Summer Research Fellowship Program, 2023

Vision for Agriculture Good

Om Surase, ENGS2316
Vishwakarma Institute of Technology

Guide: Dr. Shanmuganathan Raman
IIT Gandhinagar

Abstract

In modern agriculture, there is a pressing need for advanced technologies to ensure food security and sustainable agricultural practices. Computer vision, with its capacity to analyze visual data, offers a promising solution. The project, "Vision for Agriculture Good" addresses efficient resource management in agriculture by employing computer vision techniques for segmentation on cotton field images. This project focuses on implementing latest state of art instance segmentation algorithm (YOLOv8) to identify and delineate individual cotton bulbs within field images. Due to the unavailability of an annotated cotton field dataset during the project's initiation, an apple field dataset was utilized to simulate and prototype the envisioned solutions. Weights of the YOLOv8 model pre-trained on the COCO dataset were used for transferring learning using the apple farm dataset. The model is capable of performing instance segmentation on apples farm images into two classes: healthy and unhealthy apples. Despite its potential, implementing computer vision in agriculture poses challenges. The absence of specialized and annotated datasets for specific crops, such as cotton, is a significant hurdle. Apart from this, varying lighting conditions and complex farm landscapes also pose a challenge and affect the model’s predictions. Annotation of field images is a labor-intensive and time-consuming process, creating the biggest bottleneck for the application of computer vision in agriculture.

Contents

1. Introduction
    a. Need for deep learning in agriculture.
    b. Traditional image segmentation approaches.
    c. Deep learning approach.
    d. Aim.

2. Deep Learning for instance segmentation.
    a. Neural Network basics.
    b. Yolov8 architecture.

3. Methodology

4. Results

5. References

1.Introduction

1.a Need of deep learning in Agriculture

Pixel Wise identification of crops in field images is crucial for numerous agriculture-related applications, including crop yield monitoring and growth and health assessment. This data plays a vital role in subsidy management, as well as research and decision support for diverse agricultural objectives. Initially, such data was predominantly utilized by governmental agencies. However, in the current landscape of the burgeoning agricultural technology sector, commercial businesses also demand field data for applications such as farm management, yield forecasting, and precision farming.

Initiatives promoting the adoption of the latest technologies in farming include the launch of the Digital Agriculture Mission (2021-25) by Government of India. This mission aims to encourage and expedite agriculture-related projects based on cutting-edge technologies, including AI, blockchain, remote sensing, robots, and drones. Thus, highlighting the significance of use of the latest technology in agriculture. Currently, high-quality field images undergo manual segmentation by experts with experience in identifying healthy/unhealthy target crops, weeds, insects, and unwanted growth, as shown in figure 1. This task, while accurate, is repetitive and can also be performed by individuals with sufficient agricultural expertise. However, manual segmentation becomes time-consuming for plantations with a larger number of plants or crops that are inherently more numerous (for example, the number of useful rice crop parts is naturally greater than the number of apples in a field of the same size). Additionally, manual segmentation is susceptible to intra and

inter-observer variability due to varying experience, performance, and diligence among human operators. Thus it is a crucial need of the agricultural sector, to have an automated system who can perform the task of segmentation with accuracy and efficiently.

Figure 1: Raw image of apple tree(left) and segmented image of apple tree(right).

Agricultural farms in images captured using drones or handheld cameras are relatively simple image objects when compared to most objects in natural photos. E.g., a picture of a human body appears much more complex because it consists of a variety of subobjects with different textures and shapes (face, hands, clothes, ...). Nevertheless, designing an algorithm for the delineation of agricultural parcel objects is non-trivial, both via traditional image segmentation as well as with deep learning techniques. The model needs to correctly dismiss undesired image objects, differentiate between adjacent objects with nearly similar spectral properties or barely visible borders, and correctly delineate field objects with multiple homogenous areas.

1.b Traditional image segmentation approaches.

