WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Plant Detection Using Yolo V2 Deep Learning Algorithm

P.Hemanth, K. Mohan, S.Moinuddeen, Dr.Upendra KumarVerma

Faculty of Electronic and Communication Engineering, Madanapalle Institute of Technology & Science, Madanapalle

ABSTRACT

In current days, weed recognizable proof/plant identification in plants is more troublesome. There has been little work such a long way to distinguish weeds while establishing vegetables. Customary methodologies for the recognizable proof of horticultural weed were basically aimed at straightforwardly distinguishing weed by the by, the varieties in weed species are critical. This study presents an original strategy, which consolidates profound learning with imaging innovation, instead of this technique. To start with, the YOLO v2 model was prepared to recognize and draw limit boxes for plants encompassing it. Then, at that point, the excess green things tumbled from the line boxes like weeds. In this methodology, simply the harvests are distinguished, and other weed species are consequently kept from being dealt with. These examination results exhibit the practicality of involving the proposed strategy for the ground-based plant distinguishing proof in vegetable estate.

Keywords: Plant identification, deep learning, image processing, deep learning, YOLO v2.

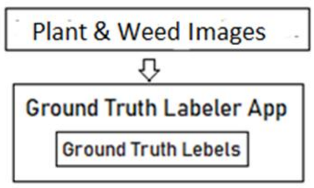
INTRODUCTION

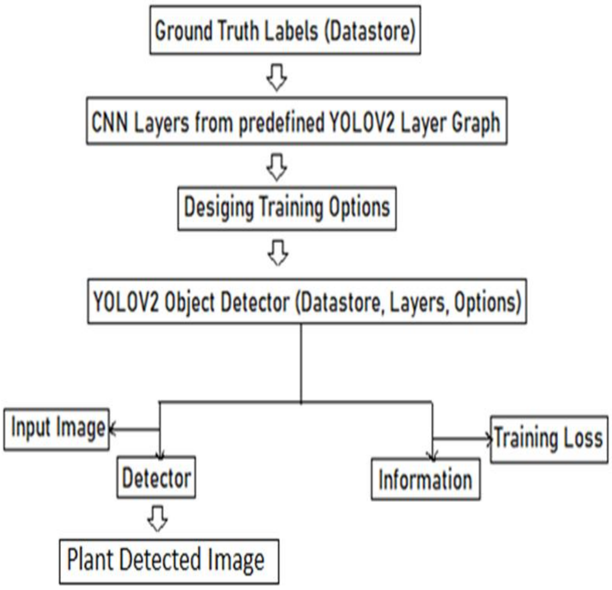
A weed is any plant that requires a type of activity to decrease its impact on the economy, the climate, human wellbeing, and convenience. Weeds are otherwise called obtrusive plants. Many plants brought into Australia over the most recent 200 years are presently weeds. Weeds regularly produce huge quantities of seeds, helping their spread. They are frequently fantastic at making do and duplicating in upset conditions and are normally the first species to settlements and overwhelm in quite a while. A weed can be an outlandish species or a local animal variety that colonizes and perseveres in a biological system in which it didn't beforehand exist. Weeds can occupy all conditions, from our towns and urban communities through to our seas, deserts, and elevated regions. A few weeds are of specific concern and, therefore, have been recorded for need the board or in regulation.

The trait of weeds to have the option to answer quickly to unsettling influences, for example, environmental change, may give them an upper hand over less forceful species. The effects of environmental change on single species and biological systems are probably going to be intricate. Environmental change, as well as the connections between environmental change and different cycles, (for example, changes to land use and to fire systems), may likewise turn a few as of now harmless species (both local and non-local) into obtrusive species and may prompt sleeper weeds turning out to be even more effectively weedy.

Environmental change is supposed to build the gamble of intrusion by weeds from adjoining domains. Environmental change may likewise incline toward weeds that have previously settled in Australia however are presently limited in range, empowering them to build their reach. As climatic zones shift, weeds that are equipped for fast dispersal and foundation can possibly attack new regions and increment their reach.

Weeds that are appropriate to adjust to the effects of environmental change may not just fill holes left by additional weak local plants, they might have a significantly more prominent impact by modifying the arrangement of biological systems and their uprightness. As a matter of fact, environmental change might lean toward specific local plants so much that they then become weeds. Expanding levels of carbon dioxide may likewise affect plant development rates, which might cause changes in weed spread.





Our Proposed Method

MODELDESCRIPTION

In YOLOv2 the subtleties of each block in the perception should be visible to drifting over the block. Every Convolution block has the Batch Norm standardization and afterward Leaky Relu actuation except for the last Convolution block. Just go for it partitions the info picture into a S×S lattice. Every lattice cell predicts just a single item. For instance, the yellow framework cell beneath attempts to anticipate the "individual" object whose middle (the blue dab) falls inside the lattice cell. Every network cell predicts a decent number of limit boxes. In this model, the yellow lattice cell makes two limit box expectations (blue boxes) to find where the individual is. In any case, the one-object rule limits how close identified items can be.
For every lattice cell,

- it predicts B limit boxes, and each crate has one box certainty score,
- it identifies one article just no matter what the quantity of boxes B,
- it predicts C restrictive class probabilities (one for every class for the likeliness of the article class).

