

Deep Learning-Based Weed Identification for Precision Farming

¹ Jigar A. Soni, ² Dr. Hetal N. Patel

¹ Ph.D. Scholar, E.C. Engineering Department, Gujarat Technological University, Gujarat, India

² Professor & Head, E.C. Engineering Department A. D. Patel Institute of Technology New V.V.Nagar Dist:-Anand Gujarat, India

E-mail: ¹jasoni280489@gmail.com, ²ec.hetal.patel@adit.ac.in

ABSTRACT

According to a recent study and analysis in agriculture, various factors influence crop yield. Weeds are the most significant threat to crop yield. Weed control is a worldwide issue that has received much coverage in recent years. This paper presents a method for developing a deep convolutional neural network (CNN) for weed identification based on the modified YOLO architecture with several pre-processing techniques. An image labeler using the Roboflow framework is used to locate the regions of interest as part of the image processing. We have used novel Mosaic data augmentation in this model to address the well-known "small object detection problem." To train the developed model, we created 3600 images with different sizes of weed. Sizes of YOLO anchor box were calculated from the training dataset using a k-means clustering approach. The model that resulted was tested on 10% of the images. We may justify that the established model could detect weed with an appropriate recall rate and mAP based on the experimental results. This method determines whether an object on the farm is a weed by drawing a bounding box around it and assigning a label to it.

Keywords: *weed detection; convolutional neural networks; deep learning; Yolo; bounding boxes*

1. INTRODUCTION

Weeds can have several adverse effects if they are not monitored, including crop yield loss and the production of many seeds; thus, a weed seed bank has developed in the area and grain contamination during harvest. One of the most challenging tasks is weed control in aspects of producing more crops. Weed detection is essential for implementing proper weed control strategies in cropping systems. It is a huge challenge to create an automatic weed control system that eliminates human intervention, improves accuracy, and saves time[1]. Object recognition is a computer vision task that involves detecting and classifying objects in an image from a collection of classes. It can be divided into classification and localization computer vision tasks. The role of localization in computer vision entails locating an object within an image. There are many options for marking the location's boundary after the object has been identified, but a bounding box is the most popular. The object detection task is problematic because it requires both image recognition and localization. As a result, any network used for this task must be capable of locating and classifying objects of interest within an image. Though state-of-the-art methods have solved many challenges in object detection, the scale problem is also a problem in object detection. Because convolutional neural networks learn through feature detection, they frequently encounter issues when objects from different classes have similar features. Similarities in the pattern can lead to edge cases where other objects have similar characteristics at a smaller scale. These cases result in classification errors, lowering the network's accuracy. This is known as the small-scale similarity problem [2]. Agriculture is vigorously adopting artificial intelligence (A.I.) into its operations in these fields to overcome challenges such as labor shortages and rising demand. During peak seasons, farmers must hire expert farmworkers with the agricultural production experience for a wide range of activities, including sowing seeds, gathering fruit, weeding, and harvesting. Many of these functions are now done by

robots, and Weed recognition is an essential computer vision application that assists robots in these activities. Highly developed discriminative technologies are required to distinguish between crops and weeds for practical applications [3]. In lettuce crops, Osorio et al. [4] proposed three weed estimation approaches based on deep learning and visual recognition. Support vector machines (SVM) were used in one form, YOLOV3 (you only look once V3) was used in another, and Mask R-CNN was used in the third. Convolutional Neural Networks were used by dos Santos Ferreira et al.[5] to recognize and distinguish weeds in soybean crop videos (CNNs). A. M. Mishra et al.[6] discuss how to detect and interpret weed-based images in this article. Pre-processing, classification, and identification of crop weed and crop categorization are the steps carried out using image processing, artificial intelligence, and deep learning techniques. Researchers[7]found a total of 34 public image datasets. They classified them into three groups based on their intended uses: 15 datasets for weed control, ten datasets for fruit detection, and the remaining nine datasets for other applications. Bo et al. [8] provide a concise description of both emerging and common weed identification strategies for selective spraying, as well as a summary of recent trends in this field.

Nima et al.[8] demonstrates a convolutional neural network-based approach for estimating the growth stage of different weed species in the number of leaves. The images included monocots and dicots from 18 common Danish weed species or families. The network achieved an accuracy of 87 percent if we agree within 1 of the actual growth stage for these species, while the average accuracy for these species was 70 percent. Renjie Xu et al.[9] Suggest a novel ensemble learning approach for detecting forest fires in various scenarios in this paper. To begin, two separate learners, Yolov5 and EfficientDet, are combined to complete the fire detection process. To that end, we present a model for identifying different kinds of weeds based on object detection.

2. MATERIAL & METHODS

2.1. Weed Model flowchart for dataset, training, and detection process

As seen in Figure 1, the proposed methodology adopted for weed detection is composed of three stages. 1) Weed Dataset Preparation, 2) Training process of Weed Model, 3) Evaluation & Testing of Weed Model.

