FINDING A NEEDLE IN A HAYSTACK: GERMINATING WEED DETECTION

Master of Machine Learning Research Project
Project Report

Name: Nimish Pathak

ID: a1807182

Supervisor: Mr. Anthony Manchin

Course: Masters of Machine Learning

Introduction

The aim of my project is detecting germinating weeds in an agricultural farm consisting of various crops like onions, carrots, potatoes lettuces etc. Weeds are the plants that compete with the yielding crop for sunlight, water and nutrients and even space. Hence it can be very tedious for the yielding crop to reach maturation in an environment infested with weeds. Weeds also tend to compete with the harvest for nutrients and hence causing low agrarian yield and ultimately decreasing the land efficacy this leads to even more loss of crops compared to pests and diseases.

It can be said that if these weeds can be detected at an early stage such as during germination it would have tremendous benefits for the farmer in terms of weed management cost, yield production, soil potency etc. Due to the versatile nature of weeds even if one weed is present in the agricultural field it has the capacity to proliferate and spread rapidly affecting neighboring plants as well thus, if a weed is detected and removed during germination this can be stopped.

It is also observed that weeds are responsible for one third of the total losses in the yield, thus if these weed plants are detected at an early stage such as during the “germinating stage” many consequences that may majorly affect the yielding crop later on can be eliminated beforehand. Despite the advanced development and adoption of new weed detection and management techniques, the “weed problem” is virtually increasing thus, the germinating weeds have to be effectively detected and countered in order to stop this weed spreading phenomena.

Since the weeds have a tendency to multiply quickly without much aid either through active agents such as humans and animals or passive agents such as wind, water it is necessary to have suitable counter measures to stop this spread early on. Further upon maturation weeds become robust in nature and stronger herbicides have to be used leading to a decrease in soil potency. The resilient nature of the weeds is further enhanced due to mulching, intercropping, crop rotations and acquirement of fixed cropping methods within the yielding crop. Hence due to the dynamic nature of weeds continuous monitoring and continued refinement of weed management techniques is required.

Many health problems and environmental pollution is attributed to the over use of herbicides in a farm that may affect the yielding crop as well hence, if weeds are detected at a germinating stage this overuse of herbicides can be significantly reduced, hence saving the farmer considerable amount of wealth. The traditional methods of weed elimination in a farm include spraying the herbicide extensively over the entire area of the farm without evaluating the essential differences as to how these chemicals may affect the crops as well which may lead to human deformities too. In recent times “Precision Agriculture” has proven to be an innovative solution to this problem which involves site-specific spraying of the chemical, such a technique has huge potential and reduces the usage of chemicals by about 40-60%. According to Australian Bureau of Statistics about $35-$40 billion dollars is spent annually on agriculture in Australia and about $1.6-$4 billion dollars is spent on weed management. Hence, it also benefits the Australian Government if a more strategic and innovative methods are used to counter weeds in a farm leading to more economic gains (Alameh, 2014).

Therefore, in order to minimize this cost “Germinating Weed Detection” can prove to be an innovative and effective solution. This could enable farmers to approach the “weed problem” in a more strategic way such as use of a real time optical sensor fitted on an autonomous bot could be used to considerably reduce human induced labor and would also save the farmer plenty of time. Thus, leading to less expenditure on chemicals. Further, in case of germinating weed detection a naïve farmer may face difficulty in removing such small weeds as they may be indifferent from germinating crops. But due to recent advancement in machine learning algorithms, a computer vision object detection algorithm can be effectively trained to distinguish a germinating weed from a crop and this information could be used to pluck out the small weeds in the farm without affecting the crop.

Literature Review

This is particularly a small object detection problem because the germinating weeds are typically in the range of 10x10 pixels (2-5cm) in size and there is lack of distinctive shape and texture as well. Hence due to the small size of the weeds present in the farm along with the yielding crop it becomes difficult for a mundane farmer to be able to detect germinating weeds and effectively remove them from the field. Further, due to the disrupting nature of the weeds it becomes important to be able to detect and remove such weeds at the time of germination as upon maturation these weeds have a tendency to vary the soil composition and affect the quality of the yielding crop as well. The soil biochemistry is altered which leads to lowering of soil productivity in terms of number of crops that can be harvested (Vladan Stojni´, 2020). Thus, the farmer suffers economically and his profits are also lowered. For example, in case of onion crops, broadleaf weeds grow much quicker in comparison to the onion crops and then aggressively competes with the growing crop. This leads to weeds disrupting the soil biochemistry and makes the growing crop weak in nature ultimately leading to lower harvest and decrease in soil fertility. Hence, I came to conclusion that the yielding crop not only depends on the quality of the seed but also on the quality of soil around it and the presence of weeds had a tendency to lower this soil quality.

Splitting the dataset:

It is necessary to split the data set into training, testing and validation set. It is a method that in turn restricts the machine learning models from generalizing the dataset too well. If the model generalizes the dataset the output obtained will be biased in nature and we won't be able to get correct prediction i.e., detect germinating weeds. A loss function is used to judge how accurately the machine learning model is performing on our dataset i.e., the loss function measures the goodness or the badness of the model used. Thus, a machine learning model is said to have good performance if it has low value of loss for the validation and the test set on being trained with the training dataset provided. A higher value of loss indicates that the model may be overfitting the dataset (J.S., 2020).

A prediction function is generated by the model depending upon the loss function by mapping the image pixels in the input image to the output image. It is a general tendency of machine learning models to overfit on the training dataset i.e., an overly specific function is learned by the model such that its performance in the training data Is good but it doesn't perform well on the dataset it has never seen before which is the test dataset. In such a scenario the loss function on the training data shows a decreasing trend but, on the validation set it will gradually increase indicating the hyper-specific nature of the model. Here it can be said that the model is able to memorize the data present in the training set whereas we want the model to learn from the training set and then make predictions on the validation set. The training set comprises of the majority of the data that will be used to make the prediction. In my case it the germinating weeds dataset that will be used which I had gathered from various sources to predict small germinating weeds in an agricultural farm. This dataset comprises of the maximum number of images as it is used by the model to learn essential parameters of weeds. Hence, 70% of the total images are kept for this purpose. These are the images that will be used by the algorithm to understand the most relevant features and training of the ML algorithm will on this dataset (J.S., 2020).

20% of the entire data set is kept for the purpose of validation. This is the set of weed images that will be used to judge how good the model is able to perform on the images it has not seen before. A good measure that is used to judge the performance of the model is the validation loss and the validation accuracy which is the mAP (Mean Average Precision) on the validation set. Looking at these metrices gives a good sense as to how the model will perform on the test set which is the set of images the model has never seen before. If it is observed that the model begins to overfit the dataset, a technique known as early stopping is used. This is an optimization technique that is used to reduce overfitting without the compromission of the model accuracy. Early stopping is a regularization technique with which we can avoid overfitting when a model learns from a dataset iteratively like in case of gradient descent. Upto this point the models accuracy increases but beyond this point the performance shows a depreciating trend and increasing the models fit to the training data will come at a cost of increase in generalization error Thus with the help of early stopping this problem can be avoided by setting the number of iterations that the model can run before it begins to overfit (Mustafeez, 2017).

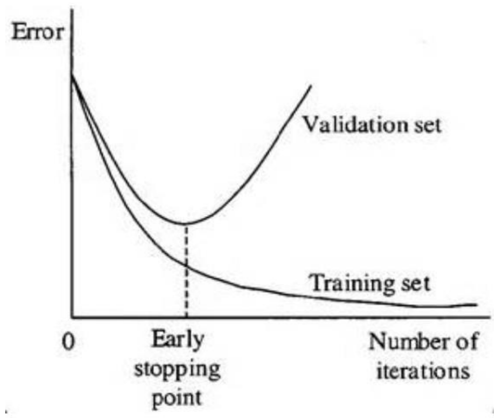


Fig – Early Stopping

Finally, the test set comprises of the remaining 10% of the data set these are the images of the weeds that the model has never seen before. It is crucial for the machine learning model to be able to give good results / be able to make good predictions on the test set as it would determine whether it will be effective in real life application or not. A good result on the test set will eventually provide with an unbiased evaluation of the final model fit on the training data. The test set is used only when the model has been completely trained.

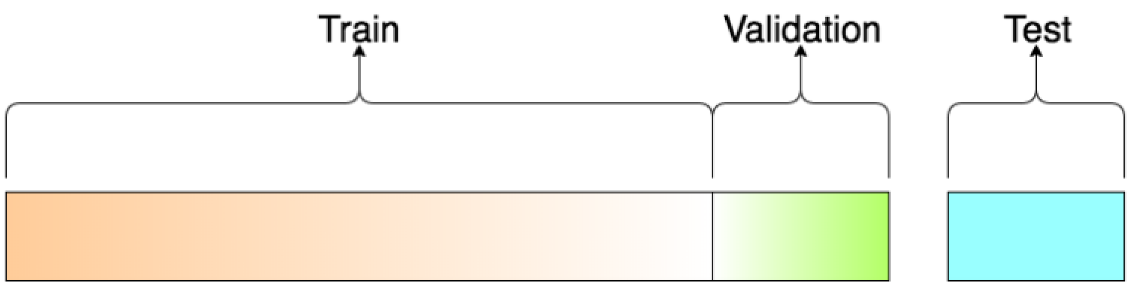


Fig – Splitting of the dataset.

A problem that I had encountered while segregating the dataset was of "train-test bleed", which arises when the training dataset is too similar to the validation/test set or due to the presence of duplicates among the data set. Hence, it was essential for me to remove the duplicate image form the dataset as without doing so the model was biased in nature and it was resulting in overfitting. Further earlier in my methods I had kept about 80% of the data in the training set. This had a negative consequence as the model was generalizing too well and insufficient number of images were present in the validation and the test set. Hence due to "too much emphasis" on the training set the model was overfitting. Hence, later I had segregated the images as 70-20-10. Which enabled me to get better results than before.

Preprocessing and Augmentation:

The main preprocessing techniques I had applied to my training, testing and validation set to ensure that the model is able to learn and generate predictions are Tiling, Anchor Boxes, Resizing.

Tiling:

If the size of the image is reduced it leads to serious lack of accuracy and details. Hence tiling is a technique in which the original image is splitted into tiles and the detection algorithm is run on each tile without serious loss of accuracy (Neskorozhenyi, 2019). Since the size of the germinating weeds is extremely small tiling proved to be useful. Prediction was made on many subtiles and the result was the aggregate of these predictions. I had made use of 3x3 tiling method where the original image is subdivided into 9 sub overlapping tiles. Tiling is done to ensure that the relative pixel area of small objects which is germinating weeds in this case increases with respect to the image that is fed in the network. As a consequence, the small objects i.e., weeds are effectively detected as image down sampling is considerably reduced. In the process of tiling each tile is treated independently and the resulting output is the bounding boxes with class probabilities. During this process there is presence of duplicates in the output in the initial results because of the overlapping between the tiles and the full frame of the image. Merging of initial results occurs in accordance to the intersection of the bounding boxes and the class scores. If this intersection of duplicates is above 25% then the tile with the higher score is considered as a better choice and the other one is deemed redundant and removed from the detection list. In the merging step the small objects i.e., germinating weeds get higher scores within the tiles as compared to the full frame of the image. This gives a boost to the accuracy score as now a more focused approach is taken by the algorithm and it can be correctly said that this method makes used of divide and conquer approach to be able to detect weeds in an image by first splitting the image running the detection algorithm on the sub images and finally aggregating the results. Further in the process of tiling the effect of down-sampling is reduced such that the small objects are effectively detected (F. Ozge ¨ Unel, 2019).