Traditional image segmentation are based on edge detection (Rydberg & Borgefors 2001, Mueller et al. 2004, Turker & Kok 2013), image value gradients (Butenuth et al. 2014), deformable “snake” algorithms (adapting to vector contours) (Torre & Radeva 2000) and

textural properties (Da Costa et al. 2007, Tiwari et al. 2009, Yalcin et al. 2016). Another approach leverages multitemporal data by combining vegetation indices and edge detection (Yan and Roy 2014). These studies generally use manually selected features or parameters, which requires a priori knowledge of the scale, physical appearance or distribution of the fields in the scene. No training data is required. Many studies additionally employ region growing as well as post-processing techniques to refine the field detection process. García-Pedrero et al. (2017) use a machine learning approach to iteratively merge selected adjacent superpixels, with the merge criteria determined by a supervised Random Forest classifier via various spectral indices and texture features. The methodology requires training data, manual feature selection and potentially further post-processing for the completed delineation of independent field parcels. The input data also needs to be manipulated such that the feature being targeted for segmentation needs to be present in the input data, orelse these traditional techniques fail to perform the task. Traditional approaches also lack the ability to learn new pattern and feature maps that are not explicitly programmed into the detection system.

1.c Deep learning approach

Deep learning approach involves use of neural networks,(non linear statistical models) generally with multiple layers. This approach involves building up a hierarchical structure of more and more abstract representation of input data. Neural networks are then able to make sense of complex datasets using this hierarchical structure and use that knowledge to make predictions about similar data. The key factor for the success of deep neural networks is the increased computation power of computers due to GPUs and massive increase in the amount of training available training data. The neural networks used for

computer vision tasks are generally convolutional neural networks (CNNs). CNNs rely on feature hierarchies and a layered architecture to perform computer vision tasks like image recognition, object detection and image segmentation. CNNs are able to use raw image pixel data and do not require manual feature engineering. Supervised training on large, annotated image datasets automatically tunes the CNNs to recognize and process relevant image features. In this way, the CNN maps the input image pixels to abstract feature representations and eventually to class probabilities. It is a general practice to have pooling, normalization and activation functions sandwiched between the neural network layers. This is done to improve the accuracy of the deep learning model.

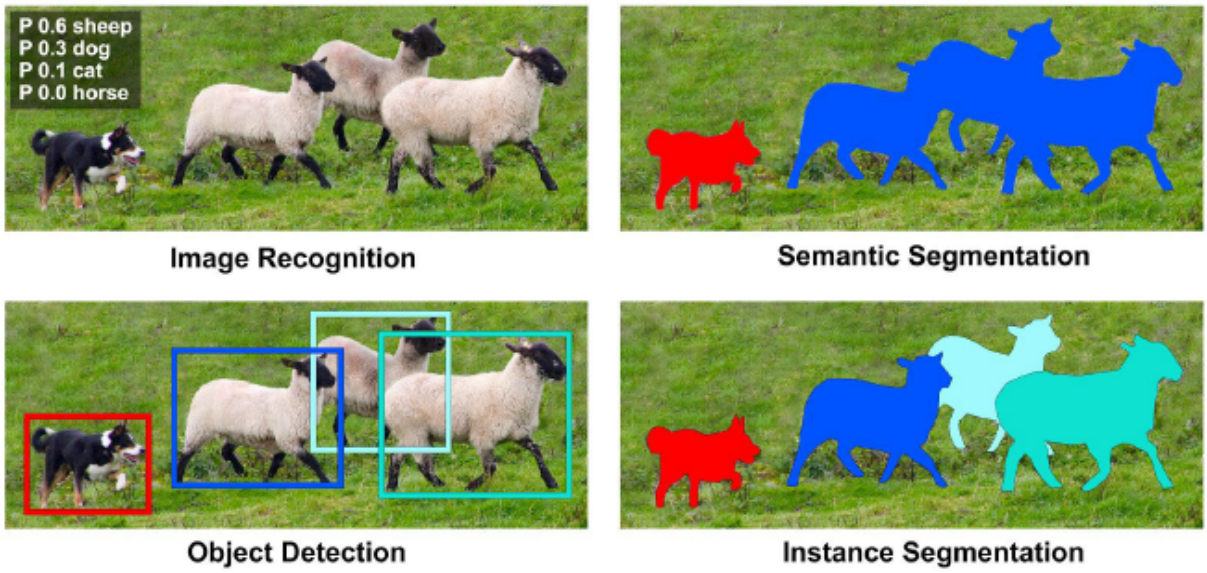

Figure 2: Image representing the difference between general computer vision tasks

There are 4 major tasks in computer vision which are image recognition , object detection , semantic segmentation and instance segmentation, as illustrated in figure 2. While using computer vision in agricultural practices , the appropriate computer vision task is instance segmentation because agricultural field objects are often directly adjacent to each other and have touching boundaries. In this case,

semantic segmentation would yield “clumped” polygon objects. Instance segmentation is able to distinguish between these connected or overlapping instances of the same class.