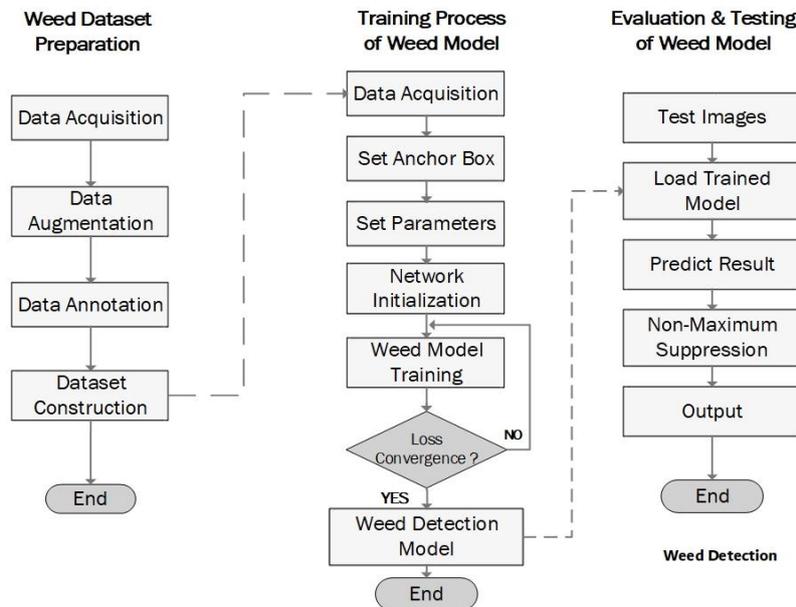


Figure 1 Weed Model flowchart for dataset, training, and detection process

2.2. Dataset Preparation

The Weed datasets used in this paper were taken using a digital camera with a 1080 x 1920-pixel resolution, RGB color space, and JPG storage format. All of the images were taken in natural daylight, capturing the complexities of the growing environments, such as illumination variation, occlusion, and overlap. The dataset consists of two raw image directories. One thousand four hundred thirty images of food crops and weeds, as well as directory annotations, i.e., their 1430 counterpart annotation XML files, can be augmented with 3600 images of food crops and weeds using data augmentation. We have used the Roboflow framework for pre-processing & augmentation techniques in computer vision. Roboflow

provides users with access to public datasets and the opportunity to upload their custom data. In addition, Roboflow supports a variety of annotation formats. All images were downsized to 640*640 pixels to investigate the effect of resizing on weed detection performance. Figure 2 shows some image samples from the generated dataset in various environments. We have used 70 % of images for training, 20 % of images for validation & 10 % of images for testing. After completing the necessary steps, we generate a version of the dataset and export it in the required format for all files in a zipped form[10].

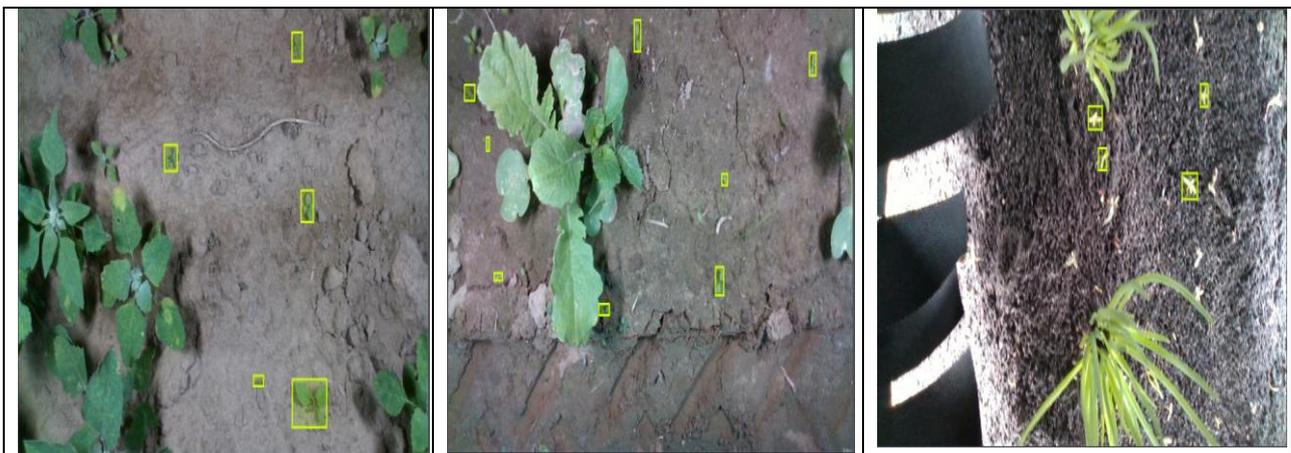


Figure 2 Weed samples with label what you see (LWYS) technique

2.3. The architecture of the Proposed YOLO model for Weed Detection

Yolov5's structure is based on a single-stage detection structure[9]. An overview of the modified weed detection model is shown in Figure 4, subdivided into the stages discussed below.

- 1) Data processing and the image input terminal: CutMix, Mosaic
- 2) Backbone: Using Focus, Cross Stage Partial Network, Leaky ReLU, and other techniques, it extracts the features of the high, middle, and low layers.

- 3) Neck: Extraction of large, medium, and minor feature maps by fusing features at different levels.
- 4) Head: Accomplish the final detection step, add the anchor box to the feature map, and generate the final output vector, including the class possibility, object score, and bounding box.
- 5) Loss: Determine the loss of the prediction outcome and ground truth, then back-propagate to change the model's parameters.

