Computer Vision – Project

Research Paper

<u>Plant Species Classification</u>

Essa Athar Zuberi, M Mubeen Siddiqui, Syed Mujtaba Rizvi, Muhammad Ijlal

Abstract:

Plant identification is an important area of research that has gained much attention in recent years due to its applications in various fields, including agriculture, ecology, and conservation. In this study, we employed YOLOv8, a state-of-the-art deep learning algorithm, to identify different plant species in Karachi, Pakistan. To simplify the development process, we used the automated workflow of Roboflow, which enabled us to create and deploy our models with ease. We used web-scraping to collect images of the plant classes used in the test dataset, to train the model. And we collected a dataset consisting of images of various plants commonly found in Karachi and used it to test our model. Our model was able to predict a good amount for our results. Our study demonstrates the potential of deep learning algorithms for plant identification and highlights the importance of using automated workflows to simplify the development process. By contributing and improving to our findings, this research could be used to develop a mobile application for plant identification that could be used by farmers, ecologists, and conservationists in Pakistan and other regions.

Keywords: Plant identification, Object Detection, YOLOv8, Roboflow, Deep learning, Karachi, Pakistan, Agriculture, Ecology, Conservation.

1. Introduction

Before the expansion and popularity of computer vision, plant species were often identified based off visible features such as branch metrics, leaf structure, petal shape and other morphological factors [1]. These required trained experts who may or may not fully agree on the metrics of identification for such plant species. However, with the improvement of machine learning as well as advancement in the speed of training and processing such classification-based models, this application could be well documented and handled by new models trained on available datasets such as LeafNet, PlantNet and others. This document aims to provide available information as well as review currently existing methods within the research domain of implementing computer vision techniques to help categorize and identify plant species. The available solutions are based on multiple kinds of features which are identified through various neural network applications that will be further discussed below.

2. Methodology

Classical Methodology:

The classical approach for image classification involves acquisition of the plant images. This is then sent to a preprocessing unit to suppress any present distortions and enhance features that are relevant for further processing (such as filtering, denoising etc.). This output undergoes feature extraction which records the relevant details of the object (such as measurements) that exist within the area of interest within the image and is extracted via segmentation. In the final step, all extracted features are then concatenated into a feature vector which is classified [2]. These can be represented as a block diagram which is shown below:

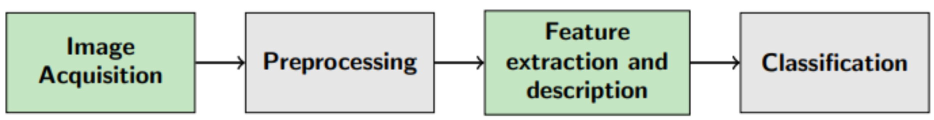


Fig 1. Block Diagram for Image Classification

In the context of plant species, the features of interest are dependent on the leaf structure, leaf type, and flower structure which corresponds to the respective plant of interest [2]. The following diagram illustrates sample features that can be considered for feature mapping:

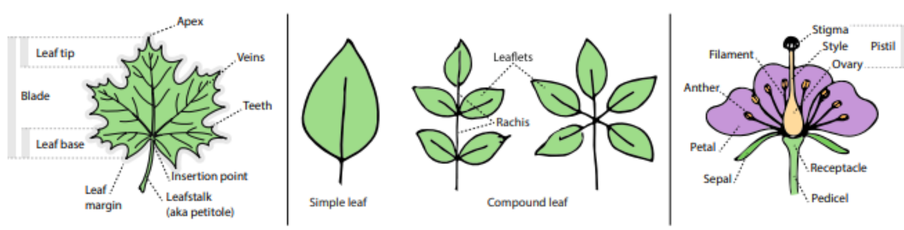


Fig 2. Sample Plant features useful for species classification

Morphology is the primary source of available features, however, other factors such as color and hue saturation may also be mapped for distinguishing plant species. This is done through image processing as the pixel intensities can reveal characteristics of color, texture as well as morphology. In the case of shape analysis, contour-based or region-based descriptors can be identified [2]. In the case of leaf analysis, the basic shape descriptors are identified as the basic geometric properties of the leaf (I.e., diameter, major and minor axis length, area, perimeter and centroid). These variables may also be assigned ratios for faster computation purposes. For example, the authors of the literature review [2] introduce a leaf width factor (LWF) which is computed after slicing the leaves from the major axis and parallel to the minor axis. A popular method for obtaining shape descriptors are Fourier descriptors (FDs) [2] [3], which contain Fourier harmonics for object outlines and being computationally efficient as only four coefficients need to be calculated. These descriptors primarily contain global shape features in the low-frequency terms and finer shape features in the higher-frequency terms.