1.d Aim

The aim of this project is to show the potential, advantages and challenges of deep learning based instance segmentation models for automating the delineation and classification of crops from medium resolution farm images. This project uses the latest fully convolutional neural network architecture(YOLOv8) adapted from (Joseph Redmon et al. 2015). This project uses an apple farm dataset containing a total of 483 images and their annotations in json format.

2. Deep Learning for instance segmentation.

The deep learning algorithm that has been used in this project is YOLOv8 which stands for You Only Look Once version 8. This algorithm was launched in Jan 2023 and is the newest algorithm that was initially introduced to perform computer vision tasks like object detection and later improved to perform image segmentation. The following sections explain the working of the YOLOv8 model. This algorithm uses 4 basic mathematical operations which are 2d convolutional function , 2d max pooling function, 2d Batch normalization function and sigmoid linear unit (SiLU) activation function.

2.a Neural Network basics

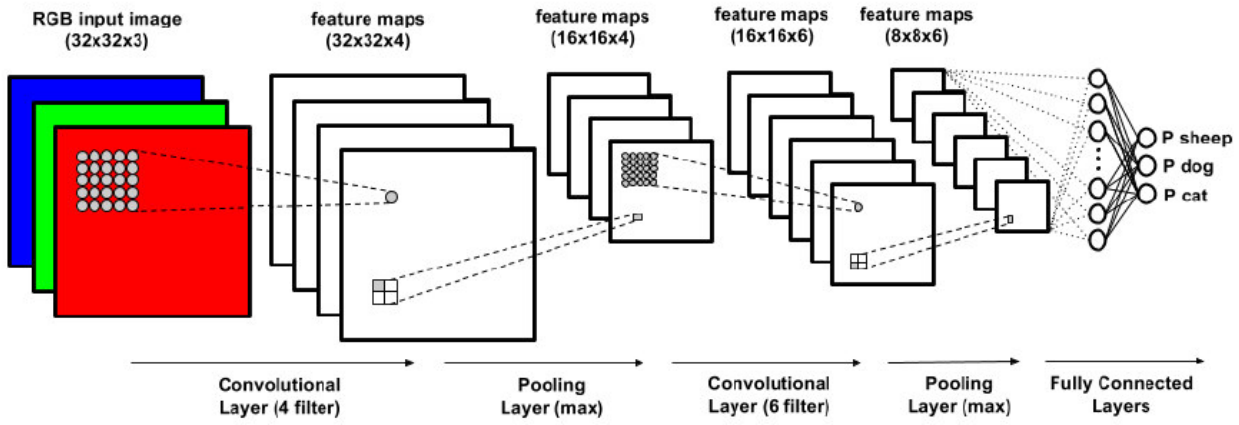

Figure 3: example of a CNN

Convolution function: A convolution function is basically used for reducing the dimension of an input tensor while simultaneously extracting a feature map or increasing the dimension of input tensor by adding padding.

Input: The input to a convolutional layer is typically a 3D tensor representing an image or a feature map from the previous layer. The dimensions of this tensor are usually height, width, and depth (number of channels). It involves a sliding window (kernel) that moves across the input, performing element-wise multiplications and aggregations. This process helps neural networks extract meaningful features from the input data.

Output: The output of a convolutional layer is a set of feature maps. The depth of the feature maps is determined by the number of filters in the layer.

Hyperparameters: convolutional function has 3 hyper parameters namely , filter size, stride, padding and number of filters.

- Filter Size (Kernel Size): A convolutional layer consists of a set of learnable filters (also called kernels) that are smaller in spatial dimension than the input. These filters slide over the input to perform the convolution operation. The size of the filters determines the spatial extent of the convolution operation. Common filter sizes are 3x3, 5x5, etc.

