WeedWise – Smart weed detection and classification for precision agriculture

Kailen Todd McCauley, Neha Nivedita Balaji, Cassandra Riquelme
CS 4365 Spring 25

Introduction

As we step into a modern age, we acknowledge that the farmers of today are expected to make decisions on their crops through traditional knowledge and known methodology. They lack the tools to effectively make data-driven decisions that would optimize their crop yields. However, recent advancements in AI revolving around agriculture have opened new doors for optimization and automation in this field.

Our project, WeedWise, aims to develop a system that not only is capable of identifying weeds but also classifying their growth stage and estimating their coverage within a field. Unlike most object detection models, our two-stage image classification pipelines integrate bounding-box-based weed detection with growth stage classification, providing a level of granularity underrepresented in existing research. Through precise weed monitoring, WeedWise enhances precision agriculture, aiding farmers make informed decisions to improve overall crop health and yield.

Motivation

Precision Weed Management

The overarching goal of this project is to supply farmers with a tool that provides a newfound layer of detail regarding weeds amongst their crops, rather than a binary weed/non-weed decision. Knowing the growth stage of the weed can help optimize both the timing and intervention method. For example, these insights might aid a farmer in deciding whether to apply herbicides or deploy mechanical weeding, reducing the waste and environmental impact.

Taking it a Step Further: Bridging the Gap in Current Research

As it stands, current research into the application of Computer Vision in weed detection focuses primarily on either bounding box detection (object localization) or on whole-image classification. While some Computer Vision research extends into classifying the type of weed, none seems to focus on the growth stage, despite the growth stage having an impact on herbicide uptake. Thus, in the two-fold pipeline (see proposed methods sections) we propose, there is an additional level of granularity, precision, and overall novelty. This approach is highly relevant for agriculture, a field where timing and accurate weed characterization can impact yield and sustainability.

Related Work

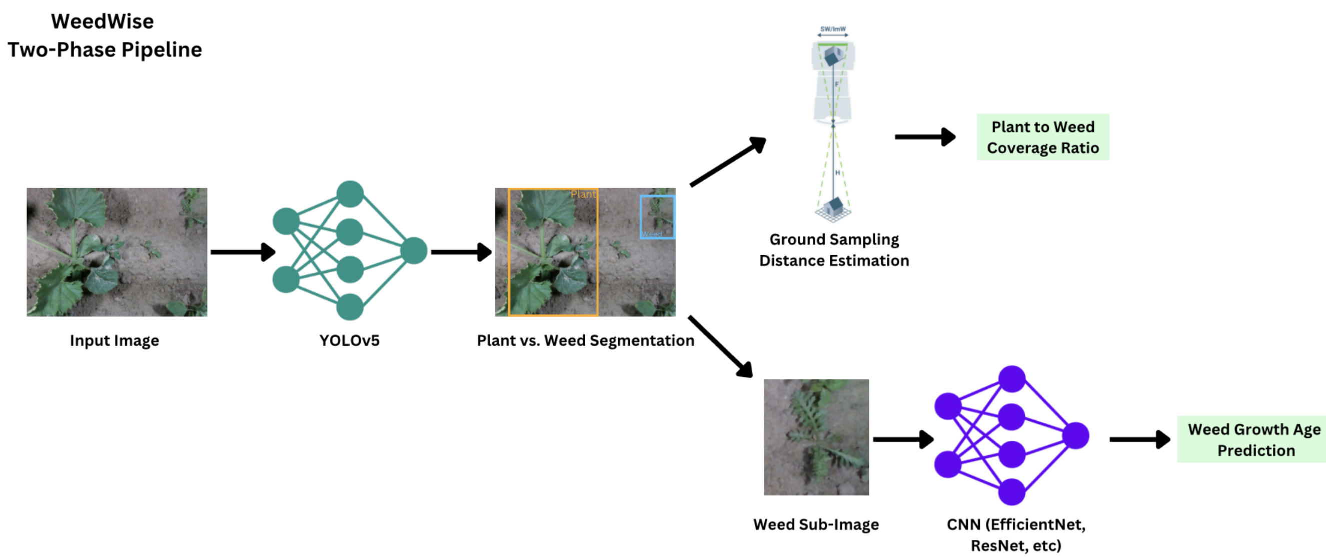
Recent advances in machine learning have spawned various approaches to weed detection and classification. A notable study in China by Tao and Wei (2022) demonstrated the efficacy of hybrid models in winter rape fields, combining CNN-SVM architectures to achieve a 92.1% accuracy on a dataset of 1500 images distinguishing between rape seedlings and four weed species. While effective for their specific agricultural context, their approach focused solely on the binary classification without addressing growth stages.