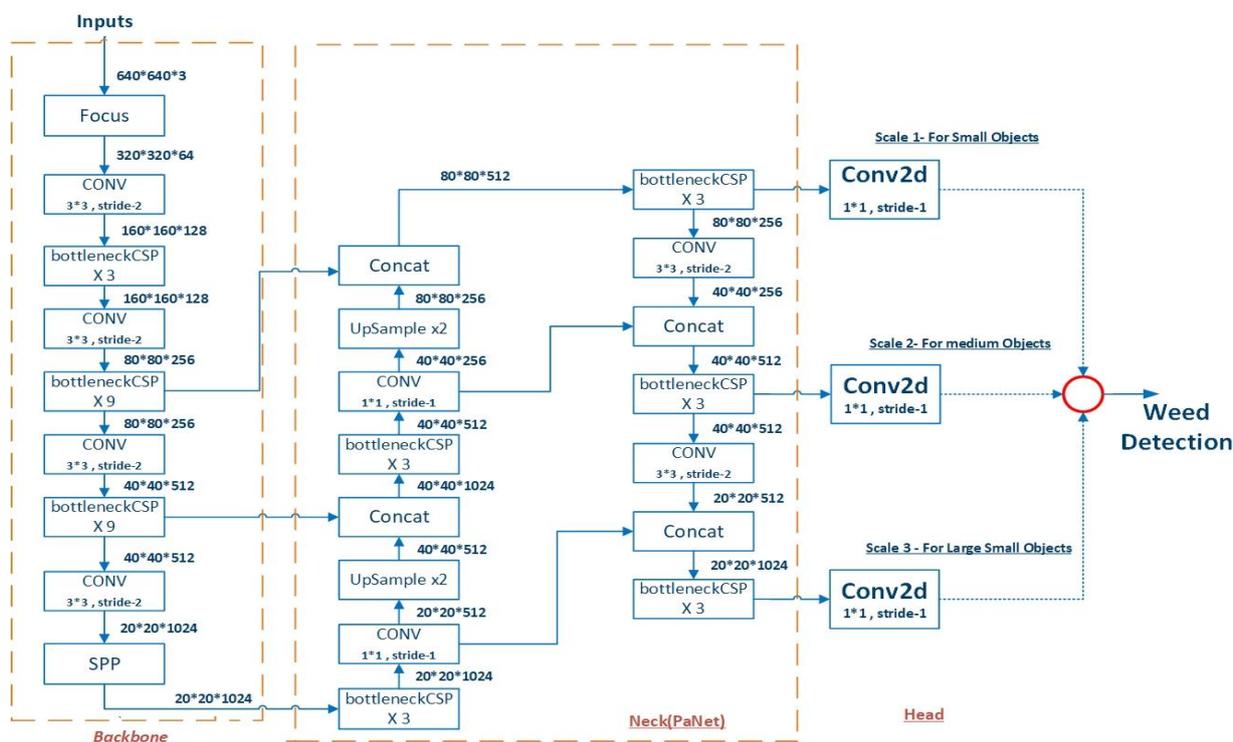


Figure 3 The architectural View of Proposed YOLO model for WEED DETECTION

2.3.1. Data processing improvements

Mosaic data enhancement: Mosaic data enhancement uses four images, random scaling, random cropping, and random arrangement, whereas CutMix only uses two for stitching[11].

Its main benefits are:

- Rich data set: Using four pictures at random, zooming randomly, and splicing randomly expands the detection data set; zooming randomly. The network is augmented by the addition of a large number of minor targets.

Reduce GPU usage: Some may contend that random scaling and ordinary data augmentation can be done as well; however, many people only have one GPU, so when Mosaic boosts training, the data of four photographs can be directly calculated, resulting in a different Mini-batch size and a GPU can achieve better results. Below figure 4 represents the results when the mosaic data enhancement is applied to the dataset in Figure 2.

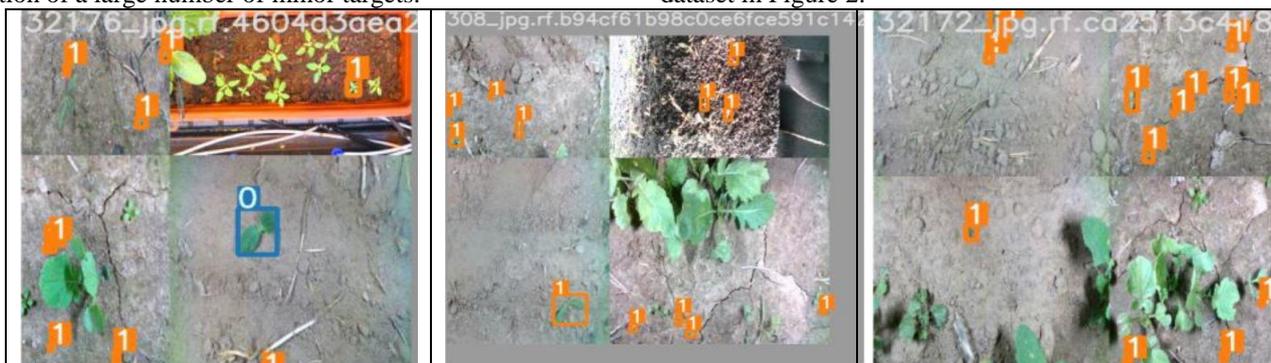


Figure 4 Weed dataset after Mosaic Data Enhancement

2.3.2. Backbone

Backbone is made up of a convolutional neural network that aggregates and shapes image features at various levels of granularity. Using Focus, CSP, Leaky ReLU, and other techniques, extract the features of the high, middle, and low layers.

a. Focus

Instead of stride-2 convolution, the first layer of the network performs pixel un-shuffling; the design of this

module is primarily to minimize the amount of calculation and speed up; Size of input in Yolov5 is 640x640x3. As depicted in Figure 5, the role of the focus layer is to copy it into four copies and then slice the four pictures into four 3x320x320 slices using the slicing operation. Then use concat to combine the four slices from the depth to generate a 12x320x320 output, and then use the convolutional layer with the number of convolution kernels set to 64 to generate a 64x320x320 output, and finally use batch norm and leaky relu to input the result to the next convolutional layer.

