Project Report

Soybean Weed Detection and Identification using Aerial Images

Abstract:

The project aims to develop an ML model for weed detection and its species ' identification using a dataset of aerial images. The model is trained to automatically identify and classify the presence of weeds and classify its specie class. This project is intended for agricultural fields and help farmers to identify different weed species. This model can significantly aid farmers in managing weed infestations more efficiently and effectively. We will primarily be using Convolutional Neural Network (CNN) algorithm with various architectures to train our model that can accurately classify weeds and their species.

Introduction:

Weed infestation is a common problem in agriculture, leading to reduced crop yield and increased costs for farmers, resulting in lower profits. Current weed management practices often rely on manual labor, which is time-consuming, costly, and not always accurate. Nowadays, many modern farmers have started using aerial images from drones for inspecting large agricultural areas for production, but their potential for weed detection and species identification remains untapped. Hence, By automating weed detection and species identification through aerial images, our proposed model will aid farmers with a time-saving and cost-effective solution for managing weeds in their fields. It will allow precise targeting of weed-infested areas, reducing herbicide usage, which minimizes the environmental impact of these chemicals. Ultimately, the technology will help to increase crop yields and improve agricultural sustainability. Therefore, our primary aim is to develop an efficient and accurate model for weed detection and species identification using aerial images. The model should be capable of detecting weeds in real-time, distinguishing between different weed species, and providing insights to farmers for better decision-making. The most challenging part of this problem is Weed Appearance Variability, Real-time Processing, and Algorithm Selection according to the dataset. Also, for preparing our model for real-time application, we need to expose our model to explicit image datasets and test this in various external conditions.

About Dataset

Our original image dataset consists of 400 Images taken from UAV. After Segmentation we have around 15,336 total segments, out of which 3249 are of soil, 7376 of soybean, 3520 of grass, and 1191 of broadleaf weeds segments. The original images were captured by the UAV, among these, all those images with the occurrence of weeds were selected, resulting in a total of 400 images. Now, through the Pynovisão software, by using the SLIC algorithm, these images were segmented and the resulting segments were annotated manually with their respective class. These segments were used in the construction of this image dataset, which was finally utilized for this project.

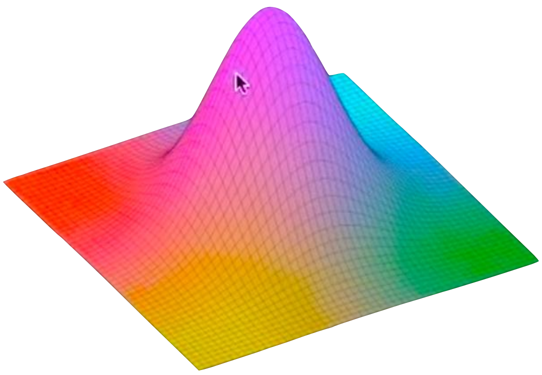
Image Processing

Image pre-processing: Images from the dataset first need to be preprocessed based on the type of noises present in them. This is also a crucial requirement for our segmentation algorithm to produce satisfactory output. Further, Image processing techniques are applied to enhance the quality of images before passing them through algorithms like Convolutional Neural Networks (CNN). Here are some suitable pre-processing techniques that will be employed for our dataset.

1. **Gaussian Blurring -** This technique is used to remove Gaussian noise that approximately follows a Gaussian distribution curve.

   The following equation will be used to construct our kernel for gaussian blurring

$$G(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Where σ^2 = variance

Similarly, we can extend this equation to two directions (x,y) or the x-axis and the the y-axis respectively

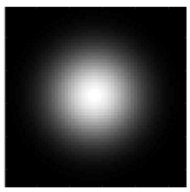
$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)} f[i-m, j-n]$$

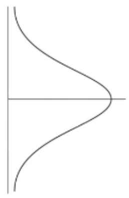
where x and y are the respective distances to the horizontal and vertical centre of the kernel and σ is the standard deviation of the Gaussian kernel.

1) **Median Blurring** - This method has been most effective for removing salt-and-pepper noises from the images. This type of noise is named so as this noise is present in the form of sprinkling salt and pepper in an image. Since we have used UAV images, hence a lot of these noise types may present in our dataset.