The limit boxes contain box certainty score. The certainty score reflects how likely the case contains an article (objectless) and how precise is the limit box. We standardize the bouncing box width w and level h by the picture width and level. x and y are counterbalances to the relating cell. Subsequently, x, y, w and h are somewhere in the range of 0 and 1. Every cell has 20 restrictive class probabilities. The restrictive class likelihood is the likelihood that the recognized item has a place with a specific class (one likelihood for every class for every cell). The class certainty score for every expectation box is registered as.

class confidence score = box confidence score * conditional class probability

It estimates the certainty on both the grouping and the limitation (where an article is found). We might stir up those scoring and likelihood terms without any problem. Here are the numerical definitions for your future reference.

$$\begin{aligned}
\text{box confidence score} &\equiv P_r(object) \cdot IoU \\
\text{conditional class probability} &\equiv P_r(class_i|object) \\
\text{class confidence score} &\equiv P_r(class_i) \cdot IoU \\
&= \text{box confidence score} \times \text{conditional class probability}
\end{aligned}$$

where

$P_r(object)$ is the probability the box contains an object.
IoU is the IoU (intersection over union) between the predicted box and the ground truth.
$P_r(class_i|object)$ is the probability the object belongs to $class_i$ given an object is presence.
$P_r(class_i)$ is the probability the object belongs to $class_i$

Just go for it predicts numerous jumping boxes per lattice cell. To register the misfortune for the genuine positive, we just maintain that one of they should be liable for the item. For this reason, we select the one with the most noteworthy IOU (crossing point over association) with the ground truth. This procedure prompts specialization among the bouncing box forecasts. Every expectation gets better at foreseeing specific sizes and viewpoint proportions.

Classification loss:

If an object is the classification loss at each cell is the squared error of the class conditional probabilities for each class:

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

where

$\mathbb{1}_i^{obj} = 1$ if an object appears in cell i, otherwise 0.

$\hat{p}_i(c)$ denotes the conditional class probability for class c in cell i.

The localization loss measures the errors in the predicted boundary box locations and sizes. We only count the box responsible for detecting the object.

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$

where

$\mathbb{1}_{ij}^{obj} = 1$ if the j th boundary box in cell i is responsible for detecting the object, otherwise 0.

λ_{coord} increase the weight for the loss in the boundary box coordinates.

We do not want to weight absolute errors in large boxes and small boxes equally. i.e. a 2-pixel error in a large box is the same for a small box. To partially address this, YOLO predicts the square root of the bounding box width and height instead of the width and height. In addition, to put more emphasis on the boundary box accuracy, we multiply the loss by λ coordinates (default: 5).

With regards to Machine Learning, counterfeit brain network performs all around well. Counterfeit Neural Networks are utilized in different grouping task like picture, sound, words. Various sorts of Neural Networks are utilized for various purposes, for instance for foreseeing the arrangement of words we utilize Recurrent Neural Networks even more exactly a LSTM, likewise for picture characterization we use Convolution Neural Network. In this we will fabricate essential structure block for CNN. A convolutional brain organization can comprise of one or different convolutional layers. The quantity of convolutional layers relies upon the sum and intricacy of the information.

Prior to plunging into the Convolution Neural Network, let us initially return to certain ideas of Neural Network. In a normal Neural Network, there are three sorts of layers:

Input Layers: It's the layer where we give contribution to our model. The quantity of neurons in this layer is equivalent to add up to number of highlights in our information (number of pixels in the event of a picture).

Hidden Layer: The contribution from Input layer is then feed into the secret layer. There can be many secret layers relying on our model and information size. Each secret layer can have various quantities of neurons which are for the most part more prominent than the quantity of highlights. The result from each layer is processed by grid augmentation of result of the past layer with learnable loads of that layer and afterward by expansion of learnable inclinations followed by actuation capability which makes the organization nonlinear.

Output Layer: The result from the secret layer is then taken care of into a calculated capability like sigmoid or SoftMax which changes over the result of each class into likelihood score of each class.

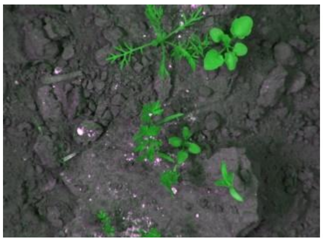
The information is then taken care of into the model and result from each layer is gotten this step is called feed forward, we then compute the blunder utilizing a mistake capability, some normal blunder capabilities are cross entropy, square misfortune mistake and so forth. From that point onward, we back spread into the model by ascertaining the subsidiaries. This step is gotten back to spread which fundamentally is utilized to limit the misfortune.

A Convolutional brain organization (CNN) is a brain network that has at least one convolutional layer and are utilized predominantly for picture handling, order, division and furthermore for other auto related information. A convolution is basically sliding a channel over the information. One supportive method for contemplating convolutions is this statement from Dr Prasad Samarakoon: "A convolution can be thought as "taking a gander at a capability's environmental elements to improve/precise expectations of its result." Rather than taking a gander at a whole picture on the double to find specific highlights it very well may be more viable to check out at more modest bits of the picture. The most widely recognized use for CNNs is picture arrangement, for instance distinguishing satellite pictures that contain streets or characterizing transcribed letters and digits. There are other very standard errands like picture division and sign handling, for which CNNs perform well at. CNNs have been utilized for understanding in Natural Language Processing (NLP) and discourse acknowledgment, albeit frequently for NLP Recurrent Neural Nets (RNNs) are utilized.

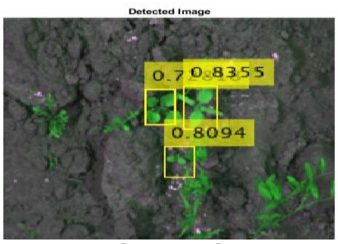
A CNN can likewise be carried out as a U-Net design, which are basically two nearly reflected CNNs bringing about a CNN whose engineering can be introduced in a U shape. U-nets are utilized where the result should be of comparable size to the info like division and picture improvement. Each convolutional layer contains a progression of channels known as convolutional parts. The channel is a lattice of whole numbers that are utilized on a subset of the information pixel esteems, a similar size as the part. Every pixel is duplicated by the relating esteem in the bit, then the outcome is summarized for a solitary incentive for effortlessness addressing a network cell, like a pixel, in the result channel/highlight map. These are straight changes; every convolution is a kind of relative capability. In PC vision the information is much of the time a 3 channel RGB picture. For effortlessness, on the off chance that we take a greyscale picture that has one channel (a two-layered framework) and a 3x3 convolutional portion (a two-layered grid). The piece steps over the information grid of numbers moving evenly segment by section, sliding/looking over the main columns in the lattice containing the pictures pixel values. Then the portion steps down upward to ensuing columns.

RESULT

First, we have taken a couple of sample inputs from the tomato field. We have taken several different Images from different positions. Which include some plants and weeds? The image also captures various factors like soil, sunlight, and color around the Image. It considers all the factors and the go into the detector in algorithm. The YOLO V2 checks every factor it considers and evaluated and finally it gives the detected image.

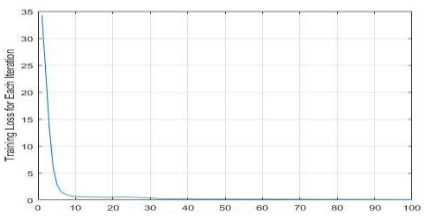


Input Image



Sample Detected Image

The above figure shows the detected image for the input image the plants are detected and showed the accuracy of the detected plants. Cleary the plants are detected in rectangular boxes other than the this we can consider and tell that the surroundings around the rectangular boxes is considered as weeds. The detected Image will be matched to our test data and results are verified with the reference of the Test data and same should be concluded.



Training Loss

The above figure shows the training loss for each iteration and the number of iterations. Here in the above figure, we can clearly see that the training loss decreases for every iteration and it becomes constant at particular iteration. The less the training loss, the more accurate the result. Generally training loss value should be considered below 1.

CONCLUSION

In this study, we proposed an approach to identify weeds in vegetable plantation using deep learning and image processing. The algorithm was depicted in two steps. A YOLO v2 model was trained to detect vegetables. The trained YOLO v2 achieved a very low in training loss and accuracy with nearly 80%.

FUTURE SCOPE

In this work, we are only willing to detect the plant but not the name of the species/plant. In future, using deep learning classification task we can classify the type/name of the plant/leaf.

REFERENCE

1. T. W. Berge, A. H. Aastveit, and H. Fykse, ``Evaluation of an algorithm for automatic detection of broad-leaved weeds in spring cereals," Precis. Agricult., vol. 9, no. 6, pp. 391405, Dec. 2008.

2. E. Hamuda, M. Glavin, and E. Jones, ``A survey of image processing techniques for plant extraction and segmentation in the eld," Comput. Electron. Agricult., vol. 125, pp. 184199, Jul. 2016.

3. H. Mennan, K. Jabran, B. H. Zandstra, and F. Pala, ``non-chemical weed management in vegetables by using cover crops: A review," Agronomy, vol. 10, no. 2, p. 257, Feb. 2020.

4. Y. Lai, F. Sun and H. Liu, "Small Object Detection Base on YOLOv3 For Pedestrian Recognition," 2020 5th International Conference on Control and Robotics Engineering (ICCRE), 2020, pp. 235-241, doi: 10.1109/ICCRE49379.2020.9096492.

5. A. J. Irías Tejeda and R. Castro Castro, "Algorithm of Weed Detection in Crops by Computational Vision," 2019 International Conference on Electronics, Communications and Computers (CONIELECOMP), 2019, pp. 124-128, doi: 10.1109/CONIELECOMP.2019.8673182.

6. [12] Y. LeCun, Y. Bengio, and G. Hinton, ``Deep learning," Nature, vol. 521, no. 7553, pp. 436444, May 2015.