International Journal of Research in Engineering, Science and Management
Volume 5, Issue 1, January 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

# Weed Identification using Regional Convolutional Neural Network

SriLakshmi Polavarapu*
*Student, M.Tech. Integrated Software Engineering, Vellore Institute of Technology, Chennai, India*

***Abstract*: In ancient days most of the people selected farming as their occupation. For that, people started to cultivate crops in their fields. But there were days where people had to fight against the invasion of weeds in their fields. Weeds that are unwanted plants present around a good plant. To overcome this problem people started to remove the weeds manually by cutting them very close to the ground. Later when days passed by people started using pesticides and some form of chemicals to cut them off. As the technology has developed and the process of removing weed by using the above practices were very much time consuming. Usage of chemicals for controlling unwanted plants has become a con for the crops, as it contains some harmful side effects when sprayed on a healthy plant. Due to this many people have moved to an advanced technique of deep learning. Convolution Neural Network have become popular and also a good way to remove weeds.**

***Keywords*: Convolution Neural Network, Image Data Generator, Regional Convolution Neural Network, fine tuning, Support Vector Machine, VGG16.**

## 1. Introduction

Agriculture is the preferred occupation done by many people. Manual labor done for removing weeds is very time consuming, weeds mainly affect the growth of a healthy plant and compete for nutrients present in the plant, for water and sunlight that is the reason the quality of crop is reduced. Some types of weeds act as pathogens and cause plant diseases which result in decrease of grain yield in the plant. An advanced technique to remove weeds may be more feasible. Use of weed detection techniques also helps us to limit the usage of pesticides in the fields. Many attempts and struggles are made to develop a Weed Identification Conventional Neural Network.

This paper uses Regional Convolution Neural Network also called RCNN to detect the weeds present in the fields. This project takes the images of the plants and identifies by drawing boundaries in the image. Mainly that image is divided into three classes namely crop, weed and background by which we can easily differentiate the class of image. To divide the image into classes, the project uses other techniques under RCNN.

## 2. Related Works

Before proceeding into the project, I have reviewed all the different techniques that are used for weed identification in the fields.

## 3. Objectives

The goal of this paper is to decrease the damage caused by weeds present in the fields. The main concentration here is to construct a model which can differentiate the good plant and an unwanted plant. Our main objective is to make free flow of water by removing unwanted plants. We can also make a good water transportation system by removing them.

## 4. Proposed Methodology

In this paper, I have implemented a Regional Convolution Neural Network. This RCNN is mainly divided into three techniques namely CNN Fine Tuning, CNN+SVM and Bounding Box Regression.

1. Priorily we perform data pre-processing by reading our dataset and dividing it into train and test images. Train data consists of 1000 images whereas test data consists of 300 images. Then for this data we apply Selective Search Algorithm for object detection then it generates 2000 images per region. Then the generated images are passed through Intersection Over Union (IOU). If IOU>0.5 then the image is saved in positive example but if IOU<0.2 then it is saved as negative example.
    a) Positive includes crop or weed.
    b) Negative includes background.
    c) Then this data is saved into three folders: crop, weed, and background for further purpose.

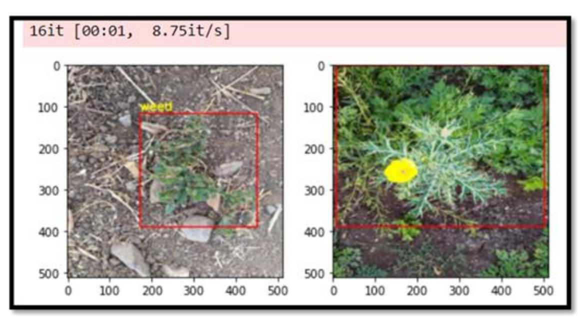


Fig. 1. Visualizing of data

*Corresponding author: srilupolavarapu05@gmail.com

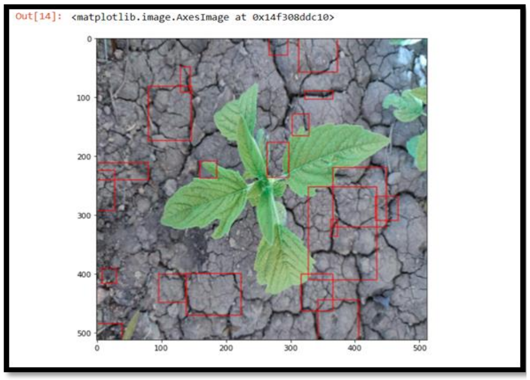


Fig. 2. Selective search algorithm