- Stride: Stride defines the step size at which the filter moves across the input. A larger stride reduces the spatial dimensions of the output.

- Padding: Padding involves adding extra pixels around the input to avoid shrinking the spatial dimensions too quickly. Common padding values are 'valid' (no padding) or 'same' (zero-padding to keep the spatial dimensions the same).

- Number of Filters: This hyperparameter determines the depth of the output volume, i.e., the number of filters applied to the input.

Effect of Hyperparameters:
Smaller filter sizes allow the network to capture fine-grained features, while larger filters capture more global patterns. However, smaller filters may require more layers to capture complex relationships. Larger strides reduce the spatial dimensions of the output, potentially discarding some information. Smaller strides may lead to larger output volumes and more computation. Padding helps preserve spatial information at the borders of the input. Without padding, the spatial dimensions can shrink rapidly, leading to loss of information. More filters enable the network to learn more complex patterns and features. However, this increases the computational cost.
.

$$y = \sigma(\sum_i x_i w_i + b)$$

Processing inside a convolutional function is described using the above formula where x is input, w is weight and b is bias. The goal of the training activity is to manipulate the values of weights and bias such that the loss is minimized.

Batch normalization : normalizes the input/output of each layer by subtracting the batch mean and dividing by the batch standard deviation. For a 2D Batch Normalization layer in a CNN, normalization is applied independently to each channel (feature map) along the batch dimension. This is done to stabilize the training , improve the learning rates and reduce the effect of abnormally high output of a layer on the entire result.

Given an input tensor X with dimensions(N,C,H,W), where:

N is the batch size.
C is the number of channels
H is the height of the feature map
W is the width of the feature map
the normalized output Y is calculated as follows:

$$Y = \frac{(X-\mu)}{\sqrt{\sigma^2+\epsilon}} \cdot \gamma + \beta$$

where:
μ is the mean of the batch across all spatial locations for each channel,
σ ^2 is the variance of the batch across all spatial locations for each channel,
ϵ is a small constant added for numerical stability,
γ is a learnable scale parameter, and
β is a learnable shift parameter.

The γ and β parameters are learnable during training, allowing the network to adapt the normalized output. This means the network can learn the optimal scaling and shifting for each channel.

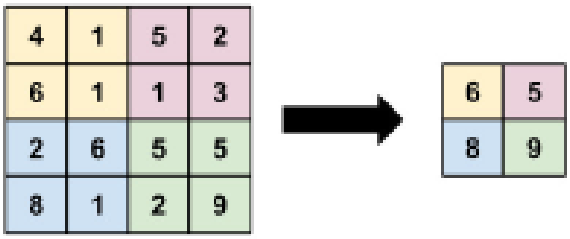

Figure 4: working of Max Pooling function

Max Pooling : Before a feature map is processed by the next convolutional layer, it is usually passed through a non-linear activation function and downsampled via a pooling layer. The pooling layer used in YOLOv8 is max pooling which is used to downsample a feature

map. A max pooling layer of size 2*2 with stride 2 will look like the above image and only the maximum value from each of the 2*2 squares from the feature map are stored.

Sigmoid linear unit (SiLU) activation function : The SiLU activation function is defined as follows:

$$\text{SiLU}(x)=x \cdot \sigma(x),$$

Where σ(x) is the sigmoid function.

In simpler terms, the SiLU function applies the sigmoid function to the input x and then multiplies it element-wise by x. The function has properties that make it differentiable, which is crucial for training neural networks using gradient-based optimization algorithms.The SiLU activation function has been found to perform well in deep neural networks and has been used as an alternative to other activation functions like ReLU (Rectified Linear Unit). Some researchers have reported that networks using SiLU can achieve improved training convergence and performance compared to networks using other activation functions.

Compared to ReLUs, SELUs cannot die. SELUs learn faster and better than other activation functions without needing further procession. Moreover, other activation functions combined with batch normalization cannot compete with SELUs.