Color space can also be utilized for assigning descriptors to the plant. This is done by multiple methods such as color moments, color histograms, color correlograms and color coherence vectors [2]. Other techniques found were edge detection and Gabor filter-based implementations [4]. This descriptor plays a greater role in species where leaf analysis may not be enough and flower analysis may also be incorporated to find distinguishable parameters.

The authors of [4] focused on classification of floral orchid species (which are likely part of flower analysis) by using a dataset of 60 orchid species. They found that color and texture features of the orchids resulted in the highest accuracy of 88.3% for species identification, while shape features were proven less effective.

According to paper [2], some of the recommended classifiers which the authors implemented on the Swedish leaf dataset were K-Nearest Neighbor Approach, CNNs, SVMs, Random Forests, and DBNs.

Our Methodology:

The following was some background on classical methodologies. However, we chose to automate almost the entire process, using the Roboflow workflow.

The workflow was as follows:

- Create a custom dataset
- Get the API Key code for the dataset
- Follow along the Google Colab Notebook, depending on the type of architecture being used, which in our case was YOLOv8.

3. Literature Review

Introduction of LeafNet, Identification of Plants based on Descriptors, and Plant Classification using PNNs:

The first research paper [13], "LeafNet: A computer vision system for automatic plant species identification" proposes a computer vision system for the automatic identification of plant species using leaf images. The proposed system, called LeafNet, utilizes deep learning algorithms, particularly the Convolutional Neural Network (CNN), for feature extraction and object detection, respectively. The LeafNet system consists of three main stages: leaf segmentation, leaf feature extraction, and plant species identification. The paper provides a detailed description of the methodology and experimental results. The authors evaluated the performance of the proposed system on three datasets (*LeafSnap*, *Foliage*, and *Flavia*) and achieved top-1 accuracies of 86.3%, 95.8%, and 97.9%, respectively. The results demonstrate the effectiveness of the LeafNet system for plant species identification using leaf images.

In the second research paper, the authors of [12] developed a plant identification system based on leaf shape descriptors. Their approach utilized Zernike moments (ZM) and Histogram of Oriented Gradient (HOG) methods as shape descriptors. The system was tested on a dataset of 50 different plants, resulting in an accuracy of 84.66% for ZM and 92.67% for HOG. Overall, the results indicated that HOG generated more robust shape descriptor features, making it a more satisfactory approach compared to ZM. The authors suggested that using these two feature extraction methods with a better classifier could further enhance the performance of their proposed system.

Finally, in the third paper [14], "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network", it presents a leaf recognition algorithm for plant classification using PNN. The proposed algorithm utilizes the shape, texture, and color features of leaves for classification. The feature extraction of the 12 Digital Morphological Features, derived from the 5 Basic Features, was done using Principal Component Analysis (PCA). The paper provides a detailed description of the methodology and experimental results. The authors evaluated the algorithm's performance on a dataset of 32 plant species and achieved an average accuracy of 90.312%. The results demonstrate the effectiveness of the proposed algorithm for plant classification using leaf images.

When comparing the three papers, it is clear that each proposed system has its strengths and limitations. The LeafNet system in the first paper showed high accuracy on all three datasets tested, demonstrating its ability to identify plant species using leaf images. The system proposed in the second paper showed promising results using HOG as a shape descriptor, but with limited testing on only 50 different plants. The third paper proposed a leaf recognition algorithm using the PNN, which showed high accuracy on the tested dataset of 32 plant species. However, the algorithm only utilized basic features for feature extraction, limiting its ability to capture complex features that may be crucial in plant species identification.

Overall, the proposed systems in these papers demonstrate the potential of computer vision and machine learning techniques in plant species identification using leaf images. However, further testing and improvements are needed to develop more robust and accurate systems that can handle a broader range of plant species and environmental conditions.

Further Methods for Identification of plant species:

When we think of detecting a plant species the first question that comes to mind is what part of the plant are we using in order to detect a specific plant. They can usually be identified by looking at features of different parts of the plant. It can be the stem of the plant, leaves of the plant or the flowers of the plant. But since we will be observing small roadside plants, the most feasible way to identify them seems to be identifying them based on the features of the leaves.

Traditionally, the features that biologists used to identify a plant species were based on the skeletal structure of the leaves, structure of veins of the leaves or the color of the leaves [8].