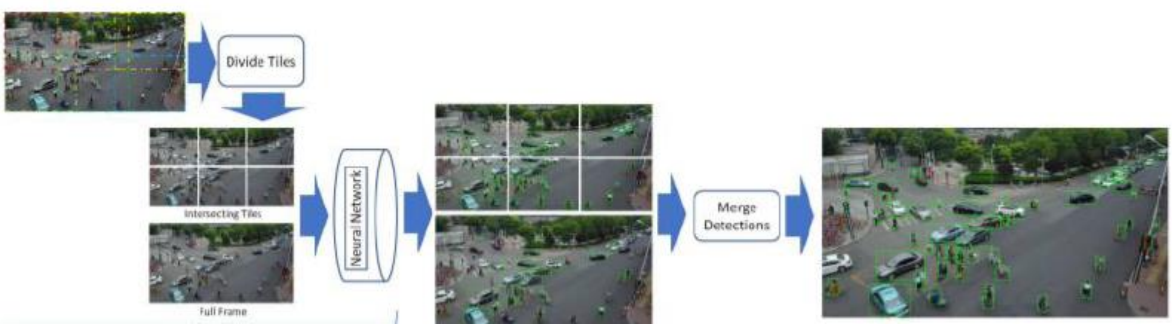


Fig- The Tiling process

Anchor Boxes:

The second preprocessing step that proved to be extremely helpful were the “anchor-boxes”. These are predefined bounding boxes with a fixed height and width. In case we want to capture the scale and aspect ratio of a particular object class anchor boxes are used. Anchor boxes are chosen depending upon the size of the objects in the training data. When placed across an image the anchor boxes are able to predict the probability that an image lies within the bounding box along with the IOU (Intersection over Union) score and the offset of each tile. With each prediction the anchor box is further refined and many anchor boxes of different sizes can be defined for a single object. In layman terms these are initial boundary box guesses for an object and are fixed in nature. The Deep neural network is able to refine these bounding boxes with each ongoing prediction to judge the class probability instead of directly predicting the bounding boxes. This is a key advantage of the Yolo family of models as the bounding boxes are automatically applied to the target images. Hence, for each anchor box a unique set of predictions is returned. Further, in the final feature map the detected object is represented with one bounding box with maximum class probability. The benefit of using anchor boxes is that objects of different scales, overlapping objects and multiple objects are effectively detected. Hence, all the object predictions can be evaluated at once without scanning the images using a sliding window that generates a prediction one at a time and is not time efficient. Thus, with the help of anchor boxes the entire image can be processed at once, making real time object detection possible. The most recent object detection algorithms such as Yolov5 have a predefined set of anchor boxes (Özgeünel, 2019).

The Yolov5 algorithm is able to generate a candidate pair by making use of the offset predicted from each anchor box. The loss function is then calculated from the ground truth images. Now if at a given time a certain anchor box overlaps with the ground truth, the probability of overlap is calculated. This is the IOU score, and if the IOU score is more than 50% the consequent loss is calculated by the algorithm. Finally, we get the resultant output as a detected image with a bounding box around the image with a confidence score around the box.

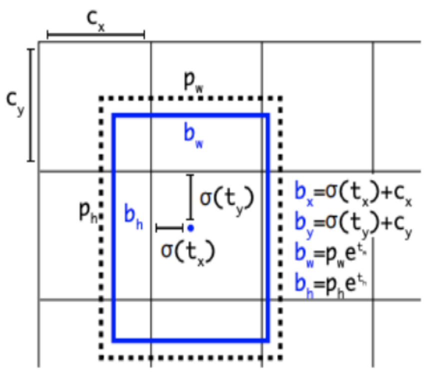


Fig – Anchor Box in Yolov5

Resizing:

The next preprocessing step used was to resize the image to a suitable size due to the small nature of the weeds present in the dataset. By resizing the input image to suitable size, the detection algorithm, Yolov5 in my case is able to run smoothly as the viewing field of the algorithm is increased. Further resizing a high-resolution image with small object to be detected can prove to be really effective and such was the case in detecting germinating weeds among onion crops in the farm.

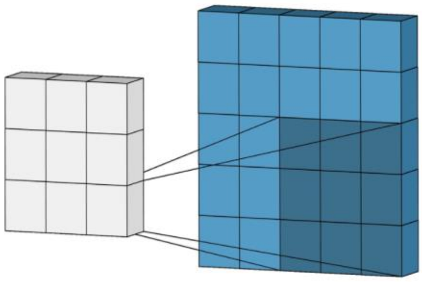


Fig - Resizing

Augmentations Steps:

Data augmentation is a technique through which the diversity of the data set can be increased that can be then fed to the object detection algorithm. It is a strategy through which more data can be generated without actually collecting new data thus in a way a synthetic dataset can be generated which is essentially larger than the original dataset that is basically used to increase the downstream performance of the model. Hence. in this why the model is better able to generalize to the situation that it may encounter during production. A few augmentation techniques that I had made use of after the preprocessing steps are grayscale, hue, saturation, flip, rotation (Solawetz, 2020).

Grayscale is a technique in which a single channel grayscale image is generated as the output for an input image. Grayscaling helps in dimensionality reduction and also reduces the model complexity.

Fig – Grayscaled Image

In case of Hue the input image color channel is altered, hence now more color channels are available for the model which it may encounter in real time environment. It is a way to know that the model is not memorizing the color channel in the input images. The model also becomes efficient in detecting the image edges and the shape of the object instead of just the color variations present in the input images (Solawetz, 2020).

Fig - Hue

The next augmentation step applied to the dataset was saturation which determines how vibrant the image is. It can be said that a fully desaturated image is a grayscale image. In my case since some color variation was present between the soil, the crop and the weeds saturation proved to be helpful for the detection algorithm to be able to successfully differentiate between the crops and the weeds present in the land.

I had also applied exposure the image which added variability to the image brightness that helped the model to be more resilient to the lighting and camera setting changes as it would encounter in the real-world scenario.

Fig-Exposure

Precision Agriculture:

Agriculture is still far off from completely deploying autonomous bots to be able to handle an entire farm, but certain tasks such as weed management is one are where robotic assistance can shine. Namely precision agriculture. Precision agriculture is a technique that used the recent advancement in technology such as machine learning and object detection using computer vision to precisely deploy the requisite amount od herbicide at the site of germinating weeds. A few applications of precision agriculture include variable-rate spraying., mapping weeds patches and crop yield management. In this way the chemical use on the agricultural field can be significantly reduced thus not disturbing the chemical ecology of the soil. Under precision agriculture three types of information is required namely information on seasonally stable conditions i.e., annual soil and yield based properties, information on seasonally variable condition that include weed infestation and the weather and most importantly the information required to diagnose, find the cause and develop effective strategies for the management of the growing crop. A computer-based imaging system has a potential to provide information for all the 3 categories at the fingertips of a farmer enabling him to take relevant counter measures way ahead of time. Such type of precision agriculture system involves an optical system at the helm of its use. For example, a monochrome, spectral, and color cameras can be used to gather the relevant information for classification of plant and the germination weeds. It can be said with the help of such a technology weed detection and prevention can begin ahead of planting the crops. Hence, the weed seeds and the small germinating weeds can be recognized and effectively removed, minimizing their growth (Shanmugam, 2018).

WeedAI:

WeedAI is a computer vision approach developed by the University of Sydney to tackle the weed detection and elimination problem and closely resembles to my stated problem i.e., germinating weed detection. In case of WeedAI a Site-Specific weed control (SSWC) method is used which is a fine-grained approach to tackle the weeds growing in a farm. Within this approach the land and the crops and not exposed to the treatment measures and only the weeds are targeted. In case of WeedAI novel weed prevention technology used comprising of lasers, microwaves and electrocution of weeds. The method is successful in accurately detecting and localization of weeds. Since there can be a close similarity between the crops and weeds the approach taken by WeedAI is urban in nature and uses the state-of-the-art classification, detection and segmentation algorithms to differentiate weeds from the crops (Sydney, 2019).

Ecorobotix:

The robot works without being controlled by a human operator. It covers the ground just by getting its bearings and positioning itself with the help of its camera, GPS RTK and sensors. Its system of vision enables it to follow crop rows, and to detect the presence and position of weeds in and between the rows. Two robotic arms then apply a micro dose of herbicide, systematically targeting the weeds that have been detected. Reliance on solar power makes the robot completely autonomous in terms of energy, even when the weather is overcast. As it adapts its speed to the concentration of weeds, it is most suitable for use in fields where the level of concentration is low to moderate, in order to cover the ground at a reasonable speed. We recommend using the machine after an initial standard application of herbicide, in order to replace subsequent applications and thus save an important amount of herbicide. The machine can be completely controlled and configured by means of a Smartphone app (Xiaolong Wu, 2020).

Models (Computer Vision Object Detection Models):

Now I present the information I had gather on the best ML models that can be used to solve the weed detection problem with high accuracy. For my project the accuracy is very important since it is recommended that each and every weed is effectively detected at the growing stage. If this doesn't happen the weeds can spread at an alarming rate hence corrupting the entire farm in a matter of weeks.

For precision agriculture recent advancement in Deep learning and convolutional neural networks have proven to be remarkably effective in terms of computer vision for weed detection. Major breakthroughs have been achieved in convolutional neural network application in weed detection, semantic segmentation and weed recognition. This can be strongly attributed to the high representational ability of image features with the help of convolutional neural networks. The previous model achieved precise detection of weeds by increasing the network depth and width hence complicating the model resulting in lower detection speeds. In a real word scenario real-time, accurate and resource saving detection has to achieved which can be provided by the Yolov5 algorithm. The reason I chose Yolov5 as the preferred model for making predictions is because of its fast detection and ability to obtain high accuracy (Li Liu, 2020).

The Yolov5 has 4 versions at its disposal and I had selected the "s" version. A few important aspects of the Yolov5 algorithm are as follows: The model architecture, the activation functions used, the optimization functions used, the loss function used and the weights, parameters, biases, gradients.

The Yolov5 which stands for "you-only-look-once version 5" detection algorithm is a single shot detector and comprises mainly of 3 parts namely the backbone, the neck and the head. The backbone of the model is used to extract the most relevant and important features from the input images. In case of the Yolov5 a CSP which stands for cross-stage -partial network is used as the backbone. The CSP backbone is used to extract all the rich and the most essential features that provide the most relevant information about the weeds present in the dataset. The benefit of the CSP backbone is that the inference time is lessened compared to the previous Yolo version which use the Darknet network as their backbone and the CSP network is much deeper in nature hence it is able to extract relevant features from the dataset with more precision (Li Liu, 2020).

The neck on the Yolov5 model is used for generating the feature pyramids. A feature pyramid is used to generalize the model by scaling it. Hence in case of the germinating weed to be detected the feature pyramid is able to recognize the weeds of different size and shapes present in the dataset. Further it will be essential when a never seen before image is encountered. Thus, it can be said that the feature pyramid is essential to be able to get good predictions on the images it has never seen before i.e., the test set. A PANnet neck is used in case of the Yolov5 model where PANet stands for path aggregation network (Shu Liu, 2018).