2) **Bilateral Blurring** - This method is capable to preserve the edges of an image, while still reducing noise. For weed detection, preserving edge information is crucial for distinguishing weeds from the plants, as weeds and plants have distinct visual characteristics. This capability is crucial for producing high-quality and accurate segments of our image. However, the largest downside to this method is that it is considerably slower than its averaging, Gaussian, and

$$BF[I]_p = \underbrace{\frac{1}{W_p}}_{\text{Normalization Factor}} \sum_{q \epsilon S} \underbrace{G_{\sigma_s}(||p-q||)}_{\text{Space Weight}} \underbrace{G_{\sigma_r}(|I_p - I_q|)}_{\text{Range Weight}} I_q$$





median blurring.

3) **Morphological Operations** - These operations are used in our weed detection dataset to enhance or extract important features, remove noise, and prepare images for further analysis. It is important to manipulate the size, shape, orientation, and structure of various objects within our dataset to maintain consistency. For our dataset, operations like Erosion, Dilation, opening, closing, Pruning, and Holes Filling are effective choices to standardize our dataset.

After preprocessing our Dataset, it becomes smoothened, Noise-free, and distortion-free, while preserving the important features of images. This processed data can now enhance the performance of our Convolutional Neural Network model, as it receives clean and consistent images, making it easier for the model to learn and distinguish between different classes (soil, soybean, broadleaf weeds, and grass weeds) during the training process.

Now, as we have Pre-processed our images, we can move forward with our Image Segmentation step which will be applied with the help of the SLIC algorithm. Here, the SLIC algorithm will take Pre-processed images as its input and will segment out the dataset objects classified under a category as output, into four classes as individual images in their respective directory. Also, the segments were then annotated with their respective class. The four directory namely soil, soybean, broadleaf weeds, and grass weeds contains their respective portion of these segmented objects.

Image Segmentation

Image Segmentation is one of the crucial steps in our project, as obtaining a accurately segmented datasets is essential for training our model with high-quality weed data. In our case, accurate segmentation enables us to precisely delineate weeds from other background elements in the images. In order to accomplish this crucial task, we have selected a high quality kaggle dataset. Please refer references for link to this dataset.

The dataset is prepared by using SLIC algorithm, which segmented the image data into four categories. The working of the SLIC algorithm is described below.

Simple Linear Iterative Clustering (SLIC) Algorithm

The SLIC is a superpixel segmentation technique commonly used in computer vision and image processing. Superpixel segmentation groups pixels into perceptually meaningful regions.

A super pixel refers to a group of pixels that are perceptually similar or belong to the same object or region within an image.

Important Notations:
- K= Number of super pixels (or Number of clusters)
- N= Total number of Pixels
- A= Approximate number of pixels in 1 Super Pixel
  A = N/K
- S= Approximate length of a super pixel
  S= sqrt(A)

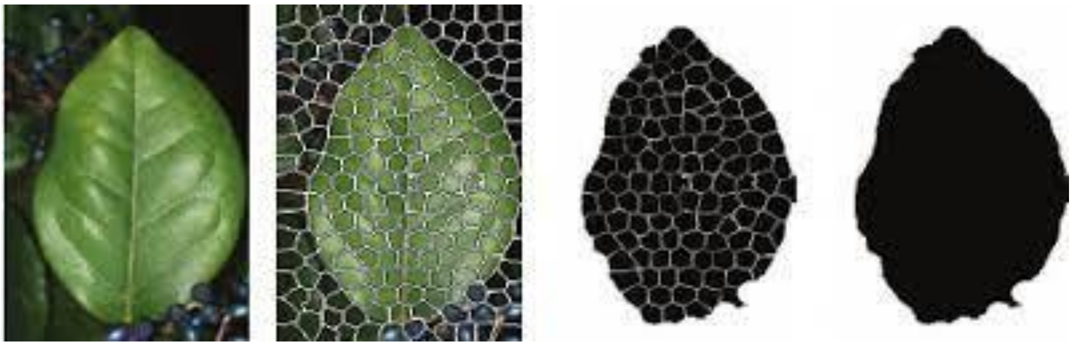
Theory
SLIC algorithm considers two types of distances such as colour distance and Euclidian distance between a pixel and a mean.