The next challenge is to identify a plant in a natural scenery, with lots of other plants around and with occluded and noisy conditions. Image processing techniques were used to separate out plant leaves and identify the plant. But the major problem with this technique was that it was very difficult to identify the occluded and overlapping leaves of the plant. With the advancement in deep learning and computer vision techniques now the accuracies of feature extraction, identification and localization of a plant in a noisy environment have reached a very good level. R-CNN, Fast R-CNN and Faster R-CNN are mainly used now in object detection and localization problems. But even with these techniques, detection of overlapping plant leaves is still an issue. In order to solve this issue, the paper [5] has proposed a modified Faster RCNN model. The authors of this paper used ResNet50 as their backbone network and added a convolutional block attention module (CBAM) before the first layer and after the last layer of the backbone network. The CBAM is a hybrid module that has the spatial attention module (SAM) and the channel attention module (CAM). This combination enhances the network's ability to learn target features and their respective locations by adjusting the weights of the CBAM. By doing so, the CBAM can effectively redistribute attention to relevant areas of the input. The authors of this paper [5] have also proposed distance-Intersection over Union NMS (DIoU-NMS) replacing the previously used NMS in order to cope with the problem of identification of overlapping leaves. This proposed method used the center-to-center distance of the intersecting object identifying boxes to decide if a box should be deleted in a densely distributed and occluded region proposal situation. With these modifications they achieved 2.9 % higher accuracy than the original Faster-RCNN model.

Now, building a model from scratch and training and testing the model takes a lot of time and computing power. Most of the time due to limited resources and inadequate amount of dataset the model does not perform as expected. In order to avoid this situation, transfer learning technique is used instead of building the model from scratch. A pretrained backbone model is used for feature extraction and only a few layers are trained with the local dataset in order to get better accuracy with lesser resources. If we are to use the model proposed above, then we can use a pretrained ResNet50 as our backbone network or we could replace it with another widely used backbone network such as VGG16 AlexNet or GoogLeNet etc. as proposed by the paper [6]. This paper's authors analyzed the performance of pretrained deep-learning models by optimizing the hyperparameters for the problem of plant identification, and the models showed very good results.

The authors of the paper [7] have proposed a methodology of implementation of Faster-RCNN with some modifications in order to classify plants on the basis of their flowers. This method also makes use of the transfer learning and ResNet-50 V1 is used as the backbone network. NAS-FPN is used in integration with the pretrained backbone ResNet-50 V1 model. FPN is a pyramid representation that combines low-resolution with strong semantic features and week semantic features with high resolution via top-down and lateral connections. According to the paper NAS-FPN network provides a flexible and efficient way for building accurate object detection model. A pre-trained model NAS-FPN shares feature map weights to ResNet 50 V1 and then the process of feature extraction is performed by using fine-tuned convolution neural network backbone architecture and then the strong feature maps are passed to the RPN which is then connected to the Faster RCNN head for object detection. The model gets 96.2 percent mAP score on flower 30 dataset.

Looking at all these methods it is evident that we must use a pretrained backbone model with Faster-RCNN for plant detection based on images of their leaves and if the accuracy does not go very well then, we should integrate the modifications as proposed by the above papers in order to improve the accuracy.

The AlexNet paper introduced a deep convolutional neural network architecture that achieved state-of-the-art performance on the ImageNet Large Scale Visual Recognition Challenge, a benchmark for object recognition. The network comprised eight layers, including five convolutional layers, two fully connected layers, and a softmax output layer. The authors used data augmentation and dropout regularization to reduce overfitting, and trained the network using stochastic gradient descent with momentum. [9] The resulting model achieved a top-5 error rate of 15.3%, significantly outperforming previous methods.

This study explored the use of transfer learning with convolutional neural networks for plant species classification. The authors used three pre-trained networks – VGG-16, ResNet-50, and Inception-v3 – [10] and fine-tuned them on a dataset of plant images. They found that fine-tuning the pre-trained networks significantly improved classification performance compared to training from scratch. In particular, the VGG-16 network achieved the highest accuracy of 96.6%.

The EfficientNet paper proposed a new method for scaling convolutional neural networks that achieved state-of-the-art performance on the ImageNet benchmark while requiring fewer parameters and computations than previous models. The authors introduced a compound scaling method that simultaneously scales the depth, width, and resolution of the network. [11] They also used a novel scaling coefficient that balances the trade-off between accuracy and computational cost. The resulting models achieved top-1 accuracy of up to 88.4% and top-5 accuracy of up to 97.1%, while being up to 8.4x smaller and 6.1x faster than previous models.