2.b YOLOv8 architecture

YOLOv8 is a state-of-the-art object detection model known for its efficiency and accuracy. Its architecture can be divided into three main parts:

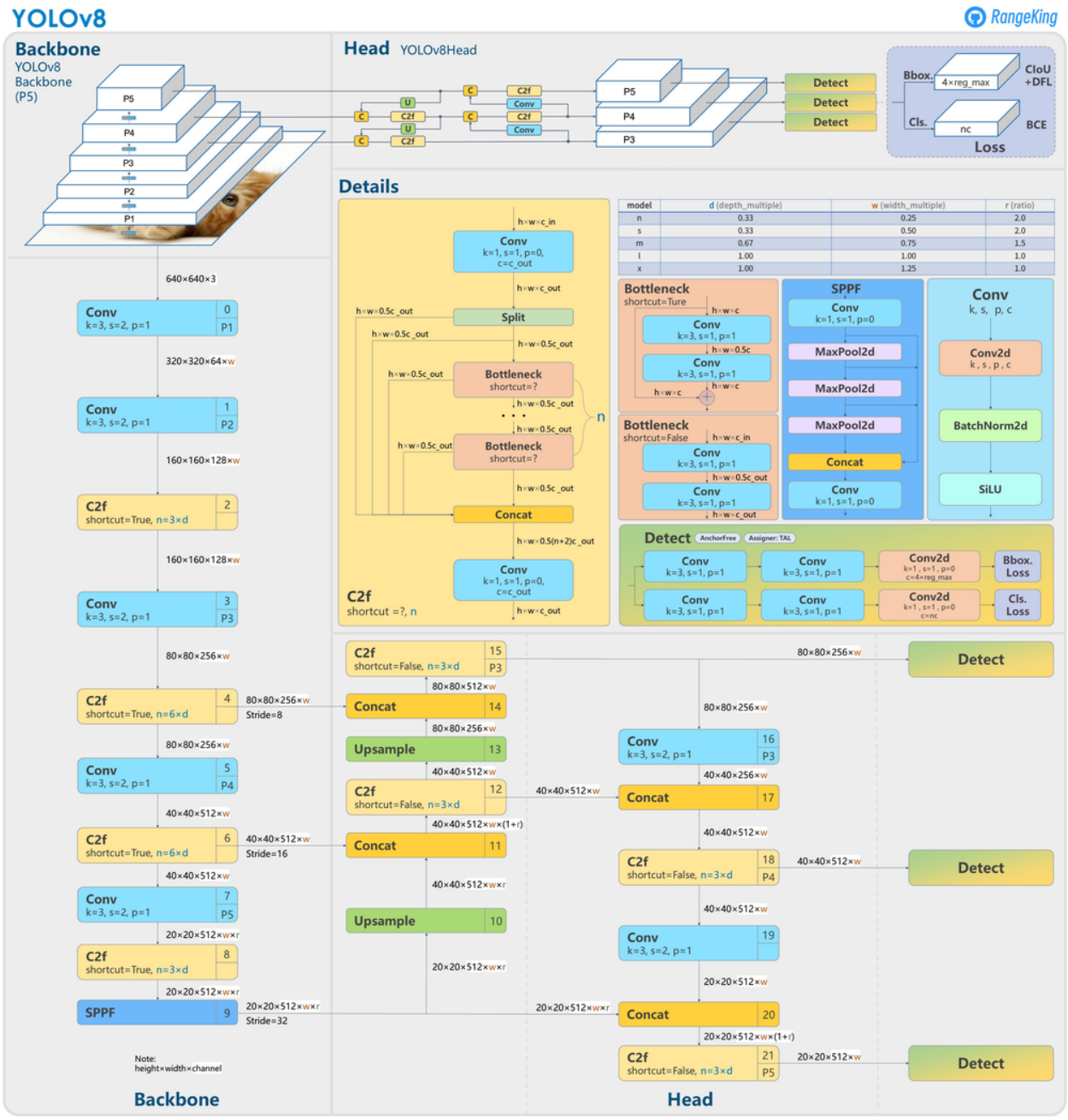

Figure 5: YOLOv8 architecture

1. Backbone:

This part focuses on feature extraction from the input image.

YOLOv8. This part is used to down sample an input image of size 640*640 and extract features at various downsampled dimensions. It comprises 53 convolutional layers and uses cross-stage partial connections to improve information flow and gradient propagation.

This structure efficiently extracts both high-level semantic features and low-level details crucial for object detection.