Each pixel can be represented as [l a b x y], where [l a b] is cielab space which is perpetually uniform for small colour distances.

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$
$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$
$$D_s = d_{lab} + \frac{m}{S} d_{xy} \ ,$$

Where m is a parameter which weights d_{xy} distance

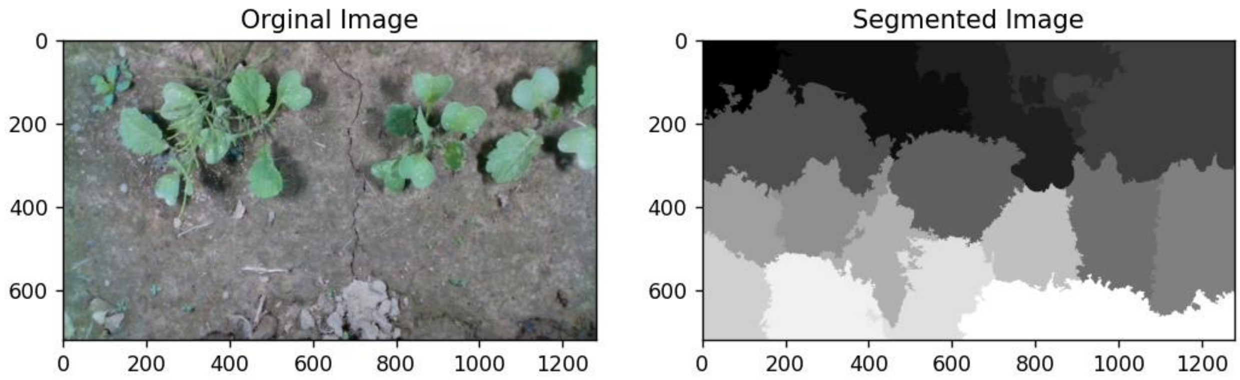
Note: In presence of noise the super pixel boundaries generated by the SLIC algorithm may become less accurate and may not closely align with the underlying object boundaries. So, the noise can lead to inaccuracies in the clustering process, potentially causing super pixel to merge or split incorrectly. Therefore, smoothening filters have been already applied to prevent this problem while using this algorithm



Here, are some representations of SLIC algorithm's output

Image Classification

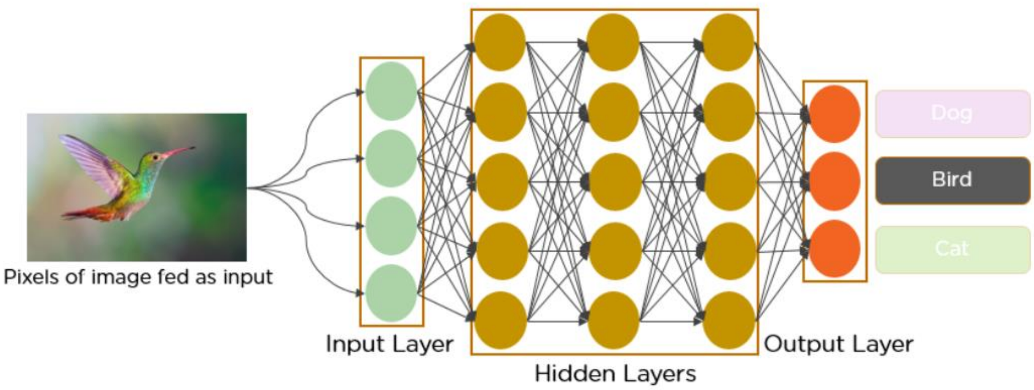
Our segmented dataset is now ready to be fed into the appropriate algorithm for training purposes. Once our model is ready, we will further test our model with testing dataset and another explicit dataset to verify its capability for on-field implementations. Considering our dataset, we will be using CNN architecture as it



has a lot of advantages over other architectures. The working of the CNN architecture is described below

Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a useful architecture that is used to extract the feature from data that contains matrix (like grids). The CNN architecture consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

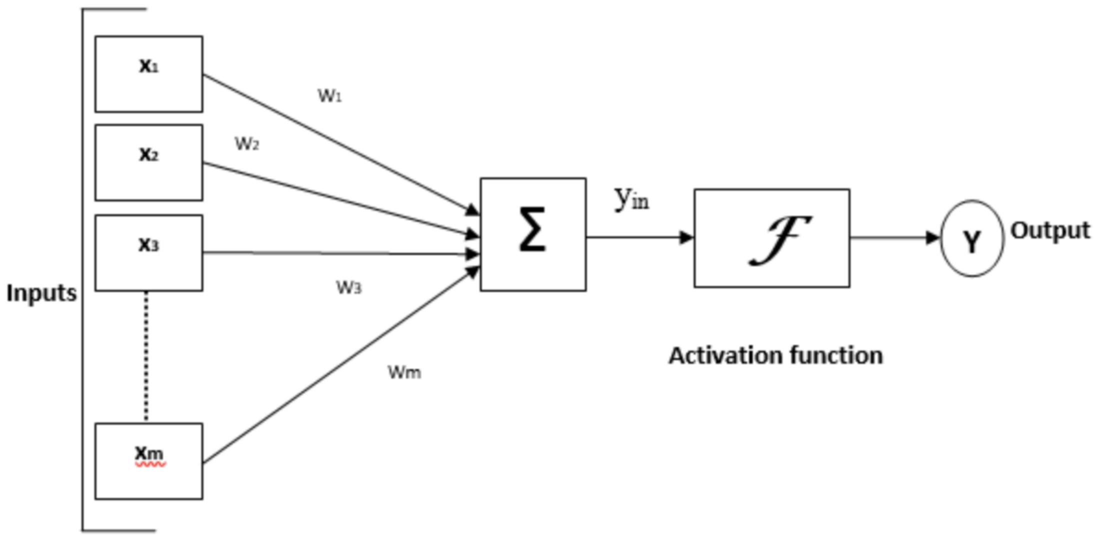


- **Feature Extraction**

  CNN-based feature extraction has convolutional layers, which use learnable filters to convolve across the input image. These filters also work as feature detectors and capture visual patterns like edges, corners, and textures at varying levels of abstraction. As the network deepens, these newly learned features become increasingly sophisticated, eventually encoding complex shapes and object parts. The Pooling layers complement convolutional layers by downsampling feature maps, which enables the network to learn spatial hierarchies and capture translational invariances. The resulting feature maps highlight important patterns while also reducing computational demands.

- **Image Classification**

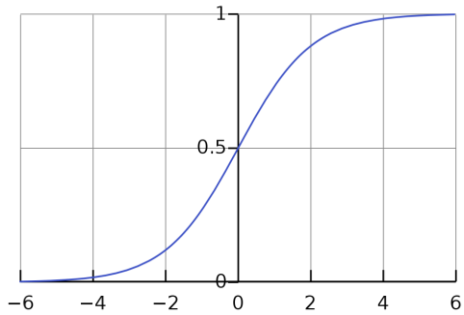
Convolutional neural networks are consist of several artificial neuron layers. Artificial neurons are a rough imitation of biological counterparts. There are mathematical functions calculate the sum of multiple inputs and outputs an activation When an image passes through a ConvNet, each layer several functions (such as ReLU, sigmoid, softmax, Heaviside function, etc) that are passed on to the next layer.



In our model, we will be using
Sigmoid, ReLU, and Softmax function
as activation functions. These functions are described below.

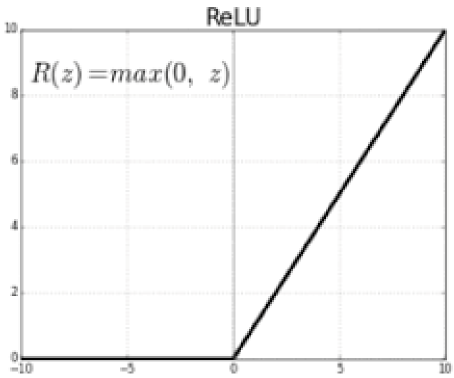
$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

- **Sigmoid Function:** The sigmoid function maps input values to a range between 0 and 1



$$S(x) = \frac{1}{1 + e^{-x}}$$

- **ReLU Function:** The Rectified Linear Unit (ReLU) function introduces non-linearity into the network, allowing it to learn complex relationships in the data and enabling the network to approximate a wider range of functions.