In summary, these three studies demonstrate the importance of deep neural networks in image recognition and classification tasks, as well as the value of transfer learning and efficient model scaling techniques for improving performance and efficiency.

4. Dataset

The Dataset is publicly available on the Roboflow's universe and can be accessed here.

Augmentation Techniques:

- Outputs per training example: 2
- Blur: Up to 16.75px
- Bounding Box: 90° Rotate: Clockwise, Counter-Clockwise, Upside Down
- Bounding Box: Shear: ±15° Horizontal, ±15° Vertical

5. YOLOv8

YOLO (You Only Look Once) is a popular set of object detection models used for real-time object detection and classification in computer vision. Originally developed by Joseph Redmon, Ali Farhadi, and Santosh Divvala, in the famous research paper [15], YOLO aims to achieve high accuracy in object detection with real-time speed. The model family belongs to one-stage object detection models that process an entire image in a single forward pass of a convolutional neural network (CNN). The key feature of YOLO is its single-stage detection approach, which is designed to detect objects in real time and with high accuracy. Unlike two-stage detection models, such as R-CNN, that first propose regions of interest and then classify these regions, YOLO processes the entire image in a single pass, making it faster and more efficient.

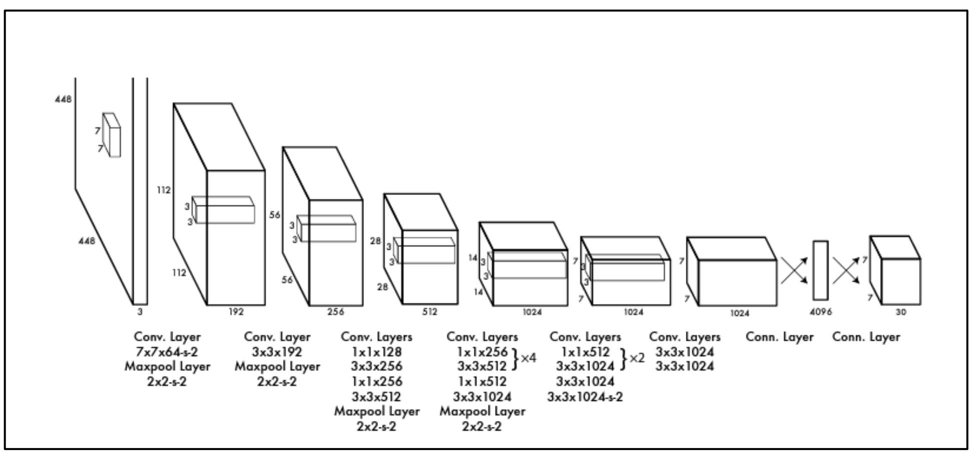


Fig 3. The YOLO architecture

However, YOLOv8 [16], still under deployment, has quite a complex architecture. YOLOv8 is the newest state-of-the-art YOLO model that can be used for object detection, image classification, and instance segmentation tasks. YOLOv8 was developed by Ultralytics, who also created the influential and industry-defining YOLOv5 model. YOLOv8 includes numerous architectural and developer experience changes and improvements over YOLOv5. YOLOv8 is under active development, as Ultralytics is working on new features and responding to feedback from the community.

