**Crop and Weed Detection System: Reducing Pesticide Waste and Enhancing Crop Production**

Aryan Shrivastav

National Institute of Technology, Goa

**Crop and Weed Detection System: Reducing Pesticide Waste and Enhancing Crop Production**

The agricultural industry faces significant challenges due to the presence of weeds, which compete with crops for essential resources such as nutrients, water, and land. This competition results in decreased crop productivity and inhibits the optimal growth of desired crops. To control weeds, farmers often resort to the use of pesticides. However, the indiscriminate application of pesticides can lead to adverse effects, such as the contamination of crops and potential health risks for humans. Therefore, there is a need for a more targeted approach to weed control in agriculture.

The aim of this research is to develop a system that can accurately detect and differentiate between crop plants and weeds in agricultural fields. By doing so, the system will enable targeted pesticide application exclusively to weeds, minimizing the mixing problem with crops and reducing the waste of pesticides. This paper presents the data preparation steps, including dataset collection, cleaning, image processing, data augmentation, and manual labeling. Additionally, it addresses the problem of weed interference in agriculture and emphasizes the importance of developing a system that can effectively mitigate the negative impacts of weeds and pesticides.

# Abstract

This paper focuses on the development of a crop and weed detection system to reduce pesticide waste and enhance crop production. The dataset used in this study contains 1300 images of sesame crops and different types of weeds, with each image labeled in YOLO format. The data preparation process involved collecting 589 images, cleaning the dataset to remove irrelevant or misleading data, resizing the images to a manageable size, and augmenting the dataset using data augmentation techniques. Manual labeling of the images was conducted by drawing bounding boxes to differentiate between crops and weeds. The paper addresses the problem of weed interference in agriculture, highlighting the negative impact on crop productivity and the potential risks associated with pesticide use. The aim of the study is to develop a system that can accurately detect and differentiate between crops and weeds, allowing targeted pesticide application exclusively to weeds, thereby reducing the mixing problem with crops and minimizing pesticide waste.

# Data Preparation

In this section, we discuss the process of collecting and cleaning the dataset. It involves capturing photos of crops and weeds, followed by dataset cleaning to ensure data quality. The resulting dataset comprises 546 images after the cleaning process.

## Work Done

### Dataset Collection.

The dataset used in this study consists of 589 images captured from agricultural fields, including photos of both crops and different types of weeds. These images serve as the foundation for training the crop and weed detection system.

### Dataset Cleaning.

Once the dataset was collected, a crucial step was to clean and refine it. Dataset cleaning is essential to ensure that only relevant and accurate data is used for training the detection model. In this step, any images that were deemed irrelevant or potentially misleading were removed from the dataset. By eliminating low-quality or misleading images, the cleaning process helps enhance the performance of the detection model. After the cleaning process, the dataset was refined, resulting in a reduced dataset containing 546 images.

### Data Augmentation

Data augmentation is a technique used to increase the dataset size and enhance the model's ability to generalize. In this research, data augmentation was applied to the dataset to create additional variations of the existing images. By applying transformations such as rotations, flips, zooms, and shifts, the augmented dataset provides a broader range of training examples for the model. The Keras ImageDataGenerator, a popular tool in deep learning, was utilized to perform these augmentations automatically. The augmentation process resulted in an expanded dataset containing 1300 images. The increased diversity and variability in the dataset contribute to improving the model's performance and its ability to handle real-world scenarios.

### Image Processing

The images obtained from the dataset were initially of high resolution, with a size of 4000x3000 pixels. However, using such large images for training can significantly increase computational requirements and training time. Therefore, image processing techniques were employed to resize the images to a more manageable size of 512x512 pixels. This resizing process allows for efficient training of the model while preserving the color information present in the images. By reducing the image size, the computational burden on the model is reduced, enabling faster and more efficient training.

### Manual Labelling

Accurate labeling of the image data is a critical step in training a supervised machine learning model. In this research, manual labeling was undertaken to annotate the images with bounding boxes that indicate the presence of crops and weeds. This labor-intensive process involved visually inspecting each image and drawing bounding boxes around the respective regions of interest. By manually labeling the images, the dataset becomes properly labeled, enabling the model to learn and distinguish between crops and weeds effectively. The labeled dataset serves as the ground truth for training the crop and weed detection system, allowing it to make accurate predictions in real-world scenarios.

By meticulously addressing each aspect of the data preparation phase, including dataset collection, cleaning, image processing, data augmentation, and manual labeling, the research ensures the availability of a high-quality dataset and sets the foundation for developing an effective crop and weed detection system.

# Training

I will be using Tensorflow 2.0 Api for all the model and training purposes

## Preparing the data

I divided the overall training dataset into training data and validation data. Then use the test data provided to check the accuracy of the model. For achieving this we will use ImageDataGenerator from tensorflow.

## Model Architecture and Training

We create a convolutional neural network (CNN) architecture for image classification using the VGG16 model. The VGG16 model is loaded with pre-trained weights from the ImageNet dataset, excluding the fully connected layers. The head of the network consists of a flatten layer, followed by two dense layers with 4096 units and ReLU activation. Dropout layers are incorporated to prevent overfitting, and a final dense layer with softmax activation assigns probabilities to each class. The base model layers are frozen to retain the pre-trained weights, while only the head model is trainable. The model is compiled with the categorical cross-entropy loss function and Adam optimizer with a learning rate of 0.001. This architecture leverages the powerful feature extraction capabilities of VGG16 and demonstrates the effectiveness of the proposed approach for image classification tasks. By freezing the base model and training only the head, the model can efficiently learn and classify images with high accuracy.

## Model Evaluation

The trained model showed a precision of 98.3% while accuracy was about 97%

## Next week plan

Come up with a better ML model like SVM

Coming up with better hyperparameters

# Challenges/Hurdles

## Computation Resources

CNN models, especially large architectures like VGG16, require significant computational resources, such as high-performance GPUs, to train effectively. Limited access to such resources can hinder training or increase the training time significantly.

## Hyperparameter tuning

CNN models have various hyperparameters that need to be carefully tuned for optimal performance. These include learning rate, batch size, optimizer choice, and regularization parameters. Finding the right combination of hyperparameters often requires experimentation and can be time-consuming.

## Transfer Learning and fine tuning

When using pre-trained models like VGG16, adapting the model to the specific task at hand requires careful transfer learning and fine-tuning. Deciding which layers to freeze or update and how much to modify the network architecture requires expertise and experimentation.