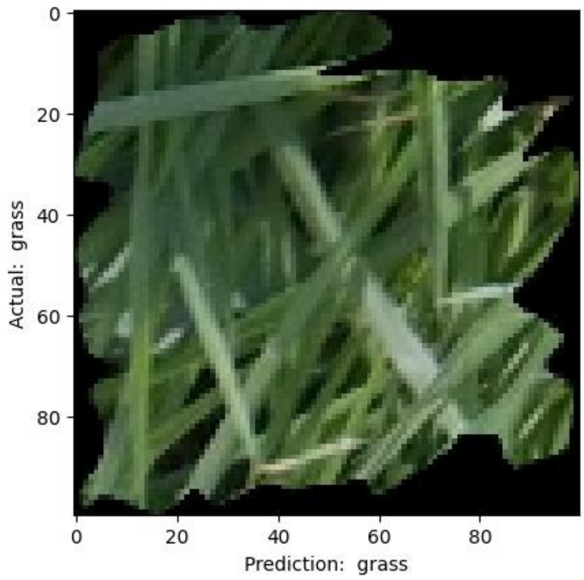
- **Softmax Function:** The Softmax function ensures that the predicted probabilities sum up to 1, making it suitable for assigning a probability distribution over multiple classes.

$$P(x) = \frac{P(x)}{P(x) + P(y)} \qquad P(y) = \frac{P(y)}{P(x) + P(y)}$$

Results

Our Model, after successful training and testing showed 89% of accuracy through CNN model. For verification and testing purposes we have used ANN and CNN models to compare the efficiency and accuracy for our dataset. We observed that ANN model showed 76% accuracy. On comparing both of these models we concluded that CNN model is more accurate than ANN model as it provides Spatial Hierarchies and Local Patterns, Parameter

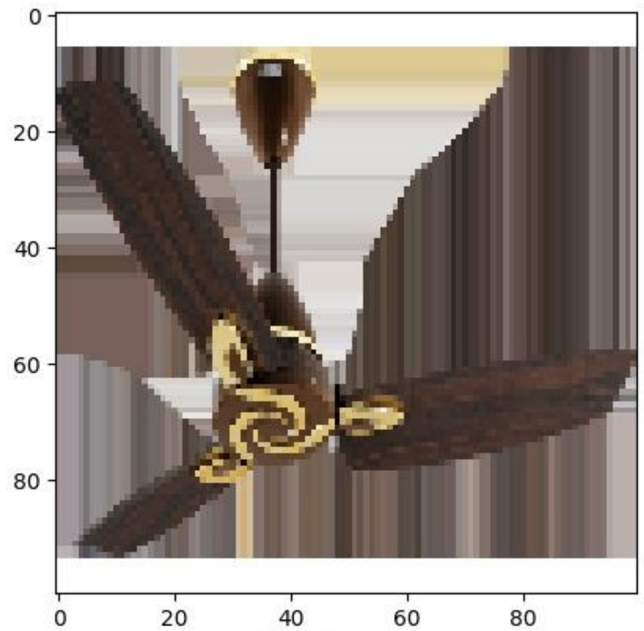
```
Text(0, 0.5, 'Actual:  grass')
```



Sharing, Efficiency for Images, Reduced Overfitting over other.

```
96/96 [==============================] - 1s 7ms/step - loss: 0.2878 - accuracy: 0.8996
[0.28776541352272034, 0.899608850479126]
```