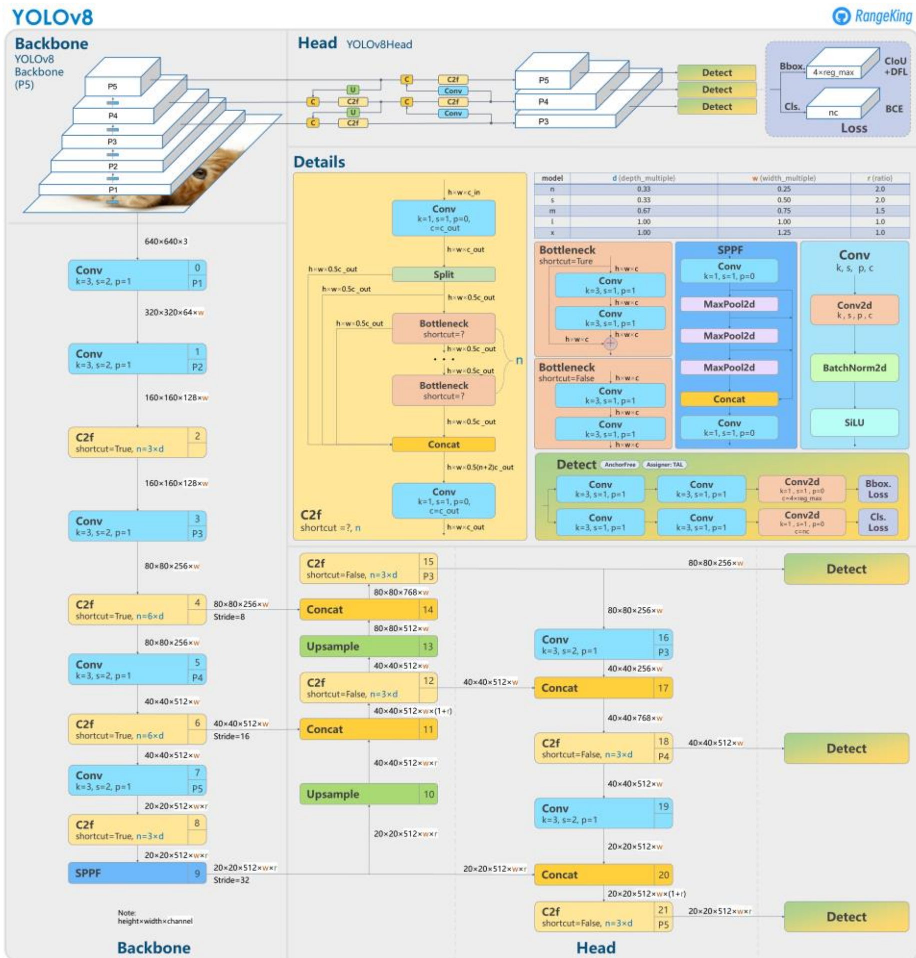


Fig 4. The YOLOv8 architecture

6. Training and Validation

A total of 178 training images were passed to the model for training and 20 validation images, which included augmentation and pre-processing techniques. The training took 0.111 hours to complete 25 epochs. The results were as follows:

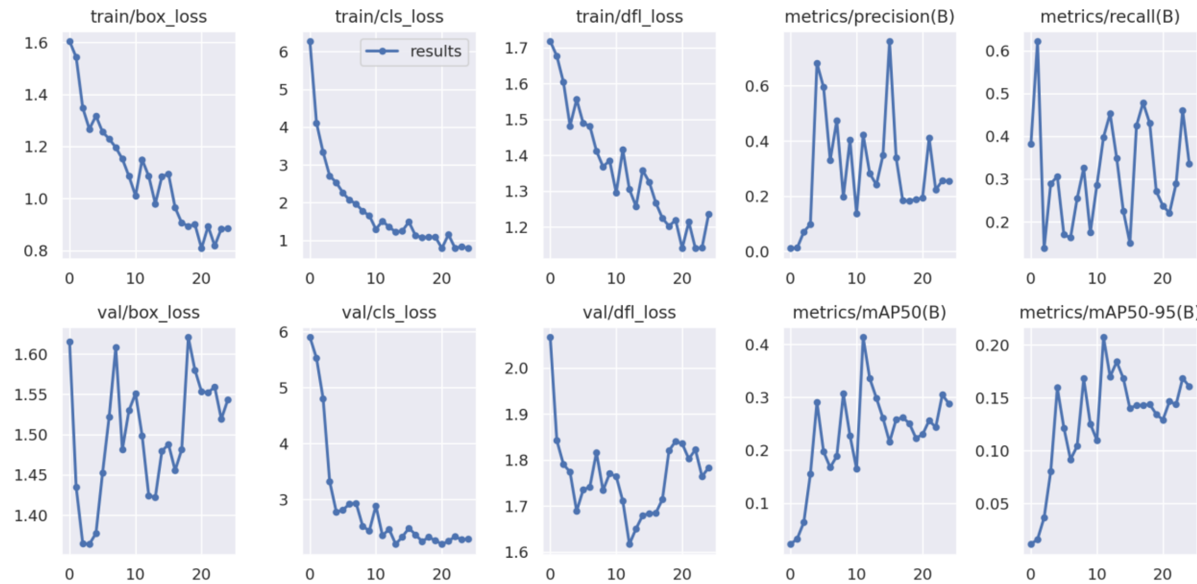


Fig 5. Training results (losses and metrics) on YOLOv8

All the different losses significantly went down during training. The precision and recall results fluctuated a lot throughout. The model performed well on the validation set, as the losses went down, and the mAP score increased.

