongside each other in the fields.
In such a scene as illustrated in Figure 10, monocots had similar visual properties with
the crops so that the detection networks were not able to find and recognize all of the
monocots within the images. Therefore, the significant number of monocots had not been
detected (Figure 10). In general, the weeds contain various sizes and orientations while
in the collected dataset in this study, small to medium-sized dicot weeds were mainly
presented in the fields and the experiments approved that our models were able to detect
most of them within the images (Figures 10–12), the models had seen approximately two
hundred thousand thumbnails (objects) within the images. This number of weeds in the
training samples played a key role forcing the models to extract only the essential features
belonging to the weeds which help both models to be robust in their predictions. Also, the
obtained results demonstrated the efficiency and capability of region-based ConvNets on
detecting the weed dataset collected in the complex environments in which many of the
monocots/dicot were hidden among the dense crops (Figure 12). However, to see how well
both models visually performed on the similar test images, the predictions of both models
were presented in Figure 12.

Figure 8. The Precision/Recall curve obtained from the results of the object detection models on the
generated synthetic images. **Left**: EfficientDet, and **Right**: YOLOv5. In this Figure, PPPMM and
PPPDD are monocot/dicot classes, respectively.

Accordingly, both models had only a few cases that the monocot and dicot misclassified
from each other. In addition, based on the obtained results from the confusion matrix, the
misclassification rates at detecting monocot/dicot were insignificant and this led us to
conclude, both EfficientDet and YOLOv5 were robust enough in detecting monocot/dicots
in synthetic samples. Furthermore, the FNR in monocot for EfficientDet and YOLOv5
were obtained 59% and 54%, respectively, which demonstrated that there were significant
undetected monocots in the predictions. The main reason for this level of error rate gained
from both models could be the difficulty of finding the monocot class with narrow leaves
in the cases where both crops and monocots appeared alongside each other in the fields.
In such a scene as illustrated in Figure 10, monocots had similar visual properties with
the crops so that the detection networks were not able to find and recognize all of the
monocots within the images. Therefore, the significant number of monocots had not been
detected (Figure 10). In general, the weeds contain various sizes and orientations while
in the collected dataset in this study, small to medium-sized dicot weeds were mainly
presented in the fields and the experiments approved that our models were able to detect
most of them within the images (Figures 10–12), the models had seen approximately two
hundred thousand thumbnails (objects) within the images. This number of weeds in the
training samples played a key role forcing the models to extract only the essential features
belonging to the weeds which help both models to be robust in their predictions. Also, the
obtained results demonstrated the efficiency and capability of region-based ConvNets on
detecting the weed dataset collected in the complex environments in which many of the
monocots/dicot were hidden among the dense crops (Figure 12). However, to see how well
both models visually performed on the similar test images, the predictions of both models
were presented in Figure 12.
Figure 9. The normalized and non-normalized confusion matrices obtained from EfficientDet (Left pictures) and YOLOv5 (Right pictures) on the generated synthetic images.
Figure 10. The EfficientDet results in a random synthetic image. **Top**: the label, and **Bottom**: the prediction. In this Figure, red and blue continuous rectangles represent TP monocot and TP dicot respectively; pink dash rectangles show FN of M/D and yellow dash rectangle is FP of M/D.
Figure 11. The YOLOv5 results in a random synthetic image. Top: the label, and Bottom: the prediction. The other information is the same as Figure 10.
4.3.2. The Second Test Scenario

In the second scenario, as mentioned earlier in the materials and methods, 1149 real field test images were picked from 10 different fields from Autumn 2017, Spring 2018, and Spring 2021. The test images were picked in a manner ensuring the variation in monocot and dicot weed densities were spanned within the separate field. The test images were annotated into monocot/dicot classes by the experts to realize how well the trained models performed on the in-field samples. The new test images were gathered and labeled from the totally new fields to ensure the trained models had not seen them before in the calibration samples and therefore, we will gain the real performance from both models.

Figure 12. The monocot/dicot prediction by EfficientDet and YOLOv5 in synthetic images: (A1) The label in the dark soil image; (A2) The EfficientDet prediction; (A3) The YOLOv5 prediction; (B1) The label in the light soil image; (B2) The EfficientDet prediction; (B3) The YOLOv5 prediction; (C1) The label in the dense image; (C2) The EfficientDet prediction; (C3) The YOLOv5 prediction. In this Figure, various crops and weeds occlusions are illustrated; red and blue bounding boxes represent monocot and dicot classes, respectively.
For the evaluation procedure, by following the second scenario, 1149 in-field test images were utilized and both models ran on them. We followed the exact same procedure as discussed in the first scenario for assessing the model robustness on the in-situ image, too. The experiments obtained from performing these two models on the synthetic images showed almost the same performance (Figure 13). This Figure illustrated the precision/recall curve and AP for monocot/dicot classes on the in-situ images. The experiments showed that EfficientDet and YOLOv5 gained the AP of 27.43% and 30.70% for detecting monocots and 42.91% and 51.50% for dicots. These results approve that YOLOv5 performance overcome EfficientDet in situ environments. The quality of some collected images in situ were not as the same as most of the training sample, and we tried to gather complicated images in terms of the visual image characteristics such as heavy occlusion to challenge both models to see their real performance and their generalizations in real field environments. This could be a rational reason for the decrease of both models in detecting monocot/dicot within the images. However, by comparing the AP value of synthetic and in-situ images, it is found out YOLOv5 model could preserve its performance on in-situ images and overcome EfficientDet.

In Figure 14, The normalized and non-normalized confusion matrices of the weed dataset were computed. It can be seen that, the false positive rate (FPR) of the actual monocot weeds that were wrongly detected as dicot were 3% and 1% for EfficientDet and YOLOv5 models, respectively; and 1% and 1% for the dicot weeds miss-classified as monocot. This is due to the fact that the extracted features from dicot are robust and make both networks capable of distinguishing dicot from monocot clearly. In terms of the FNR and FPR for both classes, YOLOv5 achieved better performance as it had a smaller number of false predictions in situ dataset. However, both models could not find several numbers of the monocot within the images. The possible reasons for false negative recognition of monocot are the various soil backgrounds and similar leaf shapes and properties between monocot and crops in the field environments. Providing more labeled data and variation in the training set could be helpful to enhance the recognition performance.
Figure 14. The normalized and non-normalized confusion matrices obtained from EfficientDet (Left) and YOLOv5 (Right) on the in-situ images.

However, the model estimation for both EfficientDet and YOLOv5 are presented in Figure 15. In this Figure, various complex and dense field environments have been investigated to assess the robustness of both models in real-field environments.
Figure 15. The monocot/dicot prediction by EfficientDet and YOLOv5 on the in-situ images: (A1) The label in the light soil image; (A2) The EfficientDet prediction; (A3) The YOLOv5 prediction; (B1) The label in the dark soil image; (B2) The EfficientDet prediction; (B3) The YOLOv5 prediction; (C1) The label in the dense image; (C2) The EfficientDet prediction; (C3) The YOLOv5 prediction. In this Figure, red and blue bounding boxes represent monocot and dicot classes respectively.

The performance and real-time of the detection algorithm determine whether it could be applied in practical agricultural production such robotic applications. The inference running time below showed YOLOv5 was faster than EfficientDet (Table 1). The main reason for this is the large computational complexity in EfficientDet in comparison to YOLOv5 architecture. Therefore, YOLOv5 is an appropriate choice if the algorithm will be utilized in real-time weed detection applications.
5. Conclusions
Weed detection is crucial in agricultural productivity, as weeds act as a pest to crops. In this study, we aimed to propose an image processing algorithm for generating weed synthetic images and two deep-learning methods to detect and recognize monocot/dicot weeds in RGB images collected in situ environments. Both EfficientDet and YOLOv5 methods could detect and recognize dicot with high performance in two scenarios (the synthetic and in-situ images). The AP of monocot/dicot on 6625 weed synthetic images for EfficientDet and YOLOv5 models were 45.96% / 64.27% and 37.11% / 63.23%, respectively. These results approved the capability of both models in detecting weeds within the weed synthetic images. However, it is crucial to determine the performance of proposed models in situ environments. Based on the obtained results by both models, YOLOv5 could preserve its performance and robustness in the real-field environments. The AP of monocot/dicot in YOLOv5 model were 30.70% / 51.50% and in comparison, with EfficientDet, YOLOv5 had better performance in detecting both monocot and dicot on the in-situ images. These results indicated that it is reliable to utilize the YOLOv5 model in the field environments as its performance and its capability had not significantly decreased in the same way as EfficientDet. Therefore, this study shows that the deep-learning YOLOv5 architecture provides an efficient approach to detect and recognize weed in RGB images captured outdoor conditions.

Future work will be carried out to assess the performance of various state-of-the-art object detection networks on different image resolutions in weed data set. Also, it would also be interesting to train and then assess the performance of detection networks on detecting and recognizing various crop types such as potato, sugar beet, maize, peas, and other agricultural crops.

Author Contributions: N.T. was responsible for analyzing the data. N.T. implemented the deep learning approaches. R.N.J. annotated the raw images. O.G. generated the initial weed detection idea. All authors took part in writing the paper. All authors have read and agreed to the published version of the manuscript.

Funding: The work was founded by the Innovation Fund Denmark via the RoboWeedMaPS project (J.nr.6150-00027B); by Green Development and Demonstration Programme under the Ministry of Environment and Food Denmark (GUDP) via the SqMFarm project (Journal no.: 34009-17-1303); and by Innovation Fund Denmark via the Future Cropping project (File no.: 156-2014-10).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The publicly available dataset named RoboWeedMap can be found in https://weed-ai.sydney.edu.au/datasets/aa0cb351-9b5a-400f-bb2e-ed02b2da3699, accessed on 29 March 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References


29. Ubbens, J.; Cieslak, M.; Prusinkiewicz, P.; Stavness, I. The Use of Plant Models in Deep Learning: An Application to Leaf Counting in Rosette Plants. *Plant Methods* 2018, 14, 6. [CrossRef]