Particularly relevant to our approach is Jin et. al's (2022) comprehensive comparison of deep learning architectures for vegetable weed detection. Their work in Chinese bok choy fields demonstrated YOLO-v3's superior performance (97.1% precision) compared to other architectures like CenterNet and Faster R-CNN, validating our choice of the YOLO architecture family for our detection pipeline. However, like other existing approaches, their system treated weed detection as a binary classification problem. Research from Australian chili farms by Islam et. al. (2021) explored scalability through UAV-based detection, achieving 96% accuracy with Random Forest algorithms on aerial imagery. Their work validated machine learning approaches for large-scale agricultural applications, though remained limited to basic presence detection. In Kazakhstan, Urmashev et al. (2021) advanced the field by implementing YOLOv5 on a dataset of over 1,000 images per weed class, achieving 82-92% accuracy across species while outperforming any traditional classifiers like K-Nearest Neighbors (83.3%) and Random Forest (87.5%).

While these approaches demonstrate the viability of weed detection, they don't address the critical timing requirements of treatment selection. Controlled studies have led to agricultural resources like the University of California's Integrated Pest Management Guidelines, which show that herbicide effectiveness varies significantly with plant development stages (Wilen et. al . 2023). Growth stage detection enables farmers to choose between alternative control methods – mechanical tillage for early stages, targeted herbicide application at optimal growth stages, or manual removal when most efficient. Teimouri et al.'s work in Danish agriculture makes progress in this direction, achieving 70% accuracy (validated through cross-validation of 2516 test images and predictions from 20 trained models) in growth stage classification across 18 weed species.

Our approach aims to build upon these related works by implementing a two-stage pipeline that not only detects weeds using YOLOv5 but also classifies their growth stages – a critical dimension for precision agriculture that remains unaddressed in current literature on the use of machine learning in weed detection. Future work may enhance this approach by incorporating environmental factors and expanding growth stage classification across different weed species.

Design & Implementation



Datasets

We have found two key dataset to work with for Plant vs. Weed Image Segmentation:

1. Sesame crop and different weeds with YOLO labels
2. Multiple Crop and Weed Types with YOLO labels

These datasets provide a variety of features to make the workflow manageable. The primary being already YOLO labeled, cutting down a major pre-processing step of manually labelling the data with bounding boxes. In addition, they provide a wide diversity of crops, for a total of seven crops and eight weed varieties. Moreover, the images are from both greenhouses and open field conditions, providing added diversity to make our model more generalizable. We can assume there will be no image repeats as both data sets focus on different crop types, thus, we do not have to account for data leakage between training, validation, and testing sets.

There is also a separate dataset focused solely on weeds, with their bounding boxes provided. It has different types of weeds commonly found in agricultural and natural environments. This set can be a training set for the CNN in the second stage of the pipeline.

Preprocessing

For the object segmentation task, our dataset is already labeled and structured to be compliant with Pytorch's YOLOv5 implementation, so now preprocessing needs to be done besides aggregating the dataset together.

For our multi-class classification task, we will have to generate a weed only dataset with labels relating to growth age. The previously mentioned dataset provides the bounding boxes of weeds within the images, thus, we can utilize this information to crop images, producing the weed-only dataset.

In terms of labeling, we will have to experiment with a variety of approaches to categorize. A naive approach may be manually labelling images for their growth age based on research regarding a given weed's lifecycle. A more sophisticated approach may be to run the images through a pre-trained CNN, and extract the weed feature embeddings, then perform clustering using K-means to produce our growth stage labels. We will experiment and decide which approach balances time efficiency, feasibility, and accuracy in labeling, as overall we must ensure we are labeling based on growth stage.