The head of the Yolov5 model is used to perform the final part of the detection task. Hence the head is used to apply the anchor boxes on the features and therefore generate the resultant output. Various factors are used to generate the final output such as class probabilities, bounding boxes and objectness score. The Head of the Yolov5 model is same as the head used in previous yolo versions.

The second model I had used to carry out the detection task is the Faster RCNN model. The Faster RCNN frame work can be used for both detection and classification task and it also has benefits in case of real-time object detection in terms of speed and accuracy. It works on Regional Proposal Network (RPN) which is based on convolutional neural network. It is observed that the time required to generate the region proposals with the RPN is very small hence it is fast in execution thus making it effective for getting quick results and detecting germinating weeds. When compared to conventional regional proposal algorithms such as selective search (SS) the RPN is able to achieve higher accuracy as it recommends fewer regions. Faster RCNN is still making progress in getting real time application as it still lags behind as getting region proposals before the prediction takes a lot of time.

Methodology and Experimental Results

Yolov5 on Lettuces

For the lettuces I was able to gather 312 images on which small germinating weeds were present. This dataset was provided to me by the Industrial partner Flux and contained images captured through a high-resolution camera in an agricultural field. In this case as well the germinating weeds had a size variation of 2-5cm in height so typically this was also a small object detection problem. Further in the images provided for the lettuce crops, the lettuces had already matured enough and could be easily distinguished from the germinating weeds as it can be seen in the images below.

Fig – Lettuce Crops with germinating weeds

The preprocessing steps most suited for the detection of germinating weeds among lettuce crops were Auto-orient, Resize and Tiling.

In case of auto orient the images are rotated by either 90 or 180 degrees for the detection algorithm to be able to predict germinating weeds in different orientations as would be the case in an actual farm. This would also enable the detection algorithm to judge and predict weeds present in various locations in the soil. The second necessary preprocessing step that was useful was the resizing of the images due to the miniscule nature of the germinating weeds. This basically increased the receptive field of the algorithm so that the Yolov5 algorithm was able to clearly learn about the small weeds and then make suitable predictions. The next preprocessing step that had helped massively was the tiling process in which the input image was split into tiles with 3 rows and 3 columns this was essential as now the detection algorithm would run on each tile individually and was able to produce food results.

The augmentation steps that were most helpful in making the data set appropriate for training were flip, rotate, bounding box rotate and brightness. These augmentations steps were done to make the model more susceptible to changes and how it would encounter the weeds among lettuce crops in the farm. Further after this the training process would be more efficient in nature. The Yolov5 model is able to generate a mosaic of augmented images due this new attribute of the yolov5 model a lot of memory is conserved and also a lot of GPU memory is saved as well. It can be seen below how the mosaic data loader works and the set of augmented images are collected together which will be then fed for the training purpose (Özgeünel, 2019).

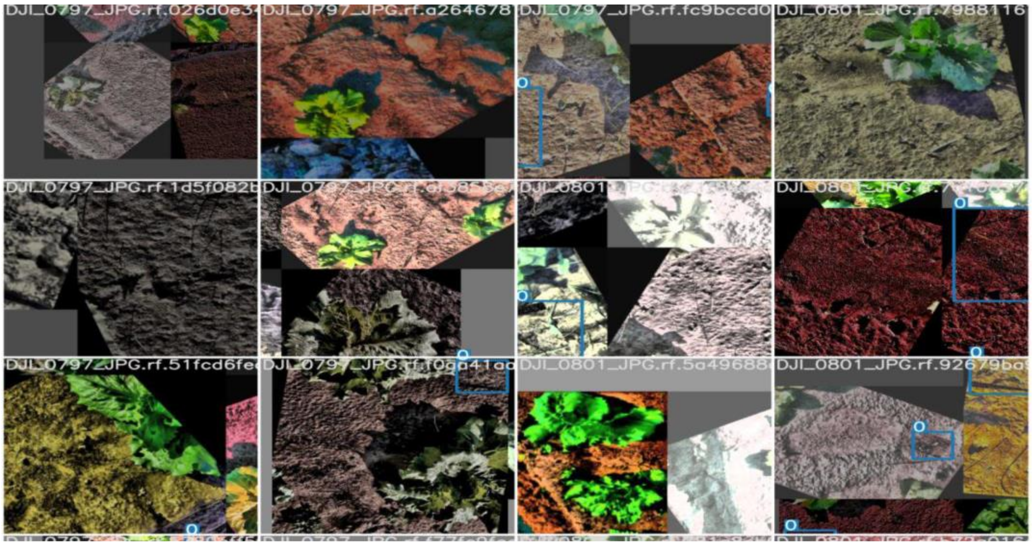


Fig – Augmented Images (Mosaic Data Loader)

After applying the preprocessing and consequent augmentation steps I was able to generate about 1000 images of which I had kept 700 images for training, 200 images in the validation set and rest of the 100 images for the purpose of testing.

For the purpose of annotating and labelling the images I had made use of the Roboflow software. I had made the bounding boxes around the germinating weeds as follows and classified crops as lettuces as well.

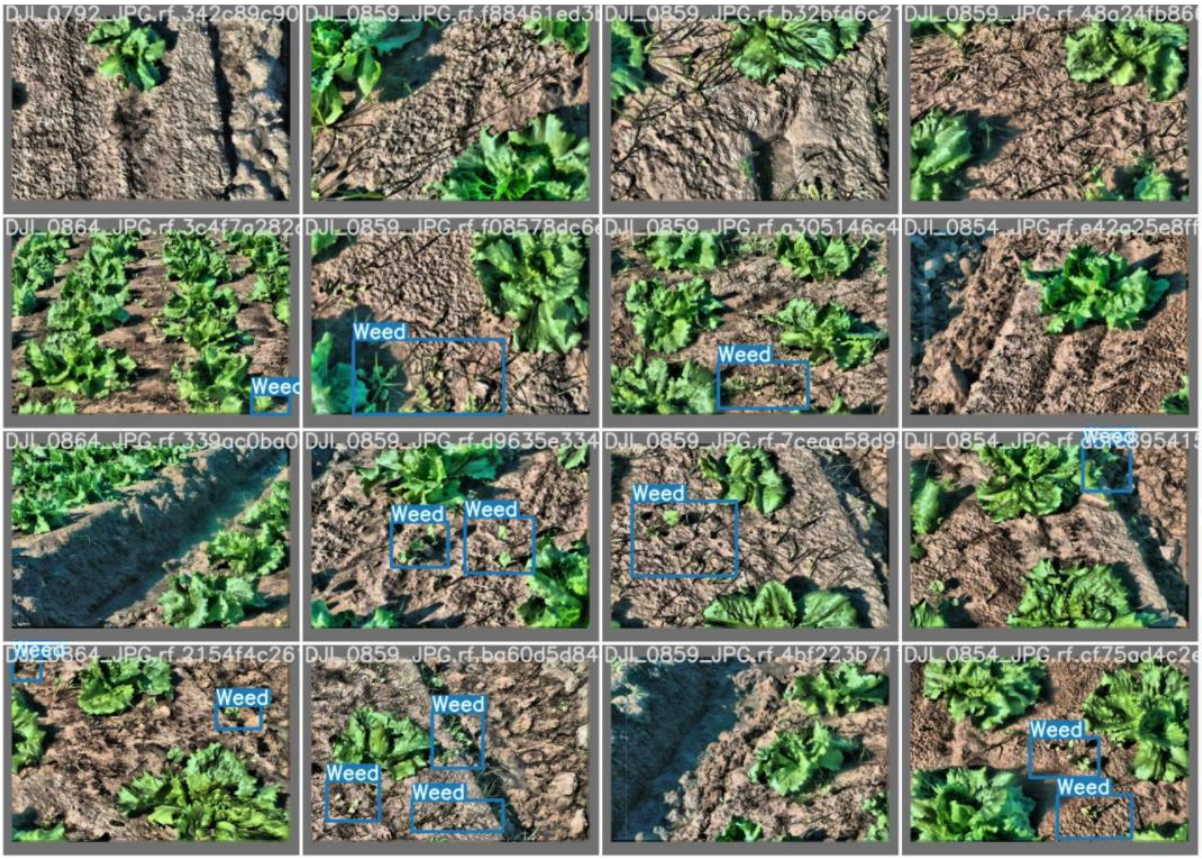


Fig – Annotated Ground Truth

In case of germinating weeds among lettuces I was able to get an mAP score of 52%. With a precision of 30.2% and recall of 64.6%.

51.7%	30.2%	64.6%
mAP	precision	recall


Weed 0.51


Weed 0.55


Weed 0.51


Weed 0.50

Yolov5 on custom dataset.

Secondly, after getting the predictions on lettuces I had created another separate dataset by gathering images from different public sources. These images had germinating weeds among different crops. I had made use of various preprocessing and augmentation steps for this dataset as well as discussed in the literature review and finally was able to generate 1500 high quality images for the purpose of training.

Fig – Gathered Images

For the custom dataset which comprised of 1500 images from public sources and the images of onions that was provided by the industrial partner FLUX. I had made use of the Yolov5 model because of its “small object detection “capabilities. Hence once the dataset was labelled using the AWS Sagemaker Groundtruth and Roboflow I was able to generate a high-quality dataset in the Yolo format.

Fig – Annotated Images

The next task associated was to keep the annotations and the images in the same directory. Once this was done, I had separated the dataset into train, test and the validation set. Hence, about 70% of the data was kept in the training set. This was the majority of the data set that will be used for the purpose of training the Yolov5 model. The validation set consists of 20% of data and the test set comprised of 10% of the data. Also, I had kept the images in an image directory and all the annotations in the txt directory. Now it was necessary to set up the environment hence a Pytorch version greater than 1.5 was required therefore I used the latest Pytorch version and CUDA version 10.2 for the detection task.

The rest of dependencies were installed using the pip command. After this step all the dependencies were successfully installed. Now the YAML files had to modified this was the file that contained the dataset parameters and it had to be configured according to the number of classes present in the dataset. Since the dataset mostly contained germinating weeds along with the mature crops of onion, potatoes, carrots etc. I had appropriately changed the YAML configurations that suited the needs of the model. Since for the purpose of detection most of the small broadleaf weeds had been labelled the number of classes were two. One class represented the germinating weeds and the second class was the class that contained all the crops. Further, the germinating broad-leaf weeds were quite distinctive from the crops, even in case of germinating crops these weeds could be easily distinguished from small and germinating crops because of the presence of broad leaves in case of the weeds. This "broad-leaf" aspect of the germinating weeds was the most important feature that separated them from rest of the crops in the dataset (Rajput, 2020).

The Yolov5 models had 4 variations s,m,l and x and I had made use of the “s” model for my purpose. As the “s” model had much less inference time compared to other models and thus resulted in faster computations and was the most suited one for real time application. The YAML file was passed to the Yolov5 “s” version and now the model was ready to begin the training.

A few important parameters of the Yolov5 model are as follows:

- Img: this indicated the size of the input image
- Batch: this indicated the batch size
- Epochs: number of the epochs the model is trained for
- Data: here the YAML file is passed with the number of classes and the corresponding class names.
- Cfg: This is the configuration file.
- Weights: the weights file is used for the purpose of transfer learning.
- Device: this is used to select the training device “0” in case of GPU and “cpu” in case the CPU is used.

Hence, I had trained my model for 100 epochs first and the resultant model was saved in the “weights “directory.

<u>The YOLOv5 model explained architecture:</u>

The computer vision object detection model Yolov5 is a single stage object detection model and basically has 3 parts of particular importance. First and fore most the Model backbone second the Model neck and thirdly the model Head.

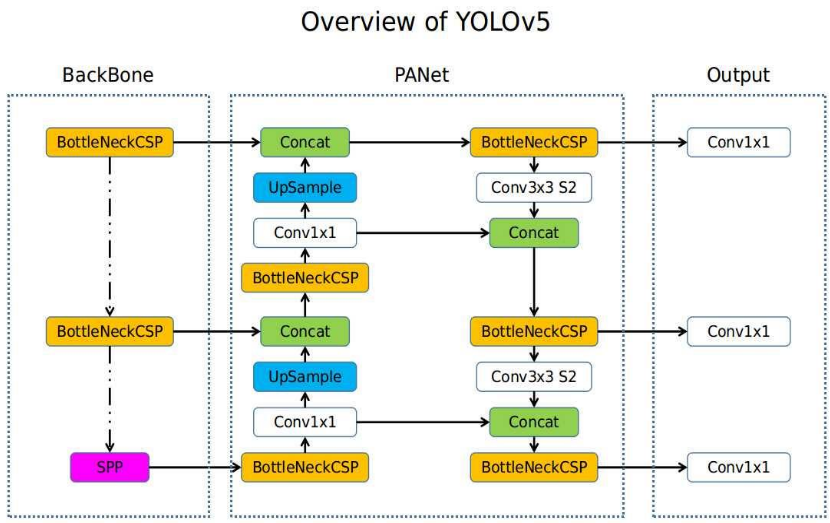


Fig – Yolov5 Architecture

The Yolov5 backbone was used to extract all the relevant and important features from my germinating weeds dataset. A CSP backbone is used in case of the Yolov5 model which stands for the Cross stage partial network. A CSP backbone basically extracts all the rich informative features about the germinating weeds from the dataset provided. Further, in recent times the CSP net has shown to have a significant advantage in terms of processing time due to the presence of deep neural nets. The model neck in the Yolov5 model is used to generate the feature pyramids which are used to generalize the target objects i.e., germinating weeds in my case. Hence, with the help of the neck same object with different sizes and scales can be easily identified by the model. Further, a good result on the training data will enable the feature pyramids present in the neck to give good performance on unseen test data. In the Yolov5 model a PANet neck is used for the purpose of generating the feature pyramids. The head of the Yolov5 model is basically used to perform the final part of the detection and it applies the anchor boxes a key feature of the Yolo models on the features generated and at the end a resultant output is generated with the vectors i.e., bounding boxes and their respective class probabilities. Objectness scores and confidence scores are also displayed with the resultant output. The head of the Yolov5 model is similar to head used in other Yolo models (Rajput, 2020).

The activation function I had used for the purpose of generating my outputs was Leaky RELU and Sigmoid activation function. It can be said that the choice of activation function for any deep neural network is of utmost importance. Hence, after testing various activation functions such as Mish, Swish, RELU, etc the Leaky RELU and Sigmoid activation functions were the most effective ones in generating correct predictions for my task. The Leaky RELU activation function is used in the middle or the hidden layers whereas the sigmoid activation function is used in the final detection layer.

The next task was to correctly choose the right optimization function and I went with the SGD or the stochastic gradient descent optimization function. The reason is that it is easier to fit this optimization function due to single training sample which is processed by the Yolov5 network. The SGD function is fast in computation because only one sample is processed at a time. Lastly due to frequent updates the steps taken towards achieving the local minima of the loss have oscillations which help in getting out of the local minimums of the loss function.

In the Yolov5 model the class probability score, objectness scores and the bounding box regression scores together give the compound loss and lower the value of the loss the more accurate the predictions are in nature. Hence in my case I had made use of the Binary cross entropy with logit loss function which was used to calculate the loss of the class probabilities and the object scores (Kapil, 2019).

A few benefits of the Yolov5 model that made me choose this model for making my predictions are as follows:

- High accuracy: Yolov5 is able to achieve good accuracy with minimum background results.
- Learning capabilities: Yolov5 is a highly tunable algorithm and has good learning capabilities through which it can learn the representations of the germinating weeds effectively and hence be able to generate good detection results.

In the working of the Yolov5 algorithm the input image is first divided in various grids. With each grid having a dimension of SxS. Each of the grid cells has equal dimensions and each of the grid cell has the capability to detect objects i.e., germinating weeds present within them. For example, if a small weed is present in a grid cell that cell will be responsible for detecting it (Section, 2021).

The bounding box is used to highlight the object to be detected in an image. This bounding box has the following attribute: Width, height, class, bounding box center.

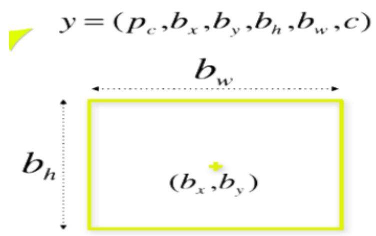


Fig – Bounding box/ Anchor attributes

Further in Yolov5 a single bounding box regression is used to predict the height, width, center and the class of the object. Hence, the name you-only-look-once. In the bounding box the probability whether the target object is present or not is represented.

The Yolov5 uses the IOU (Intersection Over Union) score which describes how the predicted bounding box overlaps with the ground truth. The yolo uses the IOU score to generate an output box that surrounds the target object perfectly and this will be the resultant output.

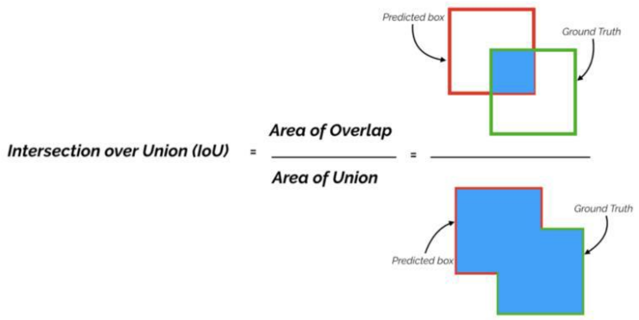


Fig – Intersection over Union

A grid structure is formed which is responsible for predicting the bounding boxes and the respective confidence scores. A value of IOU score 1 means that the bounding box completely overlaps with the ground truth bounding box. Hence, through this mechanism the bounding boxes that are not equal to the real box are eliminated as a consequence. Therefore, in the Yolov5 algorithm generation of residual blocks, bounding box regression and IOU is used to generate the resultant output.

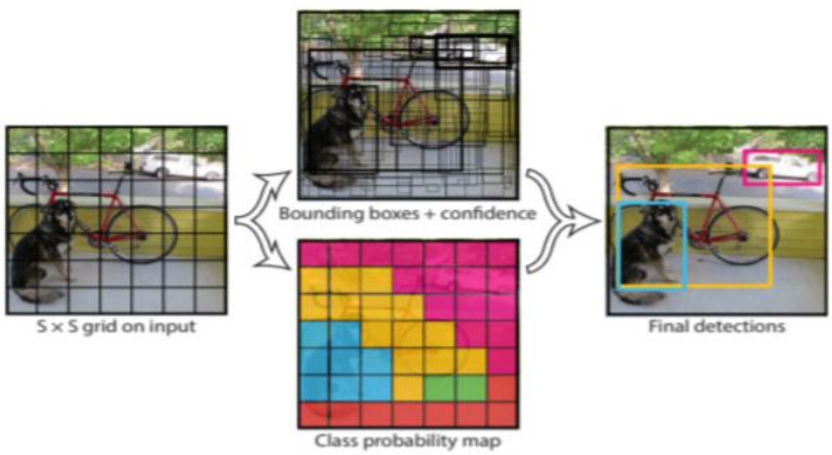


Fig – Formation of grid followed bounding boxes and hence the detection.

For germinating weed detection the image of the soil which contains both the weed and the crop is first split into grid cells. With each of grid cell, a bounding box B is formed which gives the resultant confidence score. And as a result, a class probability is predicted that enables the model to judge whether the object present in the image is a weed or crop. The IOU score gathered at the end of the predictions are used to make sure that the generated bounding-box by the algorithm is equal to the ground truth bounding-box and the unnecessary bounding-box that do not meet the characteristics of the object are eliminated.

The CSP Backbone:

In my model I have made use of the CSP (Cross Stage Partial) bottleneck as the backbone which is used to generate the image features. The benefit of the CSP is that it is able to address the duplicate gradient problem compared to the larger ConvNet backbone in other models. This benefits the model in having less parameters to deal with and therefore less FLOPS. This is essential for my model as it results in good inference speed and also supports the small nature of the model (Yolov5 version "s") making real time evaluation possible. The CSP backbone is particularly based on DenseNet (Solawetz, 2020). The DenseNets are designed to connect the layers in CNN with the following key points in mind:

(a) To alleviate the vanishing gradient problem as it is hard to backprob loss signals through a very deep network

(b) to enhance feature propagation, further encouraging the network to reuse the features and ultimately reduce the number of network parameters.

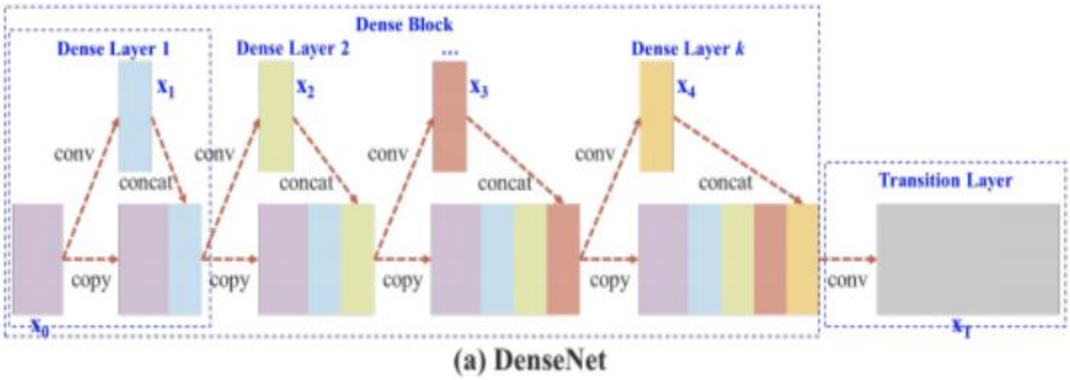


Fig - DenseNet

In the CSP bottlenect the DenseNet has been modified in order to separate the feature map generated of the germinating weeds in the base layer by copying it and then correspondingly sending one feature map through the dense block and sending another one to the next stage. The main idea behind this is to remove the additional computational bottlenecks in the DenseNet and hence, improve the learning by passing on an unedited version of the feature map.

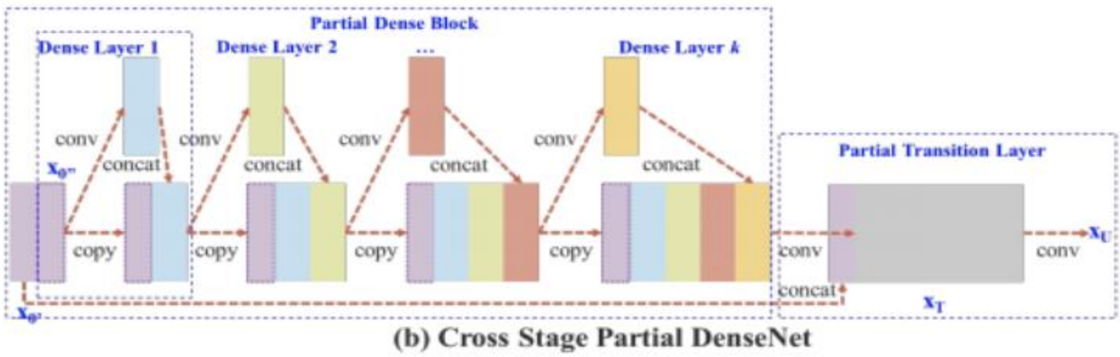


Fig – CSP Bottleneck Backbone in Yolov5

The PANet Neck : The Path aggregation network help in feature aggregation i.e. getting the most important and relevant features of the germinating weeds from the dataset provided.

