# Weed Detection and Classification using Artificial Neural Networks

# 1. Introduction

Weed detection and classification play a crucial role in modern agriculture to ensure optimal crop growth and yield. In this Assignment, we propose an Artificial Neural Network (ANN) based solution for weed detection and classification. The solution involves the development of a convolutional neural network (CNN) to classify images into two categories: "crop" and "weed". The CNN model is trained on a dataset consisting of images of crops and weeds along with bounding box annotations.

# 2. Dataset and Preprocessing

The dataset used in this Assignment consists of images captured in agricultural fields containing both crops and weeds. Each image is accompanied by x\_center, y\_center, W, H annotations specifying the location of the crop or weed within the image. The dataset is preprocessed to extract regions of interest specified by the bounding boxes, which serve as input to the CNN model.

Example:

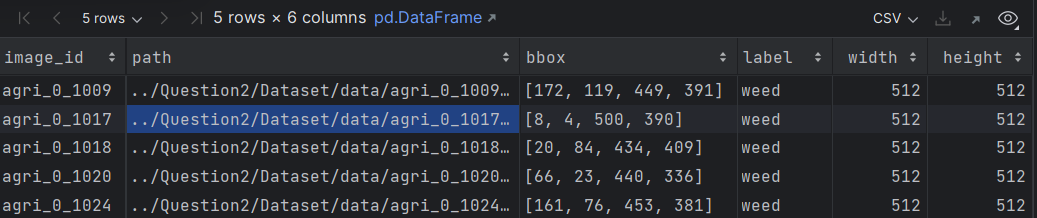


Corresponding Text File:

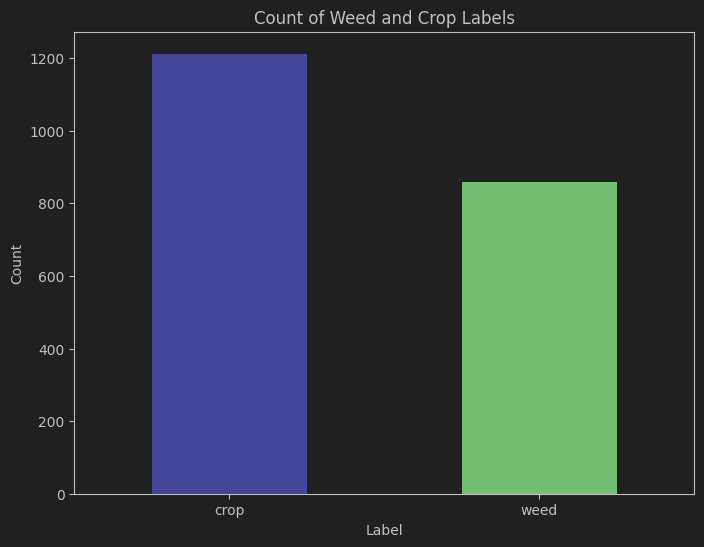
0 0.447266 0.407227 0.718750 0.669922

Dataset Link: <https://www.kaggle.com/datasets/jirayia/target-and-eliminate-weed?resource=download>

Data after some processing for the model like extracting the bbox coordinates and height and width:

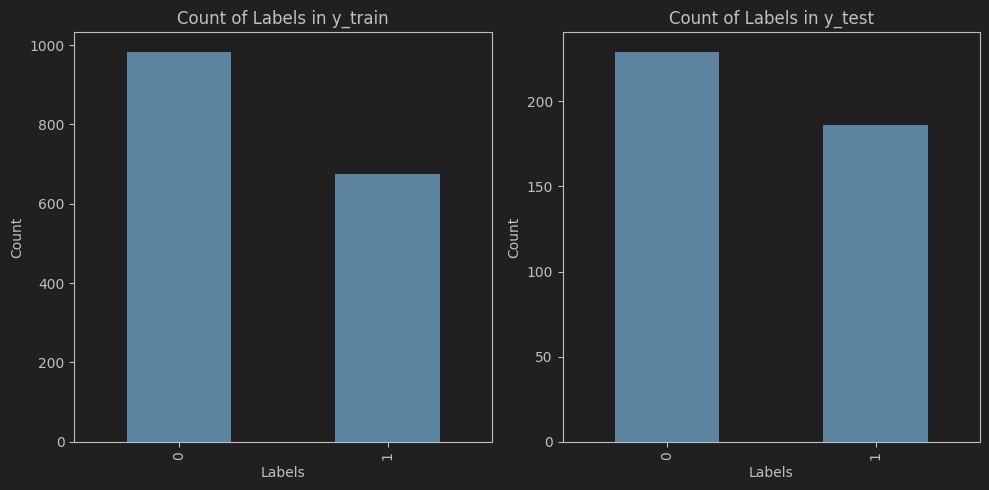


Dataset Weed Crop Ratio:



# 3. Dataset Train and Test Split:

The dataset is split into training and testing sets, with 80% of the data used for training and 20% for testing.

****

# 4. Model Architecture

The convolutional neural network (CNN) architecture used for weed detection and classification is designed to efficiently extract features from input images and classify them into two categories: "crop" and "weed". Below is a detailed explanation of each layer in the model:

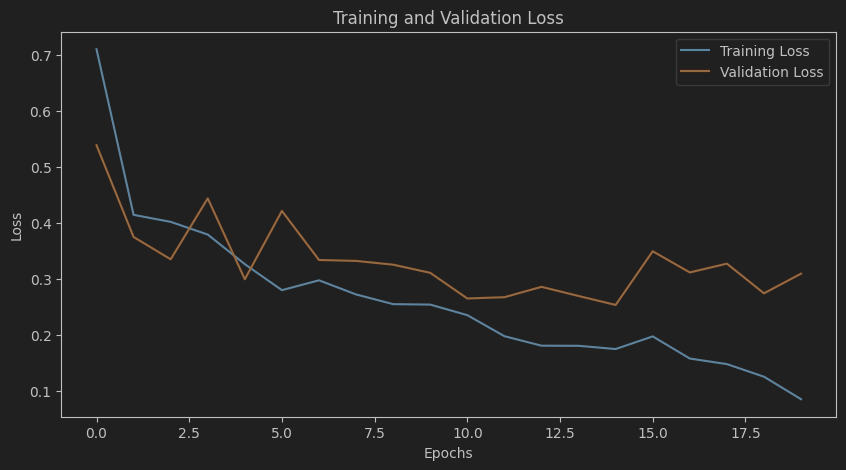
1. Input Layer:
   * The input layer receives the input images with dimensions specified by the input\_shape, which is determined by the shape of the training data (X\_train.shape[1:]).
2. Convolutional Layer 1:
   * The first convolutional layer (Conv2D) consists of 32 filters of size 3x3.
   * ReLU (Rectified Linear Unit) activation function is applied to introduce non-linearity.
   * This layer performs feature extraction by convolving input images with the filters to produce feature maps.
3. Max Pooling Layer 1:
   * Following the convolutional layer, a max-pooling layer (MaxPooling2D) with a pool size of 2x2 is applied.
   * Max pooling reduces the spatial dimensions of the feature maps, aiding in computational efficiency and increasing translational invariance.
4. Convolutional Layer 2:
   * The second convolutional layer consists of 64 filters of size 3x3.
   * ReLU activation function is again applied for introducing non-linearity.
   * This layer further enhances feature extraction by convolving the feature maps from the previous layer.
5. Max Pooling Layer 2:
   * Another max-pooling layer follows the second convolutional layer, with the same pool size of 2x2.
6. Convolutional Layer 3:
   * The third convolutional layer consists of 128 filters of size 3x3.
   * ReLU activation function is applied.
   * This layer continues to extract more complex features from the previous layers' outputs.
7. Max Pooling Layer 3:
   * A max-pooling layer with a pool size of 2x2 follows the third convolutional layer.
8. Convolutional Layer 4:
   * The fourth convolutional layer consists of 128 filters of size 3x3.
   * ReLU activation function is applied.
   * This layer further enhances the model's ability to capture intricate features.
9. Max Pooling Layer 4:
   * The final max-pooling layer with a pool size of 2x2 is applied.
10. Flatten Layer:
    * After the convolutional layers, the output is flattened into a one-dimensional vector, preparing it for input into the fully connected layers.
11. Dropout Layer:
    * Dropout is applied to reduce overfitting by randomly dropping a fraction (here, 0.5) of the neurons during training.
12. Dense Layer 1:
    * The first dense layer (Dense) consists of 512 neurons.
    * ReLU activation function is applied.
    * This layer further learns abstract representations from the flattened feature vector.
13. Dense Layer 2 (Output Layer):
    * The final dense layer consists of 2 neurons, corresponding to the two classes: "crop" and "weed".
    * Softmax activation function is applied to output probabilities for each class

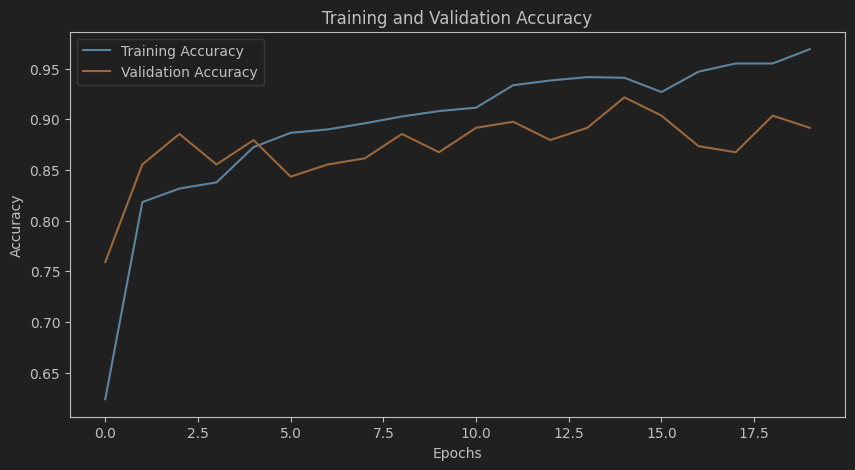
# 5. Model Compilation and Training

After defining the model architecture, it is compiled using the Adam optimizer with a learning rate of 1e-4. Sparse categorical cross-entropy loss function is chosen for multi-class classification, and accuracy is used as the evaluation metric. The model is then trained on the training dataset for 20 epochs with a batch size of 30. During training, the model learns to minimize the loss function by adjusting its weights and biases using backpropagation.

# 6. Results

The training and validation loss and accuracy graphs demonstrate the model's learning progress over epochs. After training, the model is evaluated on the test dataset to assess its performance on unseen data.





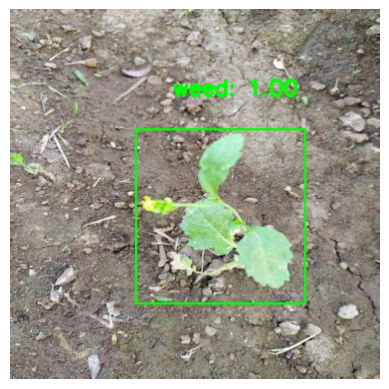
**Test Accuracy: 0.9445782899856567**

**Some Manual Test for Unforeseen Data:**

1. Actual Class: weed

****

1. **Actual Class: weed**



1. **Actual Class: weed**



# 7. Conclusion

The proposed ANN-based solution for weed detection and classification shows promising results in accurately identifying and categorizing crops and weeds in agricultural images. Further optimization and fine-tuning of the model could enhance its performance and applicability in real-world agricultural scenarios.