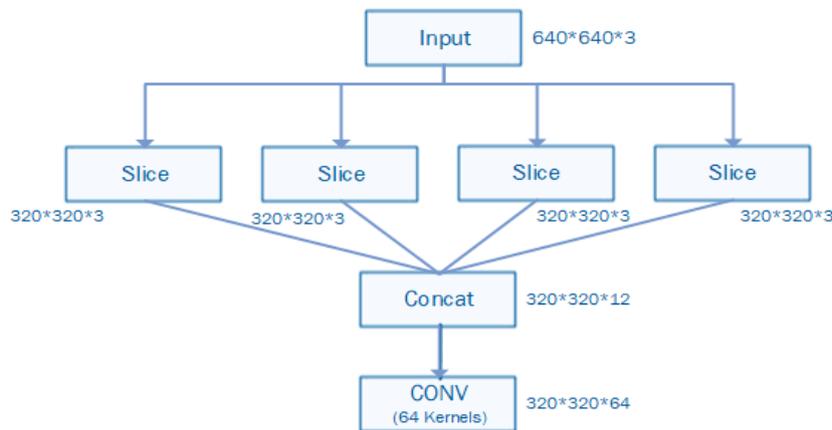


Figure 5 Focus Module

b. CSPNet- Cross Stage Partial Network

The main goal of CSPNet Cross Stage Partial Network was to make it possible for this architecture to achieve a more prosperous gradient combination while reducing computation time. This is accomplished by splitting the base layer's

function map into two parts and then combining them using a suggested cross-stage hierarchy, which is shown in figure 6. As a result, CSPNet will drastically reduce the amount of computation required and increase the inference speed and accuracy [12].

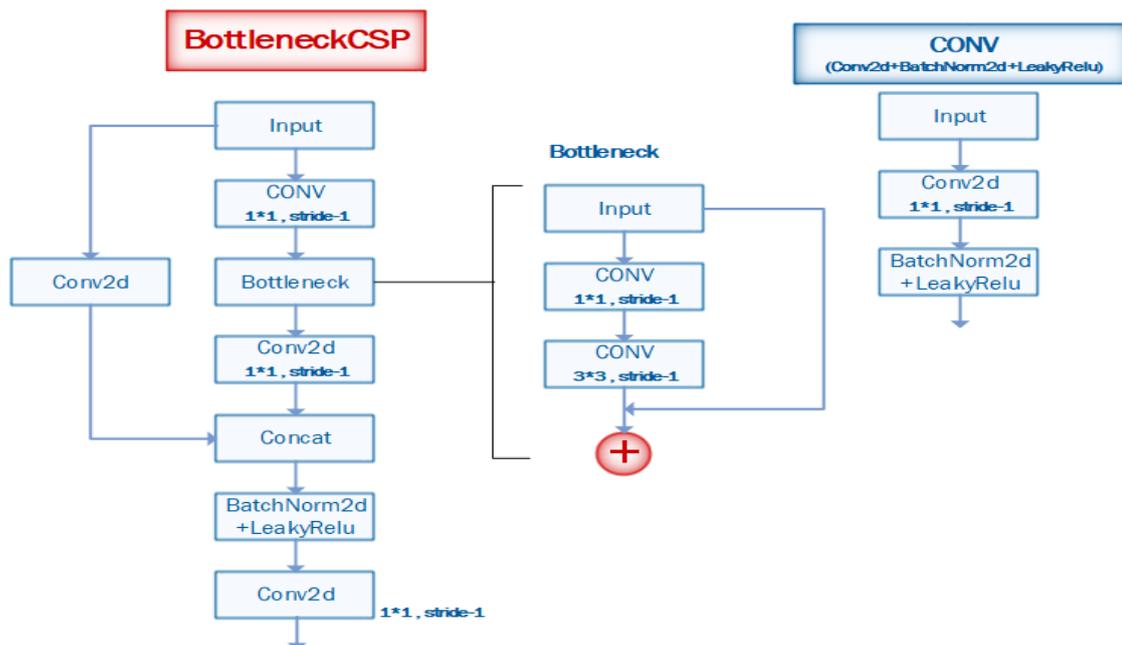


Figure 6 Cross Stage Partial Network

2.3.1. SPP-Spatial Pyramid Pooling

SPP is a pooling layer that removes the network's fixed-size constraint, allowing a CNN to work with any size input image. The maximum pooling method of 1×1 , 5×5 , 9×9 , and 13×13 is used for multi-scale fusion[13]. As shown in figure 7, the input is $512 \times 20 \times 20$, the output is $256 \times 20 \times 20$ after the 1×1 convolutional layer, and it is then downsampled by three parallel Max Pools of different kernel sizes (5, 9, 13). It is important to note that the max pool's padding is consistent throughout. After splicing the result, add it to the initial feature, output $1024 \times 20 \times 20$, and finally, with a 512 -convolution kernel, restore it to $512 \times 20 \times 20$.

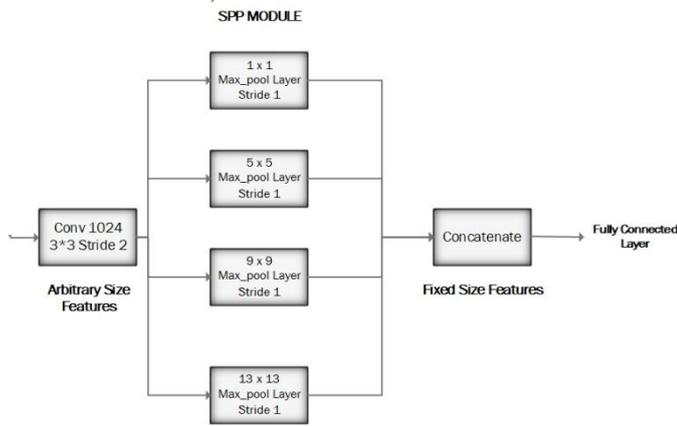


Figure 7 SPP MODULE

2.3.2. PANet Path Aggregation Network

PANet[14] is found in the YOLO model's neck, and it is primarily used to improve instance segmentation by preserving spatial information.

a. Feature Pyramid Network (FPN)

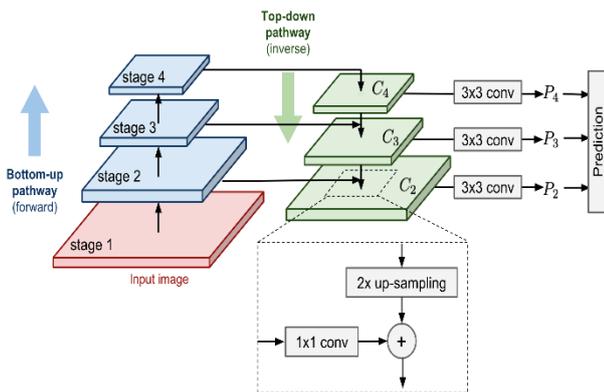


Figure 8 (a) FPN backbone. (b) Bottom-up path augmentation [15]

Bottom-Up Pathway

The feedforward computation of the backbone ConvNet is the bottom-up pathway. Each stage has one pyramid level. The reference set of feature maps for improving the top-down pathway by the lateral connection will output the last layer of each stage.

Top-Down Pathway and Lateral Connection

Higher-resolution features are upsampled from higher-pyramid-level feature maps, which are spatially coarser yet semantically stronger. For simplicity's purposes, the spatial resolution is upsampled by a factor of two using the nearest

neighbor. Each lateral link combines feature maps from the bottom-up and top-down pathways of the same spatial size. To minimize the channel dimensions, the feature maps from the bottom-up pathway uses 1×1 convolution. By element-wise addition, the feature maps from the bottom-up and top-down pathways are merged. Finally, a 3×3 convolution is applied to each integrated map to produce the final feature map, which is graphically represented in figure 8.

2.3.3. Head (detector) Output

The head's role in a single-stage detector is to perform dense prediction. The final prediction is the dense prediction, which is an array representing the calculated bounding box coordinates (center, height, and width), confidence score, and the label. Head creates three separate Feature maps to achieve a multi-scale prediction model that can accommodate small, medium, and oversized items.

2.3.4. Activation Function

In YOLO V5, the middle/hidden layer uses the Leaky ReLU activation function, while the final detection layer uses the Sigmoid activation function[16].

a. Leaky ReLU

Leaky ReLU is a ReLU variant. It is an attempt to find a solution to the diminishing ReLU problem. Rather than being completely zero, Leaky ReLU has a slight slope for negative values. When x is less than zero, a leaky ReLU will have $y = ax$. ($a=0.01$). It is called Randomized ReLU when a is not 0.01.

b. Sigmoid Function

The Sigmoid Function curve represents an S-shape. We use the Sigmoid Function because it occurs between two points (0 to 1). As a result, it is instrumental in models where the probability must be predicted as an output. Since the probability of something only exists between 0 and 1, the sigmoid is the best option.

3. EXPERIMENTAL SETUP & ALGORITHM IMPLEMENTATION

Experiments were carried out on a device with an Intel(R) Core (T.M.) i5-9300H CPU running at 2.40GHz and an NVIDIA GeForce GTX 1050 GPU. A batch size of 32 was chosen. The model was trained for 300 epochs at a learning rate of 0.001. Images with a resolution of 640×640 pixels are used as inputs. The momentum and weight decay rates were set to 0.937 and 0.0005, respectively.

3.1. Adaptive anchor frame calculation

To feed into the training phase, we need an estimate of the anchor boxes of the labelled images. Anchor boxes are a set of fixed-height and-width bounding boxes that are utilised to capture the scale and aspect ratio of the object classes being detected. They are usually selected based on the scale of the

objects in the training datasets. There will be anchor boxes with initial length and width in the Yolo algorithm for various data sets. During network training, the network generates a prediction frame based on the initial anchor frame, compares it to the ground truth of the actual frame, measures the distance between the two, and then updates the network parameters in a reverse way.

