Crop and Weed Detection Using Region Based Convolutional Neural Networks

Lanlan Gao
lanlan.gao@studenti.unipd.it

Md Rubayet Afsan
mdrubayet.afsan@studenti.unipd.it

Omer Faruk Caki
omerfaruk.caki@studenti.unipd.it

Abstract

In this project, we used Region Based Convolutional Neural Networks (R-CNN) to classify plants and weeds. The ability to distinguish between crops and weeds and to detect them accurately is crucial for many different applications, especially in precision agriculture and automated weeding systems.

1. Introduction

In this study, we address the plant detection problem, one of the major challenges in the agricultural sector. With the increasing population and the need for efficiency in sustainable farming practices, fast, accurate, and automatic detection of plants is essential. Existing systems often struggle when trying to differentiate plants with similar colors and textures, especially under varying lighting conditions or complex backgrounds, which can negatively impact detection accuracy.

The main goal of this work is to overcome the challenges encountered in plant detection by developing a model based on region-based convolutional neural network (R-CNN) architecture. Our approach aims to provide a more robust, and flexible method. This is achieved through enhancements in data preprocessing, model architecture, and optimization techniques.

A key motivation for this study is to successfully perform the classification of crops and weeds, which is the most important first step in laser weeding applications. In agriculture, weeds compete with crops for soil minerals and nutrients, reducing both yield and quality [1]. A study conducted in 1992 demonstrated that lasers could be used effectively for weed control [2]. This method is not only environmentally friendly but also cost-effective, as it reduces the use of chemicals and lowers labor costs.

2. Related Work

In recent years, many studies have focused on plant detection, and research in this area continues to grow. The increasing adoption of advanced agricultural applications, which greatly benefit from technological advancements, is the main reason behind this trend. Most research in this field concentrates on detecting plant diseases and/or identifying weeds. In our study, we focused specifically on distinguishing between crops and weeds. Effective weed control is essential for increasing agricultural productivity, and the first step is to correctly separate crops from weeds. Rapid developments in deep learning have made this task more manageable [3].

Heisel et al. (2001) conducted a study using CO_2 lasers to cut the stems of *Chenopodium album* and *Sinapis arvensis* and the leaves of *Lolium perenne*. The results showed that cutting the stems below the meristem with lasers significantly reduced the biomass and provided effective weed control [4]. However, this study did not incorporate any deep learning or image processing techniques.

Zhu et al. (2022) developed a robot equipped with the YOLOX deep learning algorithm and a blue laser to detect and eliminate weeds in cornfields. The robot comprises a tracked mobile platform, a weed identification module, and a five-degree-of-freedom robotic arm with a laser emitter. The system differentiates corn seedlings from weeds by analyzing their texture and shape, calculates the weed coordinates using a monocular distance sensor, and then directs the laser to eliminate the weed. While moving at a speed of 0.2 m/s on a flat surface, the detection rates for corn seedlings and weeds were recorded as 92.45% and 88.94%, respectively. Additionally, the system reduced the dry weight of the weeds by 85% and caused only 4.68% damage to the seedlings. These results demonstrate that the robot can accurately detect and effectively eliminate weeds in cornfields [5].

Hasan et al. (2022) evaluated the performance of five deep learning models—VGG16, ResNet-50, Inception-V3, Inception-ResNet-v2, and MobileNetV2—for recognizing weeds in agricultural images. To address class imbalance, data augmentation techniques were applied, and a larger crop-weed dataset was created by combining several smaller datasets. Transfer learning methods were used to fine-tune the pre-trained models on crop and weed images. The results indicated that VGG16 performed best on small-scale datasets, while ResNet-50 achieved superior performance on larger, balanced datasets. This study shows that data augmentation and fine-tuning significantly enhance the performance of deep learning models for classifying crop and weed images, and it highlights the need for a large-scale standard weed dataset [6].

Yan et al. (2024) introduced a lightweight model named DETR-RPC-CDF for weed detection in agricultural fields. Tested primarily on the Fine24 dataset, the model uses a restructured partial convolution (RPC) and a collection-dispersion feature fusion (CDF) mechanism. The results show a 2% increase in mAP@0.5, along with a 40% reduction in parameters, a 43% reduction in FLOPs, and a 40% increase in detection speed. These findings underscore the potential of the DETR-RPC-CDF model for real-time, lightweight, and accurate weed detection in agricultural applications [7].

3. Dataset

For this project, we required a high-quality, well-structured dataset with comprehensive annotations for the semantic interpretation of agricultural images. To meet these requirements, we chose the PhenoBench dataset [8]. PhenoBench was collected by a research team from the University of Bonn in Germany, on agricultural fields located at Campus Klein-Altendorf between Meckenheim and Rheinbach. Data collection took place on various dates in 2020 and 2021. In 2020, data was gathered on May 15, May 26, and June 6; in 2021, data was collected on May 20, May 28, June 1, and June 10. These dates cover different growth stages and varying lighting conditions.

Data was captured using a DJI M600 unmanned aerial vehicle (UAV) equipped with a high-resolution PhaseOne iXM-100 camera mounted on a gimbal with an RSM prime lens. This configuration provided motion stabilization and allowed for the capture of RGB images with approximately 100-megapixel resolution. The UAV flew at an altitude of about 3 meters, achieving a ground sampling distance (GSD) of 0.03 mm per pixel. Flight planning was managed using the DJI Ground Station Pro application, ensuring a 90% forward overlap between consecutive images and a 70% side overlap between adjacent rows. Each image was georeferenced using the onboard GNSS system.

The PhenoBench dataset is notable for its detailed, multi-layer annotations. It includes pixel-level semantic labels for over 5,000 plants and leaf-level annotations for more than 30,000 beet leaves. These annotations serve as a rich resource for various detection tasks, such as segmentation and object detection at both the plant and leaf levels. The annotation process was performed manually by experts who carefully labeled each plant and leaf. With images taken under various growth stages and lighting conditions, the dataset ensures that our models perform reliably and robustly in different environmental settings.

The high resolution and extensive size of the dataset enable our models to learn fine details and make more precise predictions. Using UAVs for image collection allowed for rapid and efficient scanning of large areas, thereby increasing the dataset's diversity and scope. The comprehensive, high-quality data provided by PhenoBench plays a critical role in achieving the objectives of our project.

4. Method

In this study, we use a simple Faster R-CNN based approach to detect plants. Our method has three main parts: the dataset and preprocessing, the model structure, and the training and evaluation process. Next, we explain each part in detail.

4.1. Data Set ve Preprocessing

The dataset used in this project is PhenoBench, which is mentioned in the previous section.

4.1.1 Data Collection & Annotation

PhenoBench dataset come with various annotations, in this project have used plant instances masks, and these are used to compute bounding boxes with given class labels. Class 1, and 2 are used for plant and weed instances, respectively. Figure 1 shows and example of training data. In the first row its the RGB image, in the second row the instance masks where green, and red colors are used for crops, and weeds respectively. Lastly, in the third column the calculated bounding boxes are drawn.

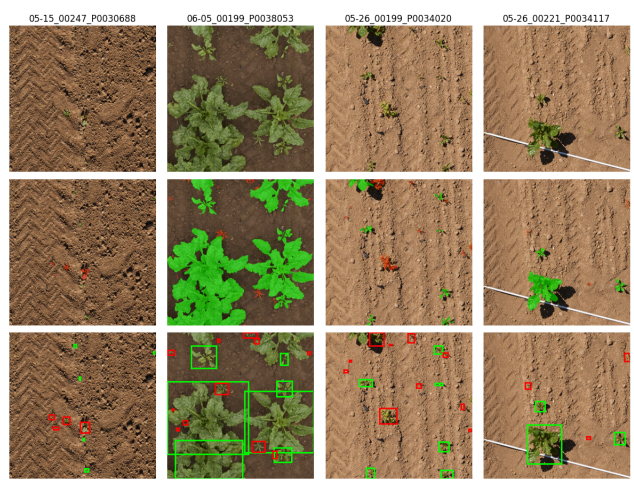

Figure 1. Example of training data.

4.1.2 Image Preprocessing

Phenobench dataset was in ready to use condition, which does not required any preprocessing or normalization.

However, in order to enhance and optimize the dataset, and reduce the training time we have applied:

- **Limiting:** PhenoBench dataset had more than 1400 training image originally, and we have added 200 augmented new image. In order to reduce training time and computational burden, we have reduced the dataset into 1000 images by randomly selecting among them.

- **Data Augmentation:** In order to reduce the overfitting risk, and enhance the dataset we have applied some data augmentation methods as proposed by [9]. These methods include color augmentation, rotating, and flipping. Example of augmentation result is shown in Figure 2. In the figure we see that brightness, contrast, and saturation color augmentations, and rotating and flipping in order.

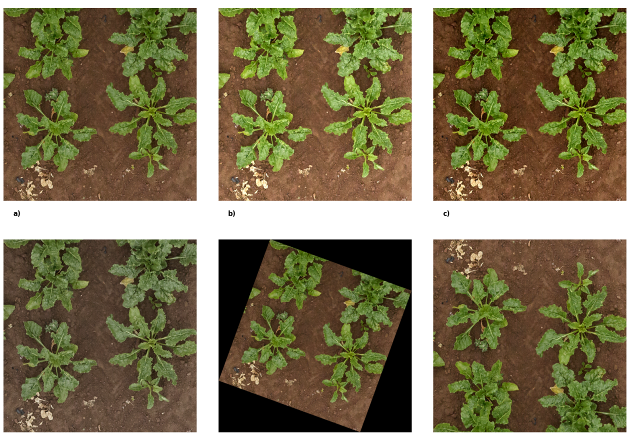

Figure 2. Data augmentation example.

4.2. Model Architecture

We aim to accurately detect multiple plant species within diverse environmental settings. Faster R-CNN is well-suited to this task, as it combines efficient region proposal generation with robust multi-scale feature extraction. Compared to one-stage detectors (e.g., YOLO, RetinaNet), our chosen approach often yields higher accuracy for objects with varying sizes and occlusions which a common scenario in plant imagery.

The model architecture is implemented in `[models/fasterrcnn.py]` as a PyTorch neural network module. The implementation utilizes torchvision's Faster R-CNN with ResNet50-FPN backbone, configured according to specifications in `[configs/fasterrcnn_plants.yaml]`.

Key components of the architecture include:

- **Base Network:** We use fasterrcnn_resnet50_fpn, which combines:
  - *ResNet50 Backbone:* Provides deep feature extraction through residual layers.
  - *Feature Pyramid Network (FPN):* Facilitates multi-scale feature representation, helping detect objects of varying sizes.

  The network is initialized without pre-trained weights (`pretrained=None`), allowing it to learn features specific to our plant dataset from scratch.

- **Classification Setup:** The network is configured for 3-class classification which are crop, weed, and one class is reserved for background.

- **Training Parameters:**
  - **Learning Rate (0.0001):** Determines how quickly the model updates its parameters during backpropagation.
  - **Batch Size (24):** Number of training images processed per forward/backward pass.
  - **Maximum Epochs (100):** Upper limit on how many full passes (epochs) we make over the dataset.
  - **NMS Threshold (0.5):** IoU threshold for Non-Maximum Suppression, which removes overlapping boxes to reduce duplicate detections.
  - **Probability Threshold (0.6):** Confidence score below which detections are discarded, helping to filter out low-confidence predictions.

- **Loss Function:** During training, the built-in Faster R-CNN model returns a dictionary containing four loss terms, which are then summed into a single scalar loss for backpropagation:
  - **Classification Loss:** Measures how accurately the model assigns class labels to each detected region. If a region is classified incorrectly (e.g., labeled as the wrong plant species), this component increases the overall loss to encourage better class predictions.
  - **Bounding Box Regression Loss:** Encourages the model to output precise bounding box coordinates. It penalizes large deviations of the predicted box corners from the ground truth (i.e., correct position and size of the bounding box).
  - **Objectness (RPN) Loss:** Evaluates how well the Region Proposal Network (RPN) distinguishes between object-containing regions and background. This helps the model filter out irrelevant areas and concentrate on regions likely to contain a plant.

– **RPN Box Regression Loss:** Similar in spirit to the bounding box regression loss, but applied specifically within the RPN. It refines the initial box proposals before they are passed to the final detection head, improving the overall detection performance.

By summing these four terms, the total loss provides a comprehensive training signal that helps the model learn to (1) classify objects correctly, (2) localize them accurately, and (3) generate high-quality region proposals in the first place.

During both training and inference, the model processes input images (and corresponding target annotations) on a CUDA-enabled GPU. For performance evaluation, we monitor the Mean Average Precision (mAP) using the `MeanAveragePrecision` class from `torchmetrics`. The mAP metric provides a robust measure of the model's accuracy across all plant classes, taking into account both the precision and recall at various confidence thresholds.

5. Experiments

This section presents the experiments we conducted to evaluate our model's performance. We compare our results with previous studies, perform an ablation study to understand the impact of different components, and test various hyperparameter settings.

5.1. Hardware Setup

For training our model, we used a rented server with an *RTX 6000 Ada* GPU, which has 81.4 TFLOPS of computational power and 45 GB of VRAM. The system was equipped with an *AMD EPYC 7443* processor with 96 CPU cores. The model was trained using the *PhenoBench* dataset with the hyperparameters decripted in 4.2.

In terms of software, we utilized a containerized environment based on *Microsoft Anaconda Development Containers* [1]. This setup provided a stable and reproducible environment for managing dependencies, including Python libraries such as TensorFlow, PyTorch, and NumPy. The use of Docker ensured consistency across different machines and simplified dependency management.

Performance was evaluated using *Mean Average Precision (mAP)* and other standard classification metrics.

[1] https://hub.docker.com/r/microsoft/devcontainers-anaconda

5.2. Results

The overall performance of our model is summarized in Table 5.2. Our model achieved a **mAP@0.5** of 0.6716, with a macro F1-score of 0.6924 and a micro F1-score of 0.7081. The detailed per-class metrics are given in Table 5.2.

Metric	**Score**
mAP@0.5	0.6716
Macro-Precision	0.5962
Macro-Recall	0.8289
Macro-F1	0.6924
Micro-Precision	0.6064
Micro-Recall	0.8509
Micro-F1	0.7081

Table 1. Overall performance metrics of the model.

Class	**AP@0.5**	**Precision**	**Recall**	**F1 Score**
Crop (class 1)	0.8271	0.6356	0.9515	0.7621
Weed (class 2)	0.5161	0.5568	0.7063	0.6227

Table 2. Per-class performance metrics.

This results showed the our model is better at detecting crops, but there is a room for improvement regarding weed detection.

Figure 3. Model detection results a) Ground truth, b) Detected

An example model detection result is shown in Figure 3. Our model is successfully captured almost all crop instances, except the one on the bottom which partially visible. However, our model did miss some very small weed instances.

6. Conclusion

In this project, we built a Faster R-CNN model to detect crops and weeds in agricultural images. Our experiments showed that the model achieved a mAP@0.5 of 0.6716 and good F1-scores for both classes. The results demonstrate that our approach can reliably distinguish between crops and weeds, which is an important step toward automated weeding systems.

We learned that using a region-based convolutional neural network can effectively handle the challenges of detecting small and overlapping plant regions. The model worked well in general. However, the detection of weeds was slightly less accurate compared to crops, indicating there is room for improvement.

For future work, there are several possible extensions. First, we could explore other deep learning models such as one-stage detectors (e.g., YOLO or RetinaNet) to compare performance and speed. Second, using a larger and more diverse dataset may help improve the model's accuracy, especially for weed detection. Also, an important extension on top of this project would be and Apical Meristematic Tissue (AMT) detection of weeds, as mentioned by [10].

References

[1] D. Patel and B. Kumbhar. Weed and its management: A major threat to crop economy. Journal of Pharmaceutical Sciences and Bioscience Research, 6:453–758, 2016.

[2] A. Bayramian, P. Fay, and W. Dyer. Weed control using carbon dioxide lasers. Proceedings of the Western Society of Weed Science, Salt Lake City, UT, USA, 10–12 March 1992.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521:436–444, 2015. DOI: 10.1038/nature14539.

[4] T. Heisel, J. Schou, S. Christensen, and C. Andreasen. Cutting weeds with a co_2 laser. Weed Research, 41(1):19–29, 2001. DOI: 10.1046/j.1365-3180.2001.00214.x.

[5] H. Zhu, Y. Zhang, D. Mu, L. Bai, H. Zhuang, and H. Li. Yolox-based blue laser weeding robot in corn field. Frontiers in Plant Science, 13:1017803, 2022. DOI: 10.3389/fpls.2022.1017803.

[6] Md. A. Hasan et al. Weed recognition using deep learning techniques on class-imbalanced imagery. Preprint or online publication, 2022. (Further publication details should be confirmed.).

[7] X. Yan, Y. Li, and Z. Chen. Lightweight weed detection using re-parameterized partial convolution and collection-distribution feature fusion. Computers and Electronics in Agriculture, 200:106–114, 2024. DOI: 10.1016/j.compag.2023.106114.

[8] Phenobench dataset. Available at `https://phenobench.github.io/`. Accessed: 15 February 2024.

[9] M. Everingham, S. A. Eslami, L. van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective, 2015. Intl. Journal of Computer Vision (IJCV), vol. 111, no. 1, pp. 98–136, 2015.

[10] Daode Zhang, Rui Lu, Zhe Guo, Zhiyong Yang, Siqi Wang, and Xinyu Hu. Algorithm for locating apical meristematic tissue of weeds based on yolo instance segmentation. Proceedings of the International Conference on Image Processing, 2022.