2. Neck:
This part integrates features extracted from different levels of the backbone. Although the neck is not explicitly mentioned in the code of YOLOv8, it is referred to as such to make explanation easy. Unlike traditional YOLO models, YOLOv8 introduces a novel C2f module instead of the neck architecture. This module uses a series of convolutions and upsampling operations to progressively increase the feature resolution while preserving semantic information. This allows the model to effectively predict objects at different scales in the image.

3. Head:
This part is responsible for making predictions based on the processed features. YOLOv8 uses a single head for both object detection and instance segmentation. The head predicts bounding boxes, class probabilities, and (in case of segmentation) mask coefficients for each object in the image.

YOLOv8 consists of 3 special modules called C2f, bottleneck and SPPF. C2f module stands for concat 2 feature, this approach is similar to that followed in resnets and is used for signal boosting and maintaining the information. Bottleneck module deals with balancing information flow and model efficiency. This is done by reducing the computational cost. SPPF stands for spatial pyramind pooling faster. It enhances the model's ability to capture multi-scale spatial information. It employs multiple max pooling layers with different kernel sizes in parallel and combines the outputs of these max pooling layers to capture context information of varying sizes. This context information is beneficial for object detection, especially for small objects.

3. Methodology:

In this image segmentation project, the initial step involved the acquisition of an annotated apple farm dataset that included both images and corresponding segmentation masks. The dataset was then augmented through the rotation of images by 90 degrees in both clockwise and anticlockwise directions, enhancing its variability. Additional augmentation techniques could be applied based on the characteristics of the dataset. Subsequently, all images were uniformly resized to 640 x 640 pixels, and pixel values were normalized to ensure consistency across the dataset.

The dataset was further preprocessed by iterating through its entirety and distributing the images into three distinct folders: train, test, and validate. The training set comprised 431 images, the testing set contained 20 images, and the validation set consisted of 35 images. These subsets were designed to facilitate the training and evaluation phases effectively.

The choice of model for this project was YOLOv8, renowned for its efficiency in object detection and segmentation tasks. Pre-trained weights from the Ultralytics package were utilized to initialize the model, leveraging transfer learning to adapt the model to the specific characteristics of the apple farm dataset. The subsequent training phase was conducted over 10 epochs, with the entire process executed on the Google Colab GPU runtime environment to exploit its computational advantages.

During training, metrics such as loss, accuracy, and Intersection over Union (IoU) were monitored to gauge the model's performance. The runtime environment of Google Colab GPU was chosen to expedite the training process, capitalizing on the available GPU resources.

Following training, the model's effectiveness was assessed on the test set, with metrics including precision, recall, F1 score, and IoU used for a comprehensive evaluation of segmentation accuracy.

The project methodology emphasized an iterative approach, with the possibility of fine-tuning the model based on evaluation results. The analysis of results involved visualizing segmented outputs on sample images and identifying potential challenges or areas for improvement. The entire process, including dataset details, preprocessing steps, model selection, and training parameters, was meticulously documented.

Evaluation metrics: the evaluation metrics used to analyze the performance of YOLOv8 are mask precision, recall, mean average precision and mean average precision.

- Precision : This refers to the percentage of detections that are actually true positives (correctly identified objects). A higher P value indicates fewer false positives.
- Recall : This denotes the percentage of true positives that were correctly detected. A higher R value indicates fewer false negatives.
- mAP50: Mean Average Precision at an Intersection over Union (IoU)(figure 6) threshold of 0.5. This calculates the average precision across all object classes at an IoU threshold of 0.5. IoU measures the overlap between the predicted and ground truth masks. A higher mAP50 signifies better overall detection accuracy at that specific threshold.
- mAP50-95: Mean Average Precision across IoU thresholds ranging from 0.5 to 0.95. This provides a more comprehensive evaluation of detection performance across a wider range of IoU values. A higher mAP50-95 indicates better object detection

accuracy across varied degrees of overlap between predicted and ground truth masks.

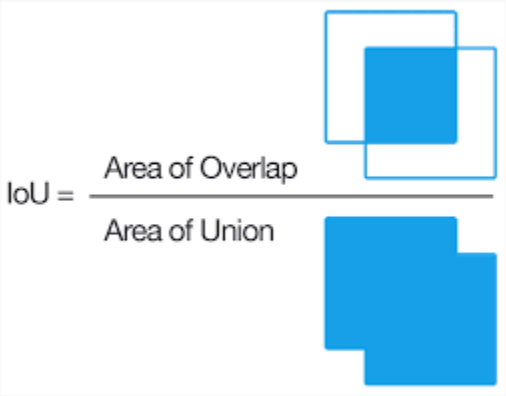

Figure 6: IoU explained

4. Results

Figure 7: bounding box and segmentation loss curves for training and and validation image sets

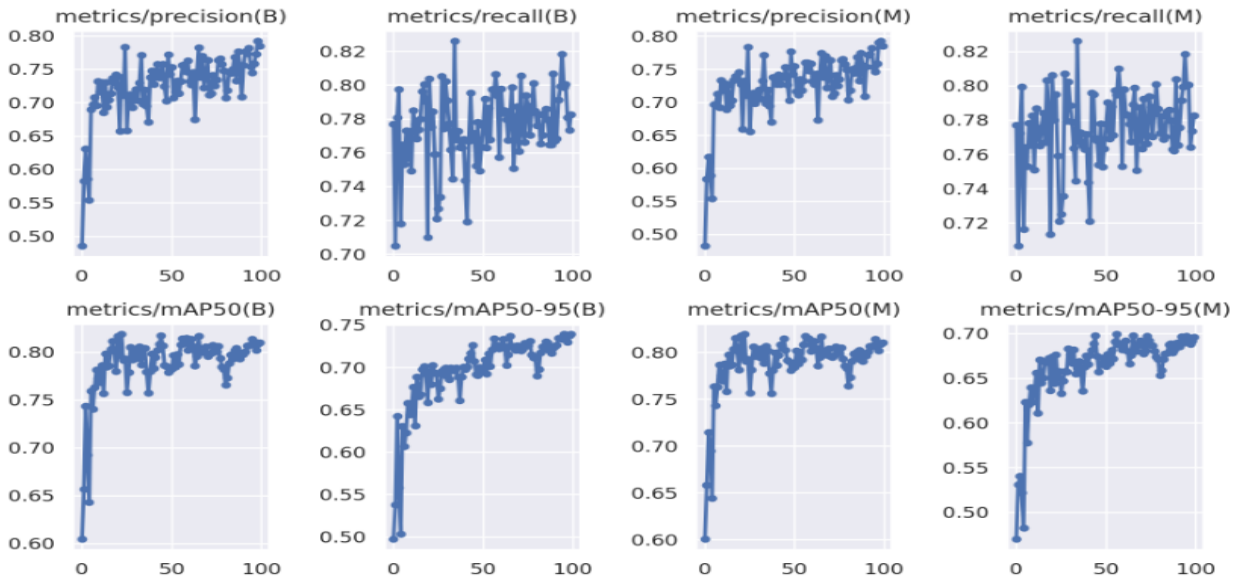

Figure 8: Precision, Recall, mAP50 and mAP50-95 curves on validation image set.

In figure 7, it can be observed that as the epoch number is increasing the bounding box and segmentation loss is decreasing. In figure 8, as the epoch number is increasing the mAP50 and mAP50-95 curves also have an general upwards trend. Thus giving promising signs that

the approach followed in the project is actually working and is able to segment regions of interests into proper classes.

Since the annotated dataset does not have a background class, identification of background is not possible. This can be verified through the confusion matrix present in Figure 9. Apart from this 82% of apples from validation dataset belonging to NOA class (Not Ok Apples) have been correctly segmented whereas 70% of apples belonging to OA class (Ok Apples) have been correctly segmented. The overall mean average precision of the model came out to be 76% which is very promising considering the dynamic nature in which the pictures were clicked.

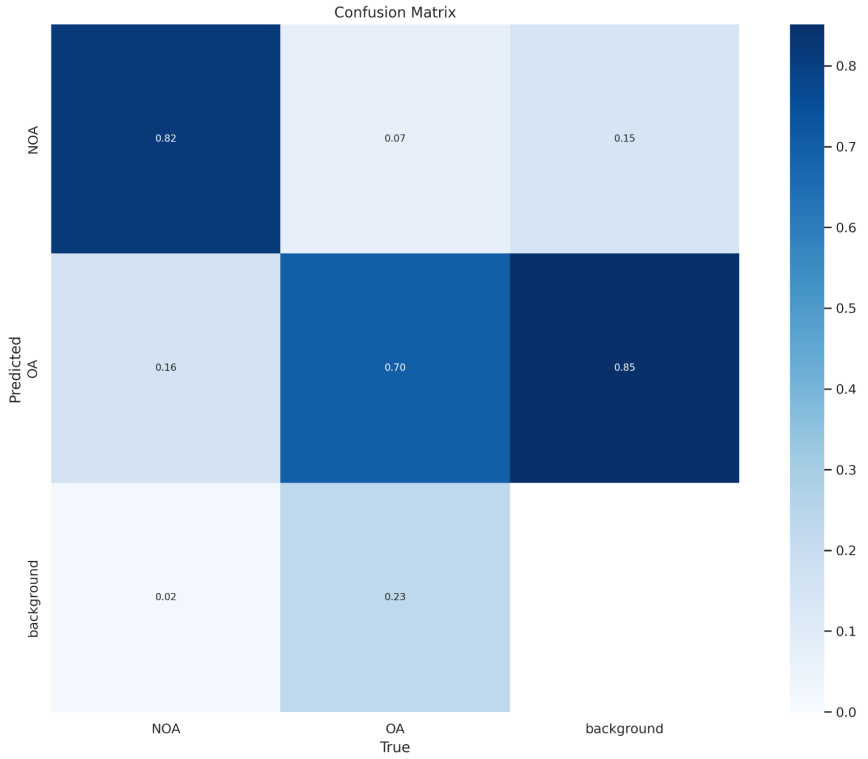

Figure 9: Confusion Matrix

Figure 10: raw image(left) and segmented image(right)

In figure 10 it is clearly visible that apples with any kind of defects are getting segmented into purple class whereas healthy apples are being segmented into red class. The model needs to have high precision and high recall inorder to have better accuracy. This can be done by tuning the hyperparameters. The total time taken for training was about 2 hours on google colab environment with a nvidia T4 GPU. The average time taken by the model to predict the mask is close to 60 msecs.

Future scopes through which the model performance can be improved are:

- Better annotated and accurate dataset: exact and correct masks are crucial for better model performance. In the current dataset, there are various instances where a leaf is covering the apple and it has been considered in the apple mask itself. This is a contributing factor for wrong segmentation.
- Use of better models: At the time of writing this report newer transformer based models, known as vision transfor have been

proposed. Use of these models for segmentation should be explored .

- Extending dataset preprocessing: augmenting the dataset to create a diverse database could result in better results but care should be taken care because augmentation often leads to changed masks and thus images are required to be annotated again.

Figure 11: raw images (left) and segmented images (right)

5. References

Butenuth, M. & Straub, M. & Heipke, C. (2004): Automatic extraction of field boundaries from aerial imagery KDNet Symposium on Knowledge-Based Services for the Public Sector. P3-4. 2004.

Rydberg, A. & Borgefors, G. (2001): Integrated method for boundary delineation of agricultural fields in multispectral satellite images. IEEE Transactions: Geoscience and Remote Sensing 39 2514–20.

Da Costa, J. & Michelet, F. & Germain, C. & Lavialle, O. & Grenier, G. (2007): Delineation of Vine Parcels by Segmentation of High Resolution Remote Sensed Images. Precision Agriculture 8 (1 –2): 95 – 110. https://doi.org/10.1007/s11119-007-9031-3

Torre, M. & Radeva, P. (2000): Agricultural Field Extraction from Aerial Images Using a Region Competition Algorithm, International Archives of Photogrammetry and Remote Sensing, Amsterdam, Vol. XXXIII, No. B2, pp. 889-896.

Yan, L. & Roy, D. (2014): Automated crop field extraction from multi-temporal Web Enabled Landsat Data. Remote Sensing of Environment 14442–64.

Redmon, Joseph, Santosh Kumar Divvala, Ross B. Girshick and Ali Farhadi. “You Only Look Once: Unified, Real-Time Object Detection.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 779-788.