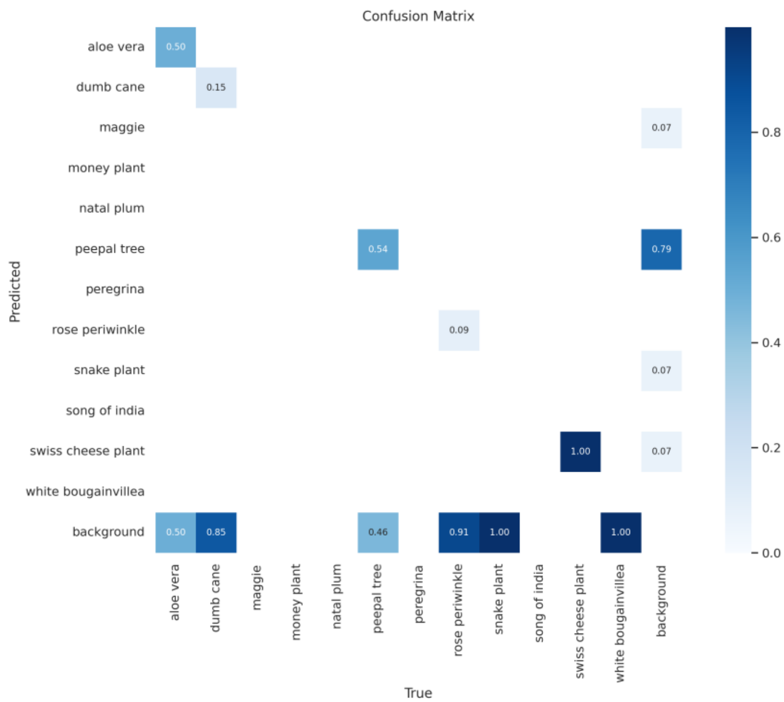


Fig 6. Confusion Matrix

The validation batch produced good results, as follows:

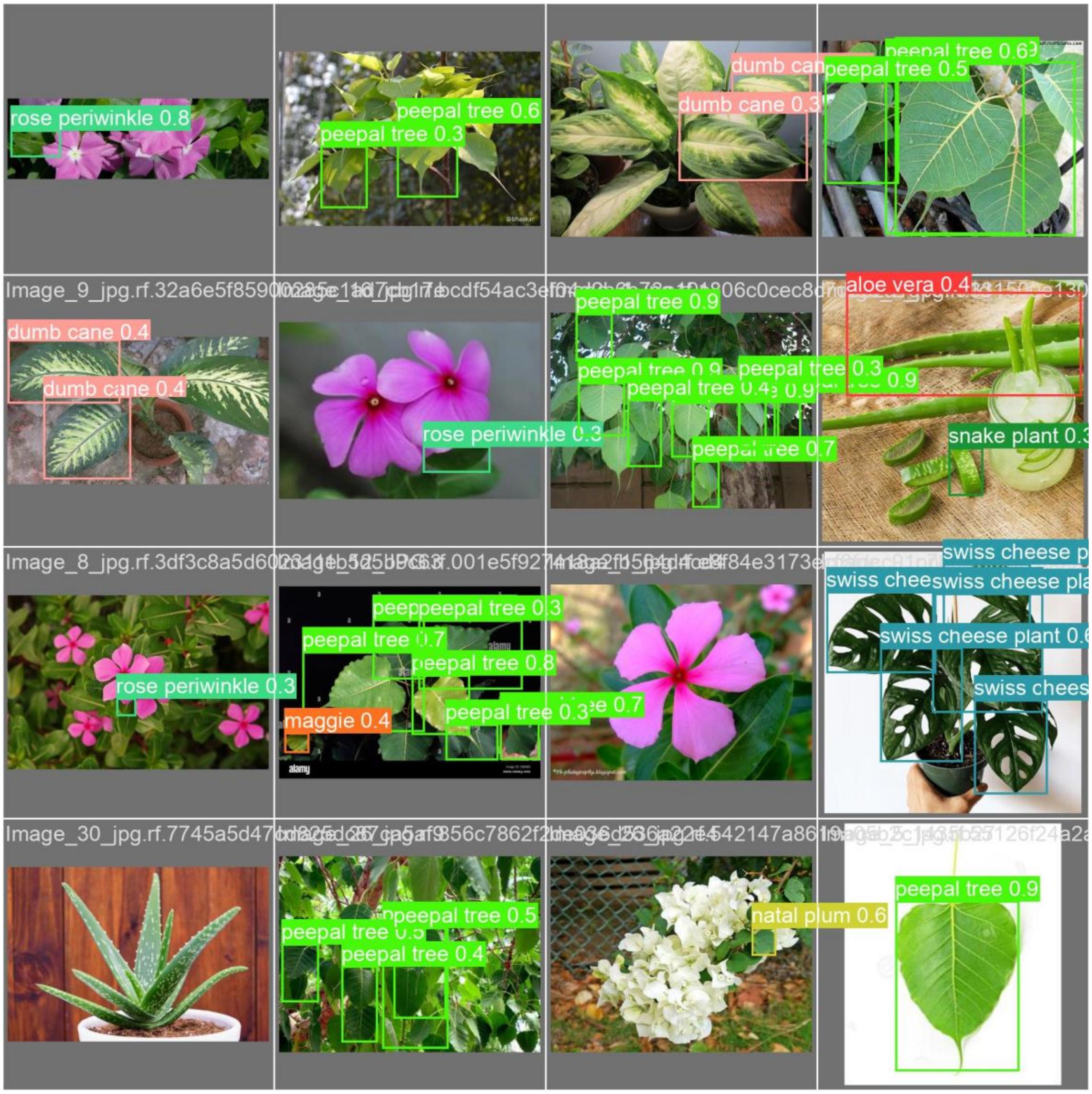


Fig 7. Validation Batch Results

The scores for *aloe vera* and *swiss cheese plant* were extremely high and resulted in mAPs of 0.995 for both classes:

```
25 epochs completed in 0.111 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.20 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11130228 parameters, 0 gradients, 28.5 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%
                   all         20         57      0.424      0.398      0.414      0.207
             aloe vera         20          2          1      0.903      0.995      0.255
             dumb cane         20         13      0.519      0.251      0.282      0.246
           peepal tree         20         24      0.509      0.542      0.523      0.276
       rose periwinkle         20         11       0.24     0.0909     0.0849     0.0349
           snake plant         20          1          0          0          0          0
     swiss cheese plant        20          4      0.698          1      0.995      0.634
   white bougainvillea         20          2          0          0     0.0148    0.00463
Speed: 0.3ms pre-process, 8.7ms inference, 0.0ms loss, 1.1ms post-process per image
Results saved to runs/detect/train
```

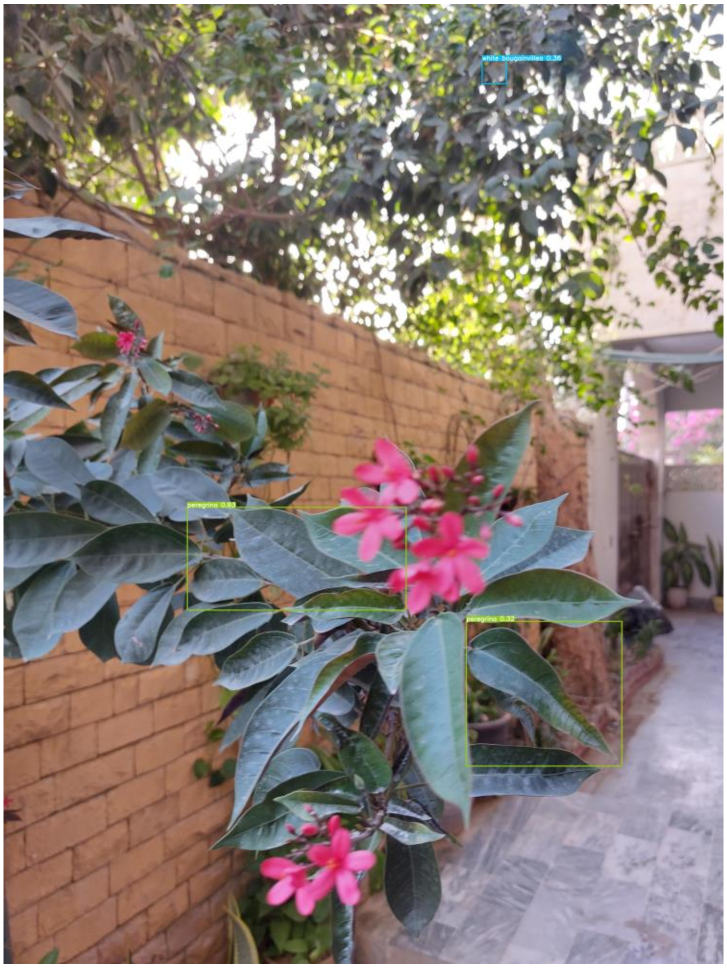
Fig 8. Training and Model Summary

7. Testing

The model performed just above average on the test dataset, on 17 test images. Some of the results were as follows:


snake plant 0.48
snake plant 0.55
snake plant 0.46
snake plant 0.62
snake plant 0.50
snake plant 0.60


aloe vera 0.90
aloe vera 0.26
money plant


white bougainvillea 0.36
peregrina 0.93
peregrina 0.32

8. Challenges

There were a dozen challenges faced at the start. The gathering of the training dataset took the most time. We looked at a couple of large datasets on plants (Leafsnap, Foliage, PlantVillage), but were stumbled upon on the exclusion of classes compared to our test dataset classes. One of the reasons, if not the only, for not achieving state-of-the-art results was the inaccuracy of making annotations (bounding boxes). Even with an online platform (Roboflow), the annotation procedure was time-consuming and of great hindrance. Talking about the difficulties in the choosing of architecture, we initially decided to proceed with the Faster-RCNN architecture. This approach led us less than halfway in the research as the complexity to implement parts of the architecture, was quite in the way (RPN, ROI Pooling etc.). The second approach was to use just the RCNN architecture. This only had too many drawbacks, performance of the model being one (Selective Search complexity).

References:

[1] P. Bonnet, A. Joly, and C. Davis, *Plant Biodiversity Science in the Era of Artificial Intelligence*. Frontiers Media SA, 2022.

[2] J. Wäldchen and P. Mäder, “Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review,” *Archives of Computational Methods in Engineering*, vol. 25, no. 2, pp. 507–543, Jan. 2017, doi: https://doi.org/10.1007/s11831-016-9206-z.

[3] N. Jamil, N. A. C. Hussin, S. Nordin, and K. Awang, “Automatic Plant Identification: Is Shape the Key Feature?,” *Procedia Computer Science*, vol. 76, pp. 436–442, 2015, doi: https://doi.org/10.1016/j.procs.2015.12.287.

[4] Apriyanti, Diah Harnoni, Aniati Murni Arymurthy, and Laksana Tri Handoko. "Identification of orchid species using content-based flower image retrieval." *2013 International conference on computer, control, informatics and its applications (IC3INA)*. IEEE, 2013.

[5] M. Wang, B. Fu, J. Fan, Y. Wang, L. Zhang, and C. Xia, “Sweet Potato Leaf detection in a natural scene based on faster R-CNN with a visual attention mechanism and Diou-NMS,” *Ecological Informatics*, vol. 73, p. 101931, 2023.

[6] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, “Plant identification using deep neural networks via optimization of transfer learning parameters,” *Neurocomputing*, vol. 235, pp. 228–235, 2017.

[7] Patel, Isha & Patel, Sanskruti. (2020). “An optimized deep learning model for flower classification using NAS-FPN and faster RCNN.” *International Journal of Scientific & Technology Research.* 9. 5308-5318.

[8] Zhang, Liankuan; Weckler, Paul; Wang, Ning; Xiao, Deqin; Chai, Xirong (2016). “Individual leaf identification from horticultural crop images based on the leaf skeleton.” *Computers and Electronics in Agriculture, 127(), 184–196.* doi:10.1016/j.compag.2016.06.017

[9] Liu, Shuying, and Weihong Deng. "Very deep convolutional neural network based image classification using small training sample size." *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. IEEE, 2015.

[10] Pan, Q., Jia, M., Liu, Q., Zhang, L., Pan, J., Lu, F., ... & Ge, H. (2021). Identifying Patient–Ventilator Asynchrony on a Small Dataset Using Image-Based Transfer Learning. *Sensors*, *21*(12), 4149.

[11] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning*. PMLR, 2019.

[12] Pradip Salve, Milind Sardesai, Ramesh Manza, Pravin Yannawar. Proceedings of the Second International Conference on Computer and Communication Technologies, 2016, Volume 379

[13] Pierre Barré, Ben C. Stöver, Kai F. Müller, Volker Steinhage. LeafNet: A computer vision system for automatic plant species identification, Ecological Informatics, Volume 40, 2017, Pages 50-56, ISSN 1574-9541, https://doi.org/10.1016/j.ecoinf.2017.05.005.

[14] S. G. Wu, F. S. Bao, E. Y. Xu, Y. -X. Wang, Y. -F. Chang and Q. -L. Xiang, "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network," 2007 IEEE International Symposium on Signal Processing and Information Technology, Giza, Egypt, 2007, pp. 11-16, doi: 10.1109/ISSPIT.2007.4458016.

[15] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[16] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics