Identifying Crops from Weeds

Joachim Tan

1 Introduction

According to the United Nations [2], plants constitute a substantial 80% of our food supply. This crucial dependency on plants underscores the significance of understanding factors affecting crop growth. In a study conducted by Hewson et al [1], the detrimental impact of weeds on crop growth, resulting in reduced yields, was illuminated. These findings underscore the profound influence of weed-related challenges faced by farmers worldwide. Understanding the complex relationship between these factors highlights the need for advanced technologies. In particular, developing tools that can accurately identify crops among weeds shows great potential for improving weed management. These technological advancements are essential for protecting agricultural productivity and ensuring food security. In light of the importance of addressing weed-related challenges, this project aims to develop a Convolutional Neural Network (CNN) for plant classification.

2 Dataset & Data Preprocessing

The utilised dataset [3], has a collection of 5,539 RGB images capturing both crops and weed seedlings. These images are categorised into 12 distinct classes, each representing a prevalent plant species as seen in figure 1b The images of varying sizes are stored in the PNG format.

The images are all resized to a uniform size of 70x70 pixels. This serves multiple purposes, including the reduction of pixels, conserving memory, accelerating data loading, and expediting model convergence. While resizing optimises training efficiency, it's noted that it may result in a loss of fine-grained image details, potentially affecting model performance on larger images. However, because an input shape of 70x70 is specified for the model, input images should be resized to the same size before using them for predictions. The images and their corresponding labels are extracted from the dataset, transforming them into a more convenient data format. Specifically, the images are converted into a NumPy array, while the labels are organised into a pandas DataFrame. This data formatting choice enhances compatibility and streamlines the handling of data for subsequent machine learning tasks. The images undergo a series of processing steps to reduce noise and remove the background. It involves applying Gaussian blur with a kernel size of (5,5) to reduce image noise while preserving fine details. This choice of kernel size balances the need for noise reduction with the retention of image details.

Next, the images are converted from RGB to Hue, Saturation, Value (HSV) to facilitate colour-based segmentation. This transformation simplifies the process of specifying a desired colour range, green in this case. The chosen colour range is used to create a mask that isolates regions of the image falling within the specified colour range. To enhance the mask's completeness, an elliptical kernel is created and applied using a morphological operation called "closing". The elliptical kernel is chosen for its ability to capture a wide range of shapes and structures within the image compared to simpler shapes like squares or circles. Subsequently, this processed mask is employed to generate a boolean mask, which is then applied to the original plant image to effectively eliminate the background. This process involves utilizing the boolean mask to selectively copy only the regions of interest from the original plant image onto an empty canvas. In doing so, we successfully remove the unwanted background, leaving behind only the plant elements.

Finally, the processed plant images undergo normalisation, a crucial step involving the division of pixel values by 255. This normalisation process is employed because pixel values in typical images are conventionally depicted within the range of 0 to 255, where 0 signifies black, and 255 denotes white. By scaling these pixel values down to the range of 0 to 1 through division by 255, we align the data to a format better suited for machine learning models. This transformation enhances model training efficiency and fosters earlier convergence during the training process.

3 Model Creation & Training

The dataset is divided into three parts: 80% for training, 10% for validation, and 10% for testing. The use of a specified random state ensures reproducibility, and stratified splitting maintains the original class distribution, preventing class imbalance concerns. The model is designed as a CNN consisting of multiple convolutional layers with batch normalisation and dropout to extract features from input images. The chosen architecture gradually increases the number of filters in convolutional layers, allowing the model to capture complex patterns in the data. The use of batch normalisation helps stabilise training, and dropout reduces overfitting. Dense layers with ReLU activations are employed for classification, and the final layer uses softmax activation for multiclass output. The model is compiled with categorical cross-entropy loss and the Adam optimiser. Design choices, such as convolutional layers for feature extraction and dropout for regularisation, are made to improve model generalisation and classification performance while avoiding overfitting. The Adam optimiser is chosen as it is known for adapting learning rates during training, leading to fast convergence and better generalisation. The specified input shape ensures that input images must be resized to the same dimensions when making predictions, ensuring compatibility with the model's architecture. The model is trained with the aid of data augmentation, it enhances the training dataset by applying random transformations like rotation, zooming, and flips. This enriches the dataset and bolsters ability to generalise and handle diverse input variations. Throughout training, the validation accuracy is closely monitored, triggering a 0.4 reduction in the learning rate if no improvement is seen after 3 epochs. To safeguard peak performance, the model checkpoints store the best weights based on the highest validation accuracy achieved. These strategies collectively optimise training, ensuring the model's effectiveness and robustness.

4 Results

Figure 1 consists of 2 subfigures: 1a, a confusion matrix and 1b a plot of label type count. The confusion matrix reveals a correlation between the volume of data available for each class and the model's predictive accuracy. Specifically, classes such as "Loose Silky-bent" and "Common Chickweed," which have a larger number of images, exhibit a higher number of true positives. This suggests that a greater quantity of training data can enhance the model's ability to learn and subsequently predict these classes more accurately. Conversely, classes like "Black-grass" and "Common Wheat," which are underrepresented in the dataset, show a higher rate of misclassification. The scarcity of examples for these classes likely hampers the model's capacity to discern their unique features, leading to a less accurate performance. The confusion matrix indicates that "Black-grass" is often incorrectly identified as "Loose Silky-bent." This frequent confusion may be attributed to several factors, including the visual similarity between the two classes, the model's tendency to favour the more heavily represented "Loose Silky-bent" due to its abundance in the training data, and the insufficient number of "Black-grass" examples for the model to effectively learn from. "Loose Silky-bent" enjoys the most representation in the dataset, which correlates with its high classification accuracy. However, this overrepresentation may also lead to overgeneralisation, where the model overly associates the characteristics of "Loose Silky-bent" with other classes, as evidenced by its confusion with "Black-grass." The bar chart displays a significant imbalance in the dataset, with "Loose Silky-bent" having the highest number of images and "Common Wheat" the lowest. Such imbalance can introduce a bias in the model, where it performs well on classes with ample data but struggles with classes that are not as well represented.

5 Conclusion

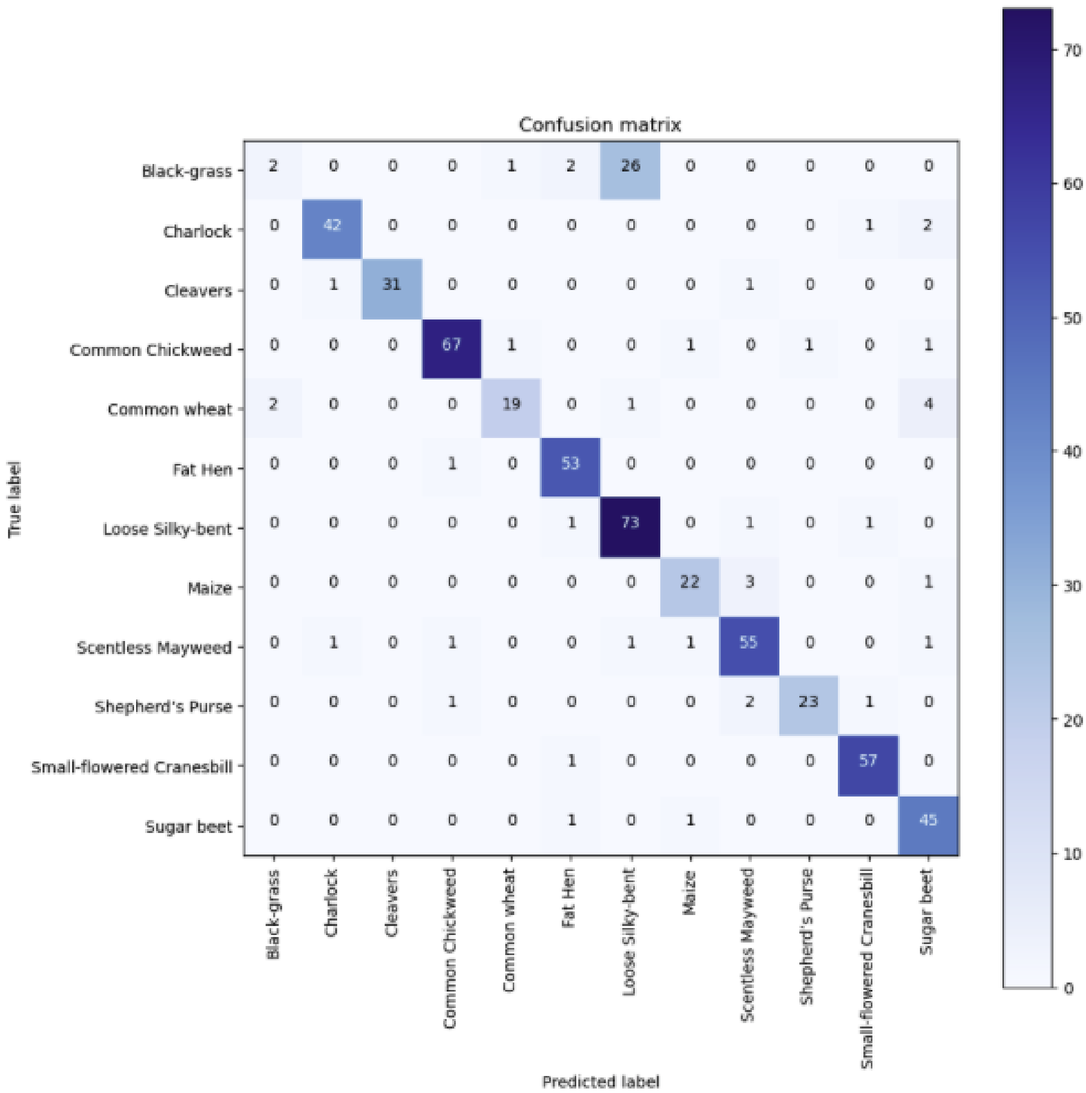
In conclusion, the performance of the model as seen in the confusion matrix can be affected by the distribution of data shown in the bar chart. The classes with more images tend to be predicted more accurately, while those with fewer images are not only predicted less accurately but are also more likely to be confused with other classes. Addressing the data imbalance through techniques such as data augmentation, oversampling the minority classes, or adjusting class weights during training could help improve the model's performance across all classes. Ensuring a balanced dataset is crucial for improving the model's ability to accurately classify all plant types, particularly those that are currently underrepresented.

References

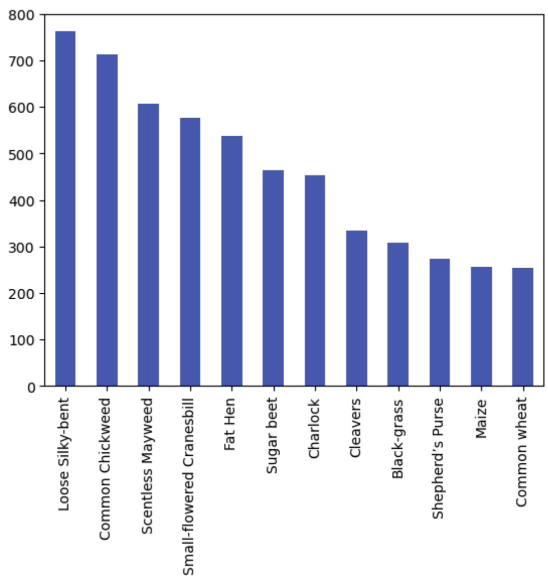
[1] R. T. Hewson and H. A. Roberts. Some effects of weed competition on the growth of onions. *Journal of Horticultural Science*, 48(1):51–57, 1973.

[2] United Nations News. Plants, the 'core basis for life on earth', under increasing threat, warns un food agency, 2019.

[3] VBookshelf. V2 plant seedlings dataset, 2023.



(a) Confusion Matrix



(b) Plot of label types count

Figure 1: Results