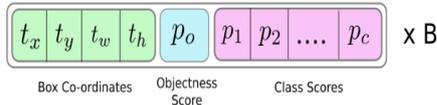


Figure 9 Attributes of Bounding Box[17]

The attributes of a bounding box are shown in figure 11, where t_x , t_y , t_w , and t_h are the box coordinates, P_0 is the objectness score, and P_1, P_2, P_3 . The class scores are P_c , and the number of bounding boxes is B .

We're not going to employ the default anchor box sizes in our network; instead, we used the K-means clustering approach to generate our anchor box sizes to cluster object bounding box sizes from the labeled training dataset. The dotted box in the diagram represents an anchor box, with p_w and p_h defining its width and height, respectively. The network predicts t_w and t_h , which, when exponentiated, scales the anchor box dimensions to match the detected object. The parameters t_x and t_y , respectively, indicate the x/y location of the center of the bounding box inside the cell. c_x and c_y represent the offset of the cell origin from the image origin to offset the bounding box center from the cell origin yields the x/y center of the bounding box for the image origin. The final predicted bounding box in figure 10, shown in blue, results from this[18].

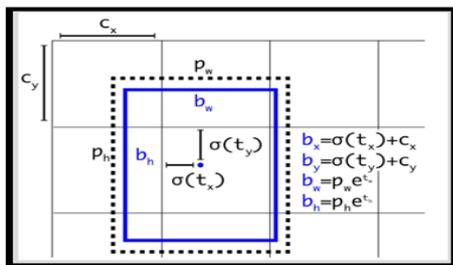


Figure 10 Bounding boxes with dimension priors and estimation of the position

3.2. The deep learning network's training procedure

We began training our network after establishing the optimal number of anchor boxes. The network was trained with a learning rate of 0.001 and a mini-batch size of 32 image samples over 300 epochs. An epoch is a complete training period in which all training vectors are being used for the first time to update the weights on the training set.

3.3. Performance Evaluation Metrics

Intersection over Union is the most commonly used loss function in object detection and instance segmentation tasks (IOU). Intersection Over Union (IOU), a Jaccard Index-based

test, tests the overlap between two bounding boxes. It is necessary to have a ground truth bounding box (B_{gt}) and a predicted bounding box (B_p). Using the IOU, we will determine whether detection is correct (True Positive) or not (False Positive). The area of intersection between the predicted and ground truth bounding boxes is divided by the Union area between them to measure IOU.[19]

$$IoU = \frac{areaB_p \cap areaB_{gt}}{areaB_p \cup areaB_{gt}}$$

3.4. Estimation of the training loss

To ensure that our training is correct, for each iteration, we look at the training loss. By adding the localization error, confidence loss, and classification loss, the Mean Squared Error (MSE) is determined. The discrepancy between the intended and ground truth bounding boxes called the localization error. When an object is discovered, the confidence loss is calculated, and the objectness error is measured when no object is detected. Finally, the squared error between the class conditional probabilities for each class is the classification loss[20].

3.5. Testing of the trained network

Ten percent of the images from each class are set aside for tests from the dataset. The goal is to see how well the network can distinguish between different types of weeds. In all ten percent of the images, the learned network detects weeds, as well as bounding boxes and class labels[20].

3.6. Non-Maximal Suppression (NMS)

The post-processing of target detection, as well as the screening of a large number of target frames, typically necessitates the use of the NMS. Only the optimal bounding box is held using Non-Maximal Suppression (NMS). The first step in NMS is to eliminate all predicted bounding boxes with a detection probability more diminutive than a predetermined NMS threshold. This NMS threshold was set to 0.6 in our experiment[21].

3.7. Precision-Recall and the Confidence Threshold

Precision and recall are essential factors to consider when assessing the trained network's overall results. Precision refers to the detector's ability to make correct classifications, while recall refers to the detector's ability to find all relevant objects. Bounding boxes are predicted by the object detector, each with a confidence score. Then, the probability of the object class appearing in the bounding box is calculated using the confidence score. As a result, we set a threshold to convert these confidence probabilities into classifications, with true positives (T.P.) defined as detections with a confidence score above the predetermined threshold and false positives defined as detections with a confidence score below the threshold (F.P.)[22].

4. EXPERIMENTAL RESULTS

4.1. Training loss

To assess the performance of the network training process, the training loss per network iteration was calculated. If the network has gone through 300 epochs in total, and we are just looking at the first 50-100 epoch; after the first 50-100 epoch, we can see that the training failure has decreased to about 50-60 %. As more iterations are completed, the model continues to learn more, resulting in reduced training loss in subsequent iterations. From the 300 epoch onwards, we can see a relatively constant loss. This means that the network's learning is becoming more accurate, and therefore the training loss is probably slight, which is graphically shown in figure 11[20].

4.2. Performance Metrics

Below table-1 summarizes the performance of proposed different yolov5 architecture against each evaluation metrics. The Objectness score, class probability, and bounding box regression score are used to calculate the loss. The loss of class probability and target score are calculated in yolov5 using binary cross-entropy and Logit's loss function.

Table 1 Recall rate & mAP for Different model depth

Yolo Models	Training Time	No of layers	Recall	mAP@0.5
Yolov5s	1 Hr 22 min	283	0.82	0.767
Yolov5m	2 hr 30 min	391	0.85	0.768
Yolov5l	4 hr 5 min	499	0.89	0.747
Yolov5x	7 hr 24 min	607	0.91	0.785

Figure 11 depicts the proposed network's training loss curve. As can be seen, the training loss dropped drastically at the beginning of the training stage before eventually settling at 300 epochs. In the validation dataset, we assessed the performance of the generated model[23].

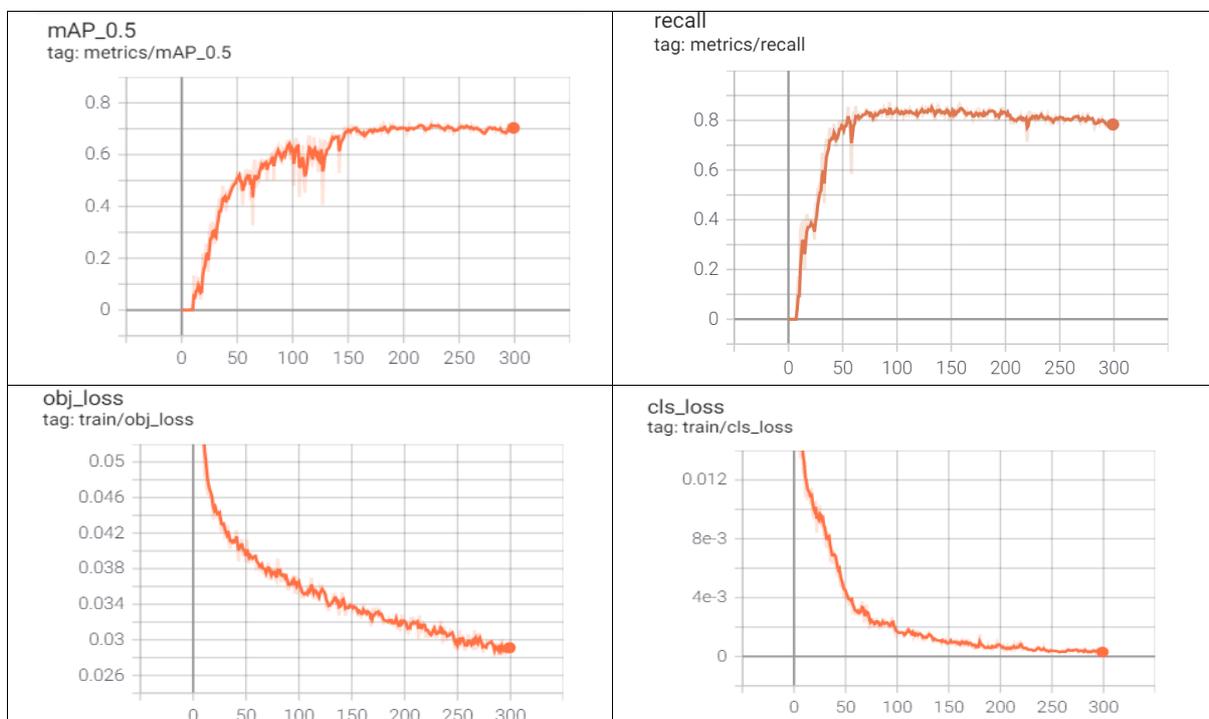


Figure 11 Various performance Metrics Evaluation

4.3. Detection/Classification of weeds using testing data

The weed model visualization results shown in Figure 14 were carried out to view the detected different sizes of weeds. This indicates that our model could identify weeds in real-time with better prediction, which is significant in agriculture.



Figure 12 Weed Detection results of Proposed Algorithm

5. CONCLUSION

In this paper, different Deep Learning yolov5 Models like yolov5s, yolov5m, yolov5l, and yolov5x were trained for more than 3600 images. After fine-tuning the parameters of the network, a Recall rate of 0.82 for yolov5s, 0.88 for yolov5m, 0.92 for yolov5l, and 0.94 for yolov5x had been achieved. From the experimental results obtained, we can justify that the developed model was able to detect weed with an acceptable recall rate & mAP. By identifying and detecting the various available weeds, this effort aims to contribute to novel deep learning technologies for precision farming. By circling an object on the farm with a bounding box and providing a label to it, this approach may determine whether or not it is a weed. This will enable weed detection and selective spraying/weeding easier for UAVs and ground robots, contributing to precision agriculture.

COMPLIANCE WITH ETHICAL STANDARDS

Any of the authors' investigations with human participants or animals are not included in this article.

ACKNOWLEDGMENT

The authors are grateful to A. D. Patel Institute of Technology, New V. V. Nagar's Principal and Head of the E.C. Department, for their relentless inspiration, guidance, and encouragement.

REFERENCES

- [1] R. Raja, D. C. Slaughter, and S. Fennimore, "A novel weed and crop recognition technique for robotic weed control in a lettuce field with high weed densities," *2019 ASABE Annu. Int. Meet.*, no. 1900029, pp. 1–5, 2019, doi: 10.13031/aim.201900029.
- [2] D. Bullock *et al.*, "Automated Weed Detection in Aerial Imagery with Context."
- [3] A. Khan, T. Ilyas, M. Umraiz, Z. I. Mannan, and H. Kim, "Ced-net: Crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture," *Electron.*, vol. 9, no. 10, pp. 1–16, 2020, doi: 10.3390/electronics9101602.
- [4] K. Osorio, A. Puerto, C. Pedraza, D. Jamaica, and L. Rodríguez, "A Deep Learning Approach for Weed Detection in Lettuce Crops Using Multispectral Images," *AgriEngineering*, vol. 2, no. 3, pp. 471–488, 2020, doi: 10.3390/agriengineering2030032.
- [5] A. dos Santos Ferreira, D. Matte Freitas, G. Gonçalves da Silva, H. Pistori, and M. Theophilo Folhes, "Weed detection in soybean crops using ConvNets," *Comput. Electron. Agric.*, vol. 143, pp. 314–324, Dec. 2017, doi: 10.1016/j.compag.2017.10.027.
- [6] A. M. Mishra and V. Gautam, "Weed species identification in different crops using precision weed management: A review," *CEUR Workshop Proc.*, vol. 2786, pp. 180–194, 2021.

©2012-21 International Journal of Information Technology and Electrical Engineering

- [7] Y. Lu and S. Young, "A survey of public datasets for computer vision tasks in precision agriculture," *Comput. Electron. Agric.*, vol. 178, no. July, p. 105760, 2020, doi: 10.1016/j.compag.2020.105760.
- [8] N. Teimouri, M. Dyrmann, P. R. Nielsen, S. K. Mathiassen, G. J. Somerville, and R. N. Jørgensen, "Weed growth stage estimator using deep convolutional neural networks," *Sensors (Switzerland)*, vol. 18, no. 5, pp. 1–13, 2018, doi: 10.3390/s18051580.
- [9] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A Forest Fire Detection System Based on Ensemble Learning," pp. 1–17, 2021.
- [10] K. Sudars, J. Jasko, I. Namatevs, L. Ozola, and N. Badaukis, "Dataset of annotated food crops and weed images for robotic computer vision control," *Data Br.*, vol. 31, p. 105833, 2020, doi: 10.1016/j.dib.2020.105833.
- [11] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020.
- [12] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.
- [13] M. O. Lawal, "Tomato detection based on modified YOLOv3 framework," *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, 2021, doi: 10.1038/s41598-021-81216-5.
- [14] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8759–8768, 2018, doi: 10.1109/CVPR.2018.00913.
- [15] X. Li, T. Lai, S. Wang, Q. Chen, C. Yang, and R. Chen, "Weighted feature pyramid networks for object detection," *Proc. - 2019 IEEE Intl Conf Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCloud/SustainCom/SocialCom 2019*, pp. 1500–1504, 2019, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217.
- [16] S. Sharma, S. Sharma, and A. Athaiya, "Activation Functions in Neural Networks," *Int. J. Eng. Appl. Sci. Technol.*, vol. 04, no. 12, pp. 310–316, 2020, doi: 10.33564/ijeast.2020.v04i12.054.
- [17] K. Majek and A. Kathuria, "How to implement a YOLO (v3) object detector from scratch in PyTorch: Part 4," *Hello Pap.*, p. 1, 2018, [Online]. Available: <https://blog.paperspace.com/how-to-implement-a-yolo-v3-object-detector-from-scratch-in-pytorch-part-5/>
- [18] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv*, 2018.
- [19] A. Kamal, "YOLO, YOLOv2, and YOLOv3: All You want to know | by Amro Kamal | Medium," *Medium*, pp. 1–33, 2019, [Online]. Available: https://medium.com/@amrokamal_47691/yolo-yolov2-and-yolov3-all-you-want-to-know-7e3e92dc4899.
- [20] M. Abdulsalam and N. Aouf, "Deep weed detector/classifier network for precision agriculture," *2020 28th Mediterr. Conf. Control Autom. MED 2020*, pp. 1087–1092, 2020, doi: 10.1109/MED48518.2020.9183325.
- [21] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," *arXiv*, no. 2, 2019, doi: 10.1609/aaai.v34i07.6999.
- [22] P. Sought, "ProgrammerSought IoU," pp. 1–13, 2021.
- [23] Y. He, T. P. Pridmore, and J. G. Pieters, J. Gao, A. P. French, "Deep convolutional neural networks for image-based *Convolvulus sepium* detection in sugar beet fields," *Plant Methods*, vol. 16, no. 1, pp. 1–12, 2020, doi: 10.1186/s13007-020-00570-z.

AUTHOR PROFILES

Jigar A. Soni is a Ph.D. candidate in the Department of Electronics and Communication Engineering at *GUJARAT TECHNOLOGICAL UNIVERSITY Dist:-Ahmedabad, Gujarat, India*. He received his master's degree from the *GUJARAT TECHNOLOGICAL UNIVERSITY* in 2014 and a Bachelor's degree in Electronics and Communication from Charotar University of Science and Technology-Changa in 2010. His current research is concerned with fruit and weed identification using computer vision.

Dr. Hetal N. Patel is Professor & Head in Electronics and Communication Engineering Department at A. D. Patel Institute of Technology, New V.V.Nagar, Dist: Anand, Gujarat, India. In 2013, she obtained her Ph.D. from S.P.University, V.V.Nagar. She is a pioneer in image processing, machine learning, and deep learning, with over 30 academic papers to her credit. She has led her research team in ISRO Respond projects as the Principal Investigator. She is Student Focused Technical Education Teacher and dedicated to Creating a Positive Learning Environment & Making a Difference in Student Lives.