YOLOv5 Training for Segmentation

We will need to train a model to be highly accurate at differentiating various plant types from various weed types. Based on the dataset we found, in conjunction with the high accuracy results from previously cited research, we intend to utilize Pytorch's implementation YOLOv5. This in comparison to other models, it is relatively light, allowing for the usage of Cloud GPU services like Google Colab.

Weeb Sub-Image dataset

Passing an image through YOLOv5 will produce bounding boxes, classification of plant or weed type per box, and confidence level of said classifications. We will use this confidence threshold, along with the bounding box to separate out the images of weeds that were detected. This processing step is vital for the second stage of our pipeline.

CNN Training for Multi-Class Classification and Supervised Learning

We will utilize the generated dataset from the preprocessing step to train this CNN. We plan to experiment with a variety of models, such as ResNet, VGG, EfficientNet, and select the model that performs the best on our data. We will primarily focus on metrics like f1-score and accuracy to evaluate model performance.

Once this model has been trained, we will need to integrate to connect with the first stage of the pipeline, taking in the weed sub-images, and then producing a growth stage prediction.

Ground Sampling Distance (GSD) Estimation

By leveraging AI-based object recognition, we can dynamically compute GSD using detected objects with known dimensions, which in this case will be our detected plans and weeds output from YOLO. Identifying these reference objects within images allows us to derive a precise pixel-to-ground ratio even when metadata for the camera is unavailable.

Foreseen Challenges

Image Format & Labeling Compatibility

Our project will be using various image formats including PNG, JPG, JPEG, and HEIC. This may introduce inconsistencies as each file type would differ in compression, metadata handling, color profiles, and encoding. Our model may encounter compatibility issues as it attempts to load, preprocess, and aggregate the images that would affect its performance. While YOLOv5 does not handle image conversions, it utilizes OpenCV and PIL libraries, both which can be used to perform these conversions to promote a consistent, suitable file type such as JPG or JPEG. However, it is integral in our pipeline that our preprocessing stage automates format conversion to YOLOv5 inputs when necessary to support image format compatibility with our model. As such, we would automate format detection and subsequent conversion to foster smooth training.

Likewise, our datasets will include different labeling formats including XML, RF, and TXT, all with varying structures and possibly lacking adequate annotations. They are expected to adhere to the format of YOLO labeling. Raw image datasets, specifically, must undergo bounding box production to maintain YOLO compatibility. It is a method of object detection to aid in our identification of specific weeds within the visual data. Combining all these datasets demands a labeling pipeline in our preprocessing stage to guarantee consistent annotations in TXT format. It will correct annotation formats through our created scripts, with manual validation where necessary. Moreover, datasets with correct labels should be preserved when we perform our data augmentation methods. These are necessary to improve the training process of our MLM and enhance its ability to generalize unseen data.

Skewed Bias

A key concern, as we begin to train our model, will be its inherent bias based on the data we use. If the distribution of our dataset only accounts for weed growth in the later part of their lifespan or only analyzes a select amount of weed species, it could influence the model's performance. We will prevent such bias by prioritizing a dataset that contains weeds found in differing geographical regions and varying stages of growth. Regular assessments will be performed on class distributions to identify any class imbalances. If such an imbalance should exist, we will apply balancing techniques (e.g., oversampling, augmentation) to deter a bias. A similar strategy is outlined in our Learned Skills—Checking Assumptions: Data Validity & Model Performance section.