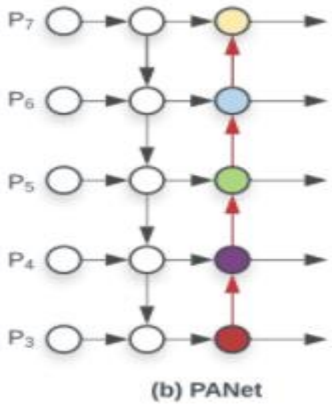


Fig – PANet Neck in Yolov5

The P(i) above represents the feature layer present in the CSP backbone.

Yolov5 has proven to be the best model for my inferences due to following capabilities:

- Easy Installation: Ease of installation due to installation of a few python libraries and Pytorch framework.
- Fast training: Due to fast training capability much of the inference time is considerably reduced and also lowers the experimentation cost as the model is built.
- Working Inference ports: Multiple versions can be used for inference such as video feeds, individual images, batch images etc.
- Small object detection problem: Most algorithms extract information by increasing the depth of the network such as VGG, this leads to high inference time. Inception models uses multi-branch structure to design and optimize the network and replaces large-size convolution with small-size convolution to reduce the network parameters. These are excellent for object detection but they do not have ideal performance in detection of small targets. But Yolov5 shines here and is able to handle the “small object” problem well. Yolov5 deals with problem by combining the high-layer network with low-level layer. This way the receptive field is enlarged and good results are obtained.
- Faster computation due to one stage detection: Feature extraction and getting region proposals then classification of region proposals by deep CNN layers are done in one step this improves speed and accuracy.
- Auto-anchors: Anchor boxes are automatically used, thus training on custom data set becomes much easier.
- A few other benefits of Yolov5 are
    - Ease of use/accessibility
    - Exportability
    - Memory Requirements
    - Speed of inference
    - High mAP score

- Easy to tune according to needs

In other models Upscaling images and passing as batches uses a lot of memory. In Yolov5" mosaic-data loader" is used which doesn't use a lot of memory. A lot of GPU memory is saved (Solawetz, 2020).

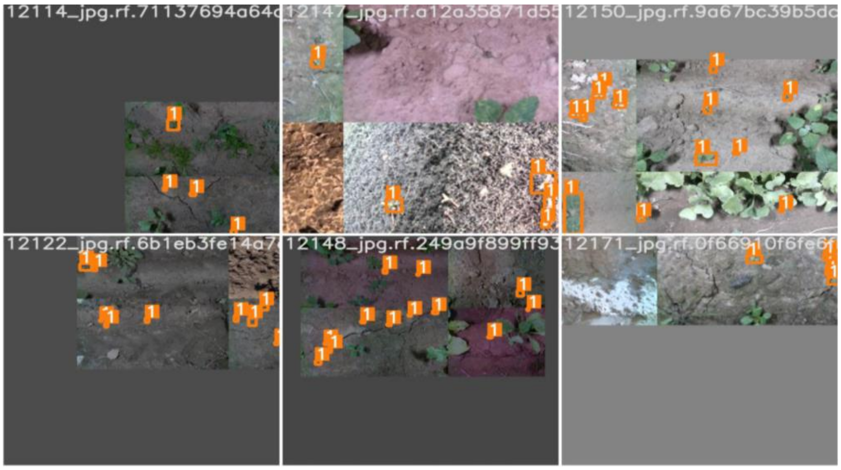


Fig – Mosaic DataLoader (Ground Truth Images)

<u>Metrics used of evaluation:</u>

<u>Confidence scores:</u> The confidence scores indicate the probability of an anchor box containing an object. The greater the confidence score the more chance is there of finding the target object i.e., germinating weeds within the predicted bounding box.

<u>IOU:</u> IOU stands for the Intersection over union i.e., the area of intersection over the area of union. It is used when the mean average precision is to be calculated. And its value ranges from 0 to 1 which basically indicates the amount of overlap between the predicted bounding box and the ground truth bounding box. An IOU value of 0 means that there is no overlap between the boxes and 1 indicated a complete overlap.

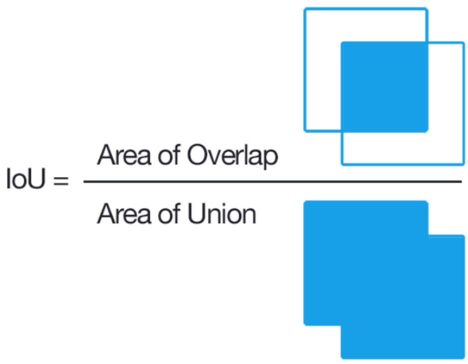


Fig - IOU

A detection is true positive or false positive depending upon the confidence scores and the IOU scores used as a performance measure. A detection is known as true positive if is satisfies 3 main conditions i.e.

- confidence score > threshold,
- the predicted class matches the class of a ground truth,

- the predicted bounding box has an IOU greater than e.g., 0.5 with the ground truth

The prediction is a false positive if any of the predefined conditions are not met. In case multiple predictions correspond to the same ground-truth, only the one with the maximum confidence score counts as a true positive, while the remaining are considered false positive. Further, if the confidence score of a detection used to detect a ground truth is lower than the threshold, the detection is FN (false negative) and if the confidence score of a detection which is not supposed to detect anything is lower than the threshold it is considered as TN (true negative) (Zeng, 2018).

Precision: It is defined as the number of the true positive divided by the sum of the true positive and false positive.

$$precision = \frac{TP}{TP + FP}$$

Recall: it is defined as the number of true positive divided by the sum of the true positive and the false negative (Zeng, 2018).

$$recall = \frac{TP}{TP + FN}$$

AP: AP depends upon the precision-recall curve for its evaluation. In layman's term AP is the precision averaged across all unique recall levels. first interpolate the precision at multiple recall levels before actually calculating AP. The interpolated precision" pinterp"at a certain recall level "rr" is defined as the highest precision found for any recall level r'≥rr'≥r (Zeng, 2018).

$$p_{interp}(r) = \max_{r' \geq r} p(r')$$

mAP: The mAP stands for the mean average precision and is calculated by aggregating a set of predicted bounding boxes and a set of object annotation present in the ground truth. Thus, for each of the predictions the IOU is calculated with respect to the ground truth bounding boxes. Now, the IOU score are subject to some threshold value that ranges from 0.5 to 0.95 and the corresponding predictions are then matched with the ground truth bounding boxes where the ones with the highest IOU score is matched first this is known as the greedy approach. And the precision-recall curve is generated for each of the class present in the dataset in my case 2 classes are present one of the weeds and the other of the crops. The AP is also computed by the algorithm. The model performance is judged by the PR (Precision Recall) curve by taking into account the true positives, false positives, true negatives and false negatives over a range of different confidence scores. For AP (Average Precision) calculation only one class is used but there are usually more than one class i.e., k>1. Hence Mean average precision is defined as the mean of AP across all "k" classes given by the below formula (Zeng, 2018):

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K}$$

Yolov5 Output and Experimental Results:

Training Metrics:

Once the model had been effectively trained on the dataset provided, I was able to generate the resultant output and the metrics that gave me a clear picture as to how the model performed on training and validation set and the final predictions were made on the test set.

So, on the training data set which comprised of about 1000 images the three losses i.e., the bounding box loss, class loss and the object loss show a decreasing trend. This is good as it can be observed that the model is performing well on the training data.

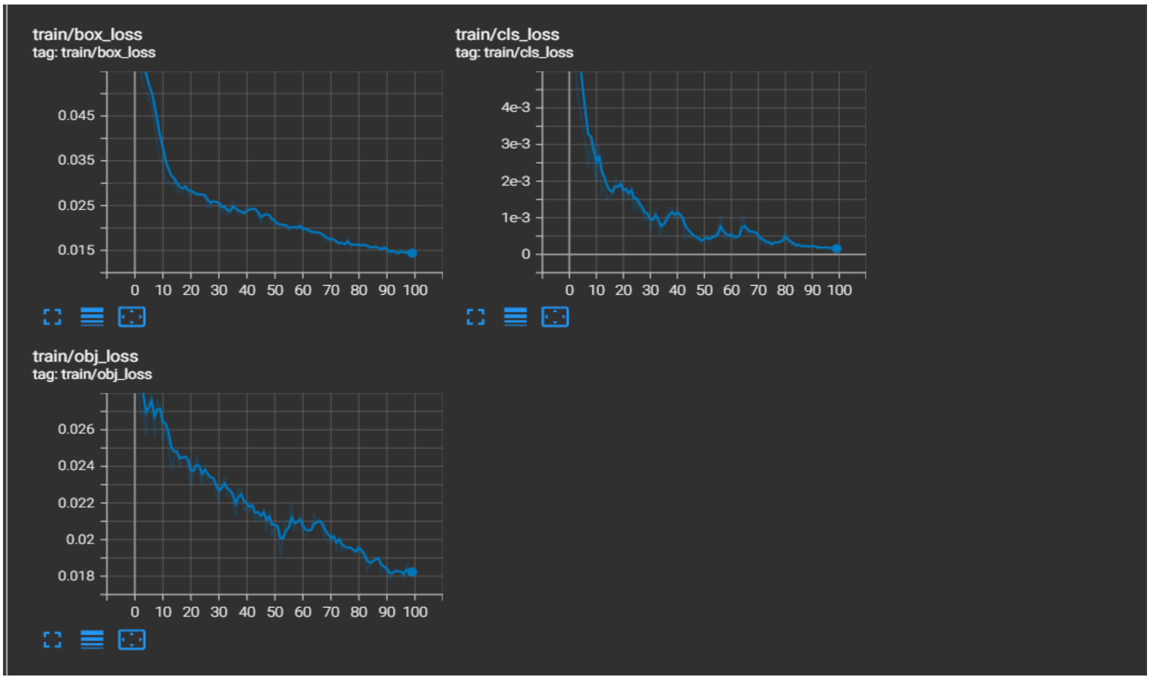


Fig – Training Metrics

The “training box_loss” is decreasing indicating that the predicted bounding boxes are getting closer to the ground truth bounding boxes. Initially this loss value is high at the start of the training process but with each subsequent epoch it shows a decreasing trend meaning that the model is able to correctly assign the bounding boxes to the germinating weeds in the training data set.

Secondly, the “training class_loss” is also showing a depreciating trend indicating that the detected targets i.e., germinating weeds are being correctly classified as weeds. It basically indicates the correctness of the detection. With each epoch the value of the “class_loss” decreases which is what I wanted on my training dataset.

Finally, the training object_loss is also decreasing which indicated the correctness of the detected class with the ground truth. This means that the detected images by the model are matching the annotated weeds in the ground truth. A lowering trend is an indication of good performance on the training dataset.

<u>Validation metrics:</u>

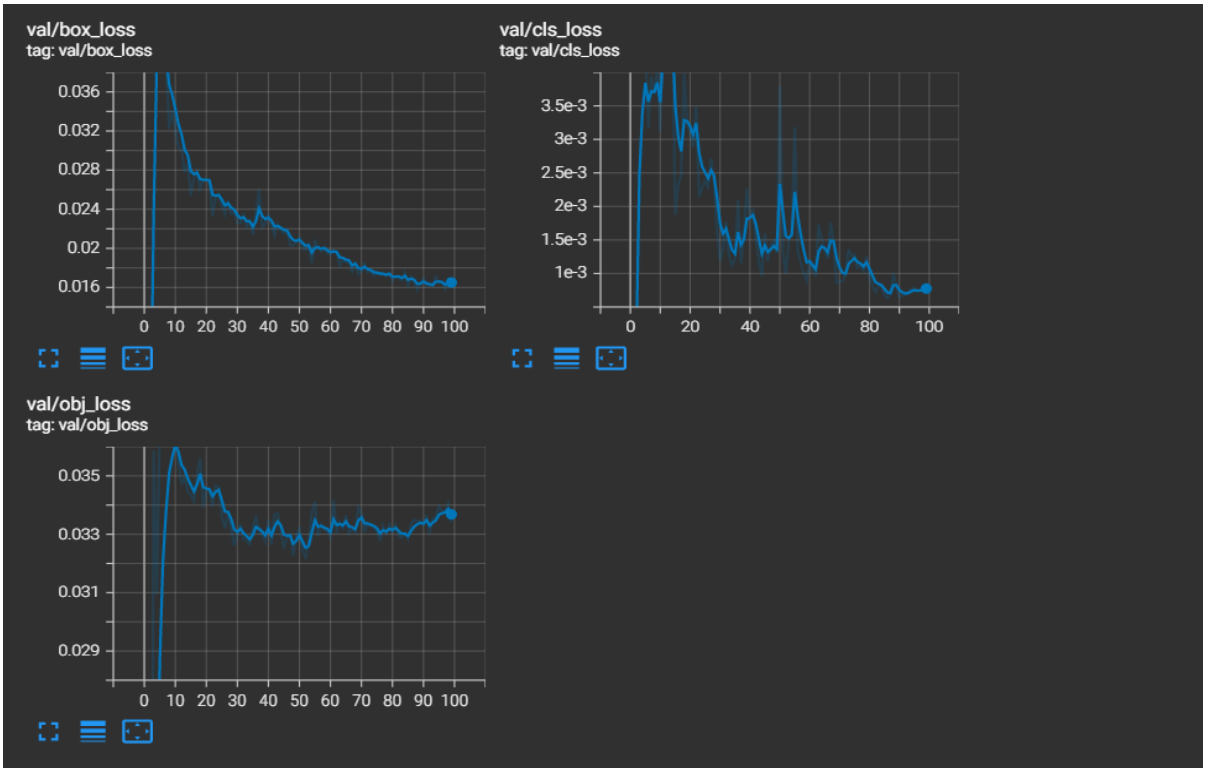


Fig – Validation Metrics

For the purpose of validation, I had used 300 images of the total 1500 images present in my dataset. In case of the validation metrics the all the three losses show a decreasing trend indicating that the model is performing well on the validation set as well. The "validation box_loss" is decreasing meaning that bounding boxes predicted by the model are getting closer to the bounding boxes in the ground truth. The "validation class_loss" is also decreasing meaning that the model is able to learn well from the training dataset and is able to correctly predict the class of weeds in the validation set as well. Further, the "validation object_loss" shows a depreciating trend indicating that the model is able to correctly classify the detected targets as weeds when compared to the labels in the ground truth. Thus, it can be said that the model is able to perform well on the validation set. It can also be said that good results on validation set means that the model will perform well on the test data which comprises the images that the model has never seen before.

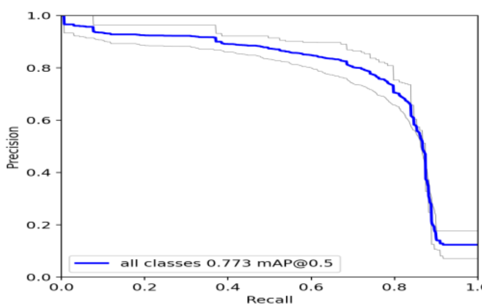


Fig – The Precision Recall Curve

The precision recall curve is similar to the ROC curve and is used to evaluate the performance of the model. Hence, the precision-recall curve indicates the algorithms performance at various threshold values. The curve above is plotted by calculating the precision and recall at various threshold values. Further, the curve above helps me to visualize how the choice of different threshold values affect the Yolov5 performance. Hence, it can be said that a model is said to have good performance because the precision stays high as the value of recall increases. Since, the aim of my project is to detect as many germinating weeds as possible it becomes necessary to have a high value of recall. Hence, by the end I was able to get a recall value of about 83% meaning that the model is 83% accurate in detecting weeds. Further, for an IOU 0.50 I was able to achieve an mAP score of 0.77, indicating that for 50% overlap of the predicted bounding boxes with ground truth occurs the model is 77% accurate in detecting the germinating weeds.

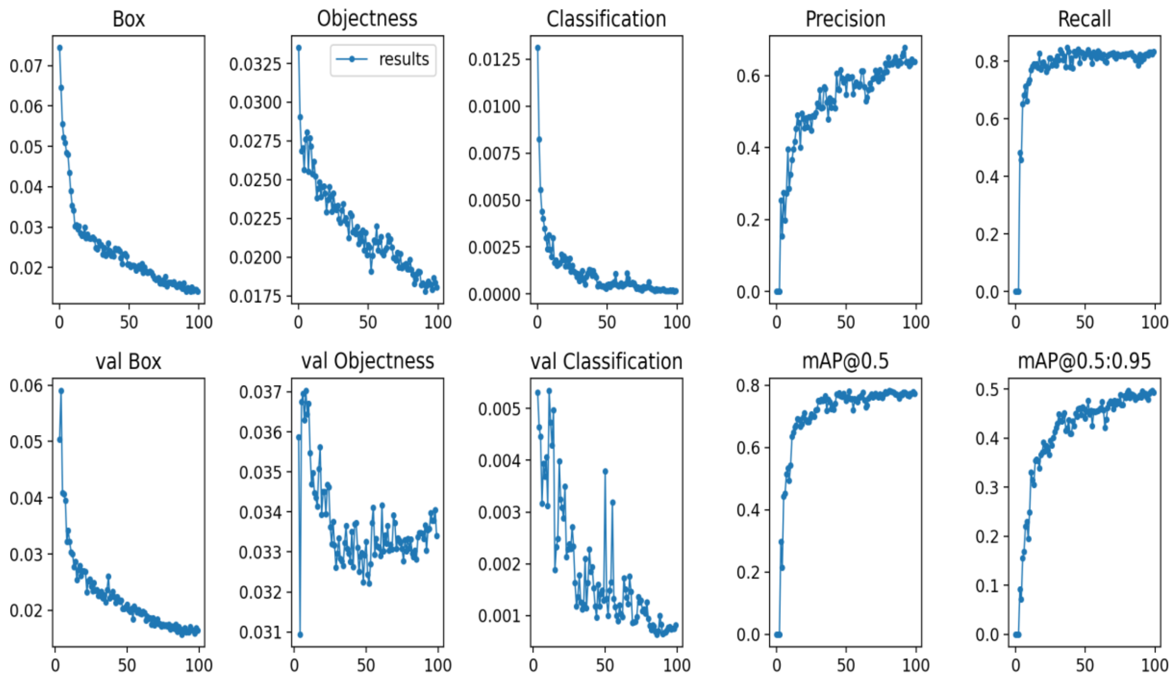


Fig – High Precision and High Recall thus high mAP score achieved.

Ground Truth Labels :

Fig : Ground Truth Labels

<u>Generated Predictions (Yolov5):</u>

Fig : Generated Predictions


crop 0.84
crop 0.72
weed 0.58
weed 0.82
weed 0.89
weed 0.78
weed 0.58
weed 0.80
weed 0.82
weed 0.87
weed 0.78
weed 0.72
weed 0.83
weed 0.65
weed 0.91
weed 0.72
weed 0.74
weed 0.85
weed 0.81
weed 0.58
weed 0.79
weed 0.78
weed 0.80
weed 0.70
weed 0.53
weed 0.73
weed 0.75
weed 0.66
weed 0.53
weed 0.75
weed 0.55

It is observed that the small germinating weeds are successfully detected by the Yolov5 algorithm. Also, the algorithm works very well in determining small weeds as can be seen from the ground truth labels and the generated output images. Compared to ground truth the Yolov5 algorithm is efficient in identifying and hence, detecting the weeds. Thus, satisfying the aim of my project.

FRCNN on Onions:

Model architecture:

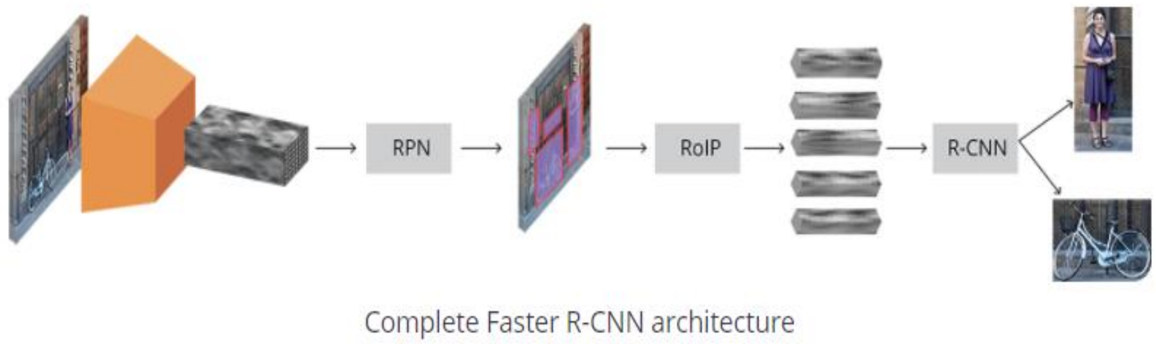


Fig – Faster RCNN Architecture

Another model I had used for the purpose of detecting germinating weeds in case of onions crops was the Faster RCNN model. The faster RCNN uses a Region Based convolutional Neural Network and uses a selective search approach to find the region of interest which are further passes to the ConvNets (Convolutional Nets). The target objects i.e., germinating weeds are detected by combining the pixels and the textures with the help of several rectangular boxes generated. In the end the outputs are further passed to Support Vector Machine for classification and finally the class probability is predicted by comparing the predicted bounding boxes and the ground truth bounding boxes. In case of the Faster RCNN model an RPN is used to propose the region of interest (labs, 2018).

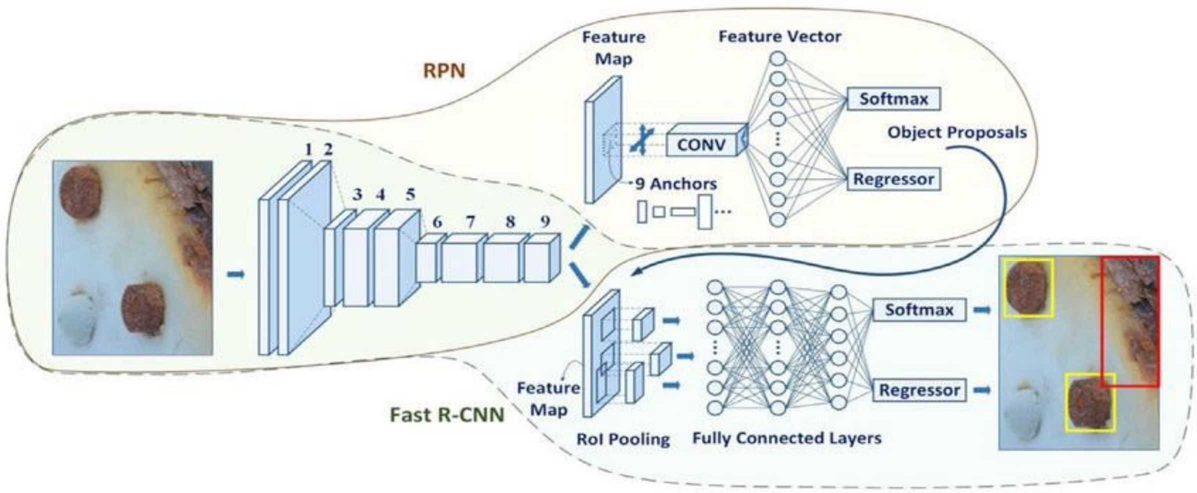
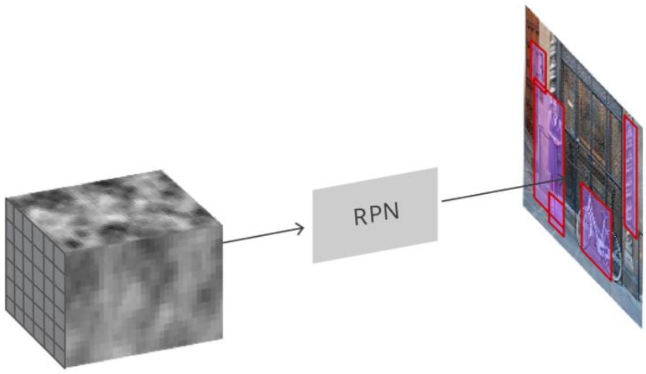


Fig – Faster RCNN Architecture

RPN: The RPN works by taking all the reference bounding boxes called anchors and gives a good set of region proposals for the network. This is done by generating two different outputs for each and every anchor. One of the outputs contains the probability that judges whether an anchor is the germinating weed, this is indicated by the “objectness score”. This objectness score is used to filter out the bad predictions generated. The second output is the “bounding box regression”, this is done to better fit the anchors on the object i.e., weeds it has to detect. The RPN is implemented in a fully convolutional way which uses the convolutional feature map that is returned by the base network as the input. Initially a convolutional layer with 512 channel is used with 3x3 kernel size with 2 layers of 1x1 kernel (Xu, 2018).



The RPN takes the convolutional feature map and generates proposals over the image

Fig – RPN (Region Proposal Network)

Once the input image of the onions was fed to the VGG16 pretrained model I was able to generate a feature map of 37x50x256 dimensions. The feature map was further defined with help of anchors. Now the Region Proposal Network was connected to a Convolution layer which contained 3x3 filters, 1 padding layer and 512 output channels. The output was further connected to two 1x1 convolutional layers for the task of detecting germinating weeds. In my case for the purpose of training I had taken all the anchors and had put them into two categories. One category contained anchors for the overlap of ground truth object with IOU greater than 0.5 and another category contained the anchors with no overlap with any of the ground truth objects. Hence, every anchor had 9 corresponding boxes in the original image which meant 37x50x9 = 16650 boxes were present in the original image. Further, I had applied non-maximum suppression to ensure that no overlapping of the proposed regions occurred (Xiaolong Wu, 2020).

Non-Maximum suppression (NMS) : It is a common tendency for the anchors to overlap and as a result the generated proposals overlap over the same target object hence to solve this problem of duplication non-maximum suppression is used. NMS solves this problem by taking the list of proposals generated and removes the proposals that have IOU larger than the predefined threshold value and favors the proposal that has a higher score. Hence, for my purpose I had taken a threshold value of 0.6. Further after applying the NMS the top proposals i.e., the proposals with the best scores are kept.

The next stage was the ROI (Region of Interest) pooling. Here a fixed size feature map was generated from the non-uniform inputs with the help of max-pooling technique on the input data. In case of ROI pooling each and every region of interest is gathered from the input and the corresponding section of the feature map is utilized and converted into a fixed dimension map. The next step I had undertaken was to flatten the proposed regions of interest and finally applying the “softmax function” to be able to generate the resultant output.

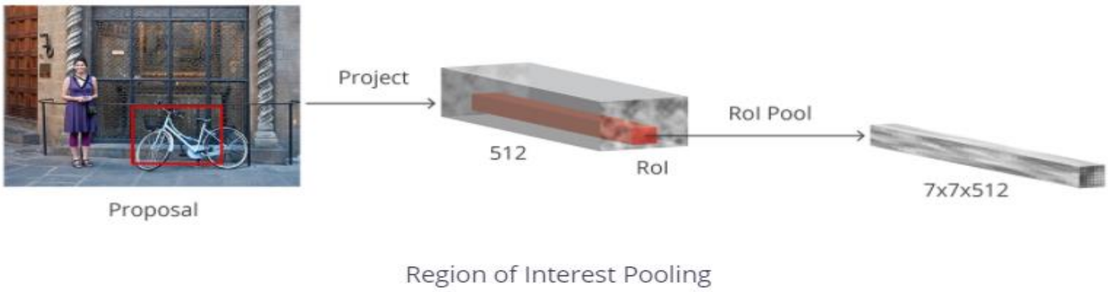


Fig – ROI Pooling

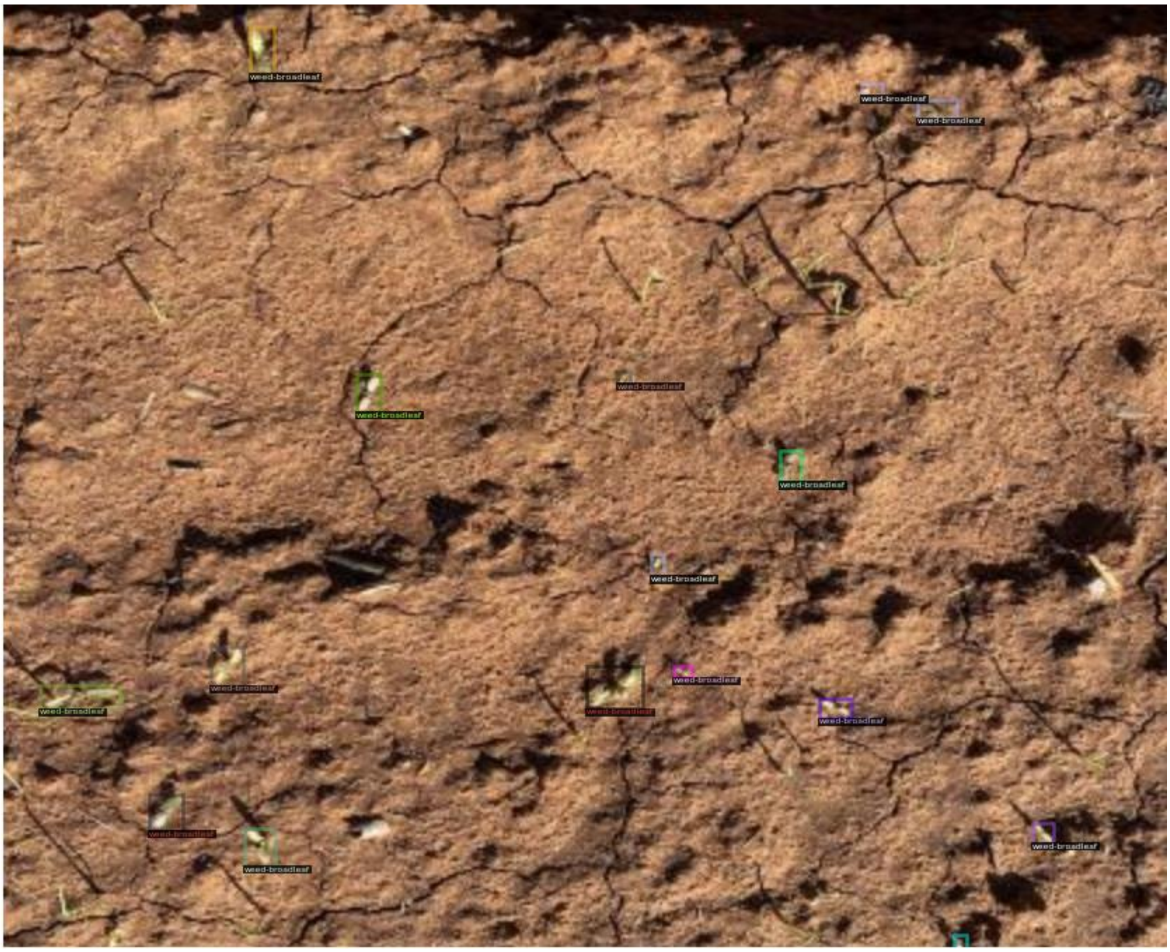


Fig – Training Data Fed to Faster RCNN after above steps

I had saved my images of onion with the germinating weeds in the csv format with the corresponding annotation and labels kept in two separate files. The annotations were saved as a “txt” file with the format like file_path, x1, y1, x2, y2, class_name. Here the file_path was the path of the images on my PC. The (x1, y1) and (x2, y2) indicated at the top left and the bottom right coordinates of the bounding box of the input image and class_name was the name of the class. Further, I had split the data as 70% for the purpose of training, 20% for the purpose of validation and finally 10% data set for the purpose of testing, these were the images that the model has not seen before (310 Images for training, 100 images for validation and 90 images for testing of 500 images of onion crops with weeds provided by Industrial Partner FLUX). The final step was to pass the proposals generated through the Region-based convolutional neural network in the FRCNN pipeline to get the desired output.

FRCNN output and Experimental Results:

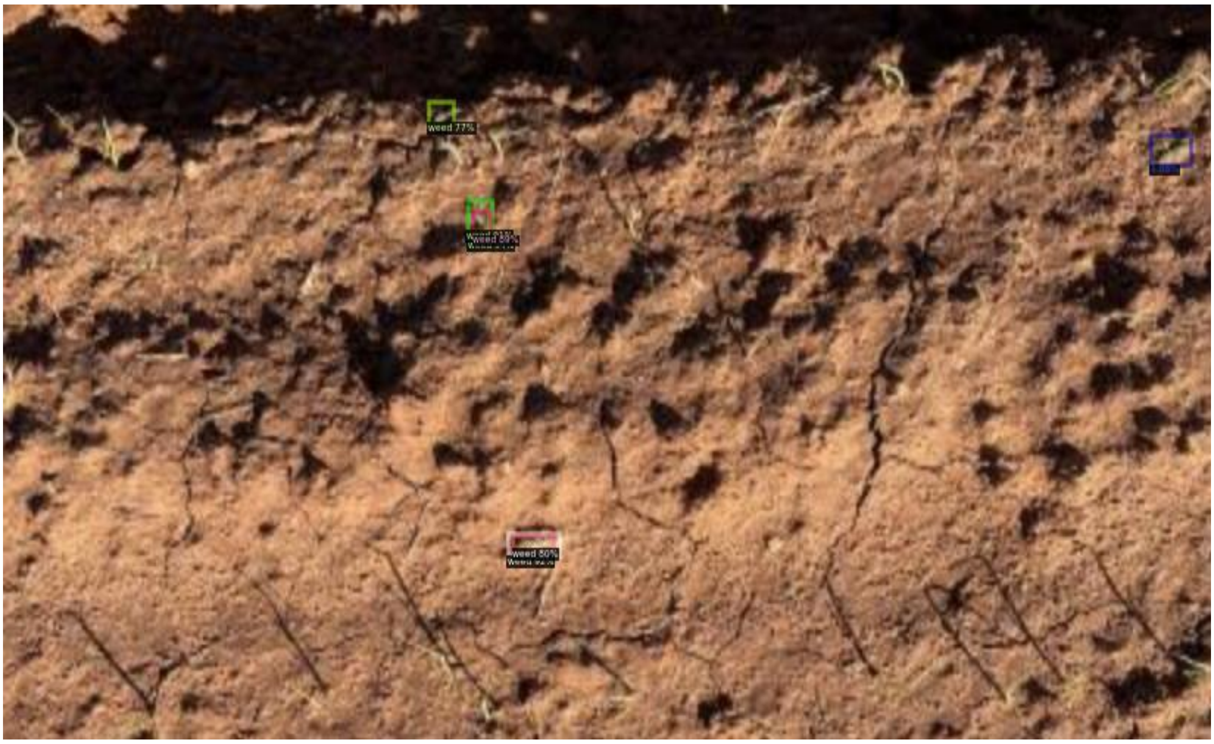


Fig – Generated Output Predictions

It can be seen in the results generated by the Faster RCNN model that small germinating weeds are being effectively detected. Further I was able to achieve an accuracy score of about 72% for the weeds and about 50% for the onions. Since there were not a lot of onions in the images provided by FLUX it shows lower accuracy. But incase of the germinating weeds which is the goal of my project almost all the small and germinating weeds were positively detected by the algorithm that I had developed. The metrics of my evaluation can be seen below. Since only onion (class 1) and weeds (class 0) were present in this onion-dataset rest of the classes have a value of 0.

category	AP	category	AP	category	AP
weed	nan	0	71.409	1	49.020
potato	0.000	weed-broadleaf	0.000	weed-grass	0.000

Plan vs Progress

I had initially understood how the weeds basically disrupt the chemical ecology of the plants around it, understanding the geographical impact of weed proved to be helpful as it enable to choose the right machine learning algorithm to go along with. I had to ensure how the camera would be able to capture the images and had asked the industry partner to supply me with relevant dataset that could serve as the basis of my detection task. I had taken into account the dynamic nature of the field as it was necessary to understand how quicky the weeds spread and when was the right time to deploy the autonomous bot to such that it was able to detect the germinating weeds and effectively remove them by spraying herbicides.

Another challenge I had faced early on was the labelling task which was tedious and time consuming. I had started of with the labelling of lettuces followed by onions. It was observed that the dataset I had was still less and more images were required to effectively train my ML model. So I had gather ed images of germinating weeds from various sources these included public sources and many ither University libraries which had done similar work. I had also made use of images of weeds present in Kaggle datasets, but it turned out that most of the images present of Kaggle were of mature weeds and germinating weeds. I had made use of a two-step process for my labelling task first step being, separation of the background and foreground using the maximum likelihood classification and then manually labelling the germinating weeds.

It was observed that if only a minute quantity if weed is present during the germination stage it becomes essential to able to detect this uncertainty in the field, as the ML algorithms are seldom prone to overfitting in such situations such as leading to very high accuracy. This posed a challenge as, if an ML algorithm is used it might result in a usually higher accuracy in the range of 90-99% it might be an optimistic result for the farmer with a naive understanding of the ML application but in actuality the 1% of weed detected has the potential of affecting the entire farm upon maturation further leading to increased cost to eliminate it. Thus, my ML model had to be trained with a sufficient quantity of the crop and weed dataset such that no overlapping occurs and biased results are not present.

Due to the small appearance of the germinating weeds in the farm I had concluded that the presence of weeds was much greater compared to the crop counterpart as the crops were sown in a ridge pattern whereas the weeds ahs germinated all over the place. Hence, it was necessary to train the detectors in a way such that it was able to correctly identify and detect small weeds present in between the crops. Further the pixels used to represent the germinating weeds were comparatively less to the crops present. Which meant that there was less information about the weeds for the detector to be trained on. For this purpose, I had prepared a high-quality dataset in which germinating weeds were present in high quantity such that the ML model could be trained effectively and give reasonable detections.

Future Works:

Since the model is performing well in detecting the germinating weed the next task will be use it for real time detection of weeds in a farm. This can be done by installing a processing mechanism onto an autonomous bot which can then traverse the field and gain information about the crops and the weeds through an optical sensor. This optical sensor could be in the form of a high-resolution camera capable of capturing the images real-time. The data then can be fed to the algorithm that has been trained on the germinating weeds and the output can be obtained by the autonomous bot. With the help of precision agriculture and site-specific spraying mechanism installed on the bot the chemical herbicide can be efficiently delivered at the sight of the germinating weed. This would be extremely helpful for the farmer as it would reduce human labor considerable and would also be economically beneficial for the farmer as the chemicals are quite costly and won't have to be sprayed on the entire farm that may result in damage of the crop plants as well.

Another thing that can be done is to sort out the weeds at different germinating stages once the information is collected by the autonomous bot. This would enable the farmer to determine how much herbicide should be sprayed onto the weed depending upon how mature this weed is. In case of recently germinating weed less harmful chemical can be sprayed with a limited quantity whereas for a robust and mature weed strong chemical can be used at varied quantity depending upon the strength of the chemical. This would ensure that the soil quality is maintained and the soil can be used again for sowing crops next season.

The presented object detection model can be further fine tuned as there is still research going in to increase the performance of such model and better accuracy can be attained later on. More data can also be gathered with germinating weeds in various crops and combined with the dataset that I have presented such that the computer vision model is able to detect germinating weeds in various other crops as well. Parameters such as "Prebudding"and "Cotyledon shape" can also be considered. A primitive technique is to wait for the early weed bud to show signs of cotyledons. It can be pertinent feature that can be utilized for differentiating the weed from the crop plant. It is usually determined by the nature of grass present around the weed plant which has significantly different characteristics than the soil around the crop plant and can lead to cotyledons of different shapes and sizes. This can be further reinforced into an ML algorithm for early stage weed detection. Once the weeds with the cotyledon have been identified by assessing the soil through spectroscopy methods it can give the farmers a head start for spraying of the pesticides. But it possesses another challenge that the weed has to be actively budding and growing in order to be detected and this growing weed is capable of spreading its seedlings through air. The weed plant should be small i.e., in pretillering phase and should be devoid of any signs of moisture stress.

Spectral and infrared characteristics can also be understood to make the model robust in nature. The application of precision agriculture already exists for the deployment of site-specific dosage of the herbicide. In this technique the presence of an optical weed sensor using the laser as a medium is generate images can be used to identify key feature to differentiate a weed and a crop plant. It is a novel approach as in contrast to the pixelated with the help of laser technology various attributes can be considered at a singular time. Further this technology also has a real time application. But how to use this technology for detecting weed upon germination still remains a challenge. Spectroscopic laser analysis can also help in generating heat maps that can prove to be essential in determining the weed as the weed plant as a high acceptance of infrared radiations compared to the crop plant and hence can be identified and eliminated.

Conclusion

Upon completion of my project, I was able to understand the application of Machine learning and Computer Vision in the agricultural sector. And how such a novel technology can provide the farmers with tremendous power by lowering the effort and saving a lot of money. With the passage of time, I firmly believe that technologies like precision agriculture, autonomous bot, micro aerial vehicle for deploying chemical on a farm will become more common than now. I was also able to understand the power of object detection algorithms like Yolov5 and Faster RCNN in making predictions better than a human ever could.

References

Alameh, P. K., 2014. *Development of an optical sensor for real-time weed detection using laser based spectroscopy.,* s.l.: Edith Cowan University.

F. Ozge ¨ Unel, C. C. ı., 2019. *The Power of Tiling for Small Object Detection,* s.l.: CVPR.

J.S., R., 2020. *The Train, Validation, Test Split and Why You Need It,* s.l.: Roboflow Blog.

Kapil, D., 2019. *Stochastic vs Batch Gradient Descent.* [Online]
Available at: https://medium.com/@divakar_239/stochastic-vs-batch-gradient-descent-8820568eada1

labs, t., 2018. *Faster R-CNN: Down the rabbit hole of modern object detection.* [Online]
Available at: https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection

Li Liu, B. Y., 2020. *Weed Detection in Images of Carrot Fields Based on Improved YOLO v5,* s.l.: IIETA.

Mustafeez, A. Z., 2017. *Educative.* [Online]
Available at: https://www.educative.io/edpresso/what-is-early-stopping

Neskorozhenyi, R., 2019. *Towards Data Science.* [Online]
Available at: https://towardsdatascience.com/tile-slice-yolo-dataset-for-small-objects-detection-a75bf26f7fa2

Özgeünel, F. a. Ö. B., 2019. *The Power of Tiling for Small Object Detection.* [Online]
Available at:
https://openaccess.thecvf.com/content_CVPRW_2019/papers/UAVision/Unel_The_Power_of_Tiling_for_Small_Object_Detection_CVPRW_2019_paper.pdf

Rajput, M., 2020. *Elephant Detector Training Using Custom Dataset & YOLOV5.* [Online]
Available at: https://towardsdatascience.com/yolo-v5-is-here-b668ce2a4908

Rajput, M., 2020. *YOLO V5 — Explained and Demystified.* [Online]
Available at: https://towardsai.net/p/computer-vision/yolo-v5%E2%80%8A-%E2%80%8Aexplained-and-demystified

Section, 2021. *Introduction to YOLO Algorithm for Object Detection.* [Online]
Available at: https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/

Shanmugam, S., 2018. *Automated Weed Detection Systems: A Review,* s.l.: Knowledge E.

Shu Liu, L. Q., 2018. *Path Aggregation Network for Instance Segmentation,* s.l.: IEEE.

Solawetz, J., 2020. *Roboflow Blog.* [Online]
Available at: https://blog.roboflow.com/boosting-image-detection-performance-with-data-augmentation/

Solawetz, J., 2020. *YOLOv5 New Version - Improvements And Evaluation.* [Online]
Available at: https://blog.roboflow.com/yolov5-improvements-and-evaluation/

Sydney, U. o., 2019. *ai.sydney.edu.au.* [Online]
Available at: https://weed-ai.sydney.edu.au/about

Vladan Stojni´, M. M., 2020. *A Method for Detection of Small Moving Objects in,* s.l.: MDPI.

Xiaolong Wu, S. A., 2020. *Robotic weed control using automated weed and crop,* s.l.: HAL.

Xu, Y., 2018. *Faster R-CNN (object detection) implemented by Keras for custom data from Google's Open Images Dataset V4.* [Online]
Available at: https://towardsdatascience.com/faster-r-cnn-object-detection-implemented-by-keras-for-custom-data-from-googles-open-images-125f62b9141a

Zeng, N., 2018. *An Introduction to Evaluation Metrics for Object Detection.* [Online]
Available at: https://blog.zenggyu.com/en/post/2018-12-16/an-introduction-to-evaluation-metrics-for-object-detection