Challenges: While testing our model with explicit Images, we observed



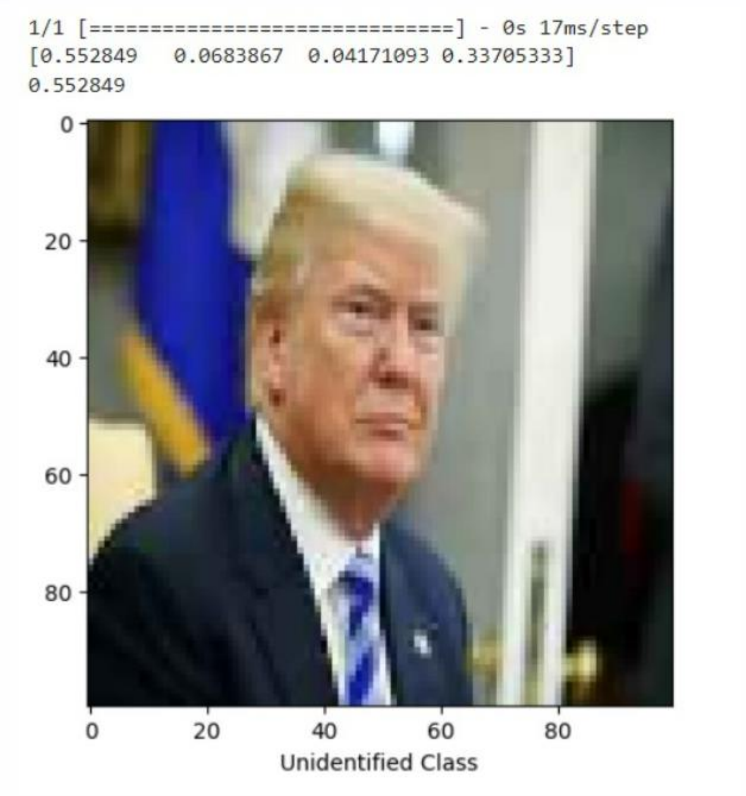
following problem in our model
Our model is misclassifying a fan image as grass, despite the fact that there is no data about fans in our training dataset, we figured out following possible reasons for this behavior:

1. **Confusion with Similar Patterns:** The features present in the fan image might resemble similar grass in our model. Even if our model hasn't explicitly seen fan images, it might be detecting certain textures, shapes, or patterns that are common in the grass and wrongly associating them with the fan image.

2. **Feature Combination:** Neural networks are capable to learn complex combinations of features from the training dataset. If certain combinations of features present in the fan image resemble to be the features our model learned from grass images, it could lead to misclassifications.

3. **Overfitting:** Overfitting occurs when the model learns the training data too well and fails to generalize it to new and explicit data. If our training dataset is small or unbalanced, our model might have memorized patterns in the training images without truly understanding the underlying concepts. This can lead to poor performance on new and unseen images.

We decided to solve this problem by using “Thresholding” method, as the wrong predictions are reaching a threshold. Hence, we provided a threshold value and conditioned our class prediction for that value. If value closes that threshold our model will flag those objects as “Unclassified” object.



Conclusions

In conclusion, the development of a weed detection model with an effective image processing module is a significant step toward addressing the weed infestation problem in agricultural fields. The image processing module plays a crucial role in preparing the suitable input image format for the weed detection algorithm by enhancing their quality, smoothening, removing noise, and extracting relevant features. Through a series of steps, including image preprocessing, segmentation, and feature extraction, the image processing module ensures that the weed detection algorithm receives consistent data. By employing advanced image processing techniques, such as color correction, thresholding, edge detection, and feature extraction, we can increase the model accuracy to distinguish weeds from the background and crops accurately. The success of the weed detection model heavily relies on the choice of the detection algorithm and the quality of the training dataset. Techniques like Convolutional Neural Networks (CNN) are a good choice for this kind of problem statement as CNN contains all the required tools for modeling an efficient classification model with satisfactory results.

References

- Dataset —> https://www.kaggle.com/datasets/fpeccia/weed-detection-in-soybean-crops
- SLIC: https://www.researchgate.net/publication/44234783_SLIC_superpixels
- https://ieeexplore.ieee.org/document/1056489
- https://projecteuclid.org/euclid.bsmsp/1200512986
- https://www.sciencedirect.com/science/article/pii/S0167865510000102
- https://pyimagesearch.com/2021/04/28/opencv-smoothing-and-blurring/
- https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/
- https://cs231n.github.io/convolutional-networks/