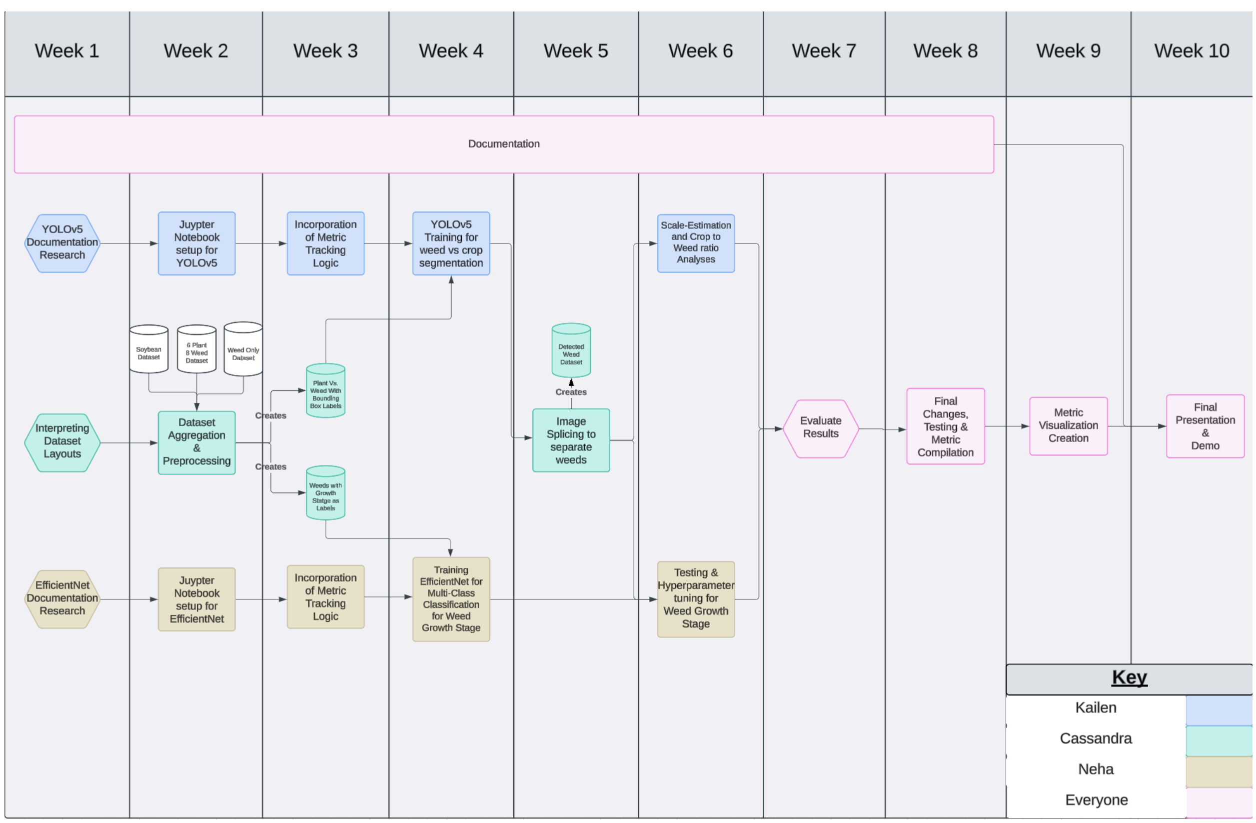
YOLOv5 integration into CNN Classification

A vivid concern of our project is the transition of our YOLOv5 weed sub-image dataset into the CNN classifier for analysis on weed growth. Any number of issues could arise in the transfer. Errors in resizing, preprocessing, or bounding box extraction could be detrimental to classification accuracy. A meticulous preprocessing pipeline will ensure seamless integration across models (e.g., ResNet, VGG, EfficientNet). Additional testing on the retention of quality in our weed images for relevant image data will protect accurate classification of our model.

Computational Limitations

Our deep learning models, especially YOLOv5 and EfficientNet, tend to have large memories and will use a substantial amount of GPU resources. Training our model with high-resolution images could prove challenging with our local computers that contain mediocre computing power in comparison. Furthermore, storing and re-running the model requires a large amount of storage on our machines. Mitigating this limitation calls for the use of cloud-based training solutions as outlined in our Learned Skills—Learning New Technologies: Machine Learning Models & Cloud Computing Platforms section.

Work Schedule



Task Allocation

Task	Team Member
YOLOv5 Jupyter Notebook Setup	Kailen
Dataset Aggregation and Preprocessing	Neha
EfficientNet Jupyter Notebook Setup	Cassandra
YOLOv5 Metric Tracking	Kailen
EfficientNet Metric Tracking	Neha
Image Splicing to Separate Detected Weeds	Cassandra
Scale-Estimation and Crop Weed Ratio Analysis	Kailen
Testing and Hyperparameter tuning for Weed Growth Stage	Neha
Results Evaluation	Kailen, Neha, Cassandra
Final Changes and Metric Compilation	Kailen, Neha, Cassandra
Visualizations for Final Deliverable	Kailen, Neha, Cassandra
Final Deliverable	Kailen, Neha, Cassandra

Learned Skills

Data Aggregation

Just like in the real world, it is not common to find a dataset that exactly fits the need of the task at hand. In this project, we will be exploring five separate dataset, and combining the best ones to suit our needs. We will aim for diversity and balance in our dataset to produce a model that is generalized sufficiently so that it can be applied to new data. With this approach, our model will be resistant to knowledge obsolescence, as new images of fields may be presented later on.

Checking Assumptions: Data Validity & Model Performance

A big component of this project will be assumption checking, primarily in the form of data verification. We need to ensure that the data we feed into our model is balanced and well distributed, as this will greatly impact model performance and lead to a “garbage in, garbage out” (GIGO) effect.

In addition, we must not assume accuracy to be the only validating metric. We must consider other metrics such as f1-score, f2-score, precision, recall, etc. Moreover, since we are utilizing multiple models in a “chained” manner, we must ensure that the output of the first model is accurate enough before connecting it to the next model.

Learning New Technologies: Machine Learning Models & Cloud Computing Platforms

We will be utilizing various models through this project. The first being YOLOv5 for object localization and detection for weeds and crops. The second model will be a CNN architecture like EfficientNet, VGG, or ResNet for our weed growth stage classification task. This will familiarize us with state-of-the-art models, and their application in the real-world.

A big drawback of training ML models is the computational cost and time needed. To speed this process up, we will need to utilize GPUs for their parallelization potential. One key platform for this is Google Colab, which offers the potential to utilize cloud computing, specifically, GPUs. Accessing this provides us with more computing power, allowing for the training process to be much quicker, rather than training locally on a personal CPU.

References

1. *Agriculture*. MDPI, vol. 11, no. 5, 2021, article 387, https://www.mdpi.com/2077-0472/11/5/387.
2. Gupta, Vatsal. *Weed Detection*. Kaggle, n.d., https://www.kaggle.com/datasets/vvatsalggupta/weed-detection.
3. NVIDIA. *EfficientNet*. PyTorch Hub, n.d., https://pytorch.org/hub/nvidia_deeplearningexamples_efficientnet/.
4. Ravirajsinh45. *Crop and Weed Detection Data with Bounding Boxes*. Kaggle, n.d., https://www.kaggle.com/datasets/ravirajsinh45/crop-and-weed-detection-data-with-bounding-boxes.
5. Ultralytics. *YOLO Common Issues: Deployment Challenges*. Ultralytics, https://docs.ultralytics.com/guides/yolo-common-issues/#deployment-challenges. Accessed 5 Feb. 2025.
6. Ultralytics. *YOLOv5*. PyTorch Hub, n.d., https://pytorch.org/hub/ultralytics_yolov5/.
7. University of California Agriculture & Natural Resources. *"Integrated Pest Management (IPM) – Agriculture."* UC ANR, n.d., https://ipm.ucanr.edu/agriculture/#gsc.tab=0.
8. Wang, A., et al. "*The VGG-SVM Weed Classification Model Proposed in This Study Provides a Valuable Reference for Intelligent Mechanical Weeding.*" *Plant Methods*,

vol. 18, no. 1, 2022, p. 69. BioMed Central, https://plantmethods.biomedcentral.com/articles/10.1186/s13007-022-00869-z.
9. *"Weed Detection and Classification Using Machine Learning." PubMed Central*, 2018, https://pmc.ncbi.nlm.nih.gov/articles/PMC5981438/.
10. *"Weed Detection Using Deep Learning Methods." SSRN*, 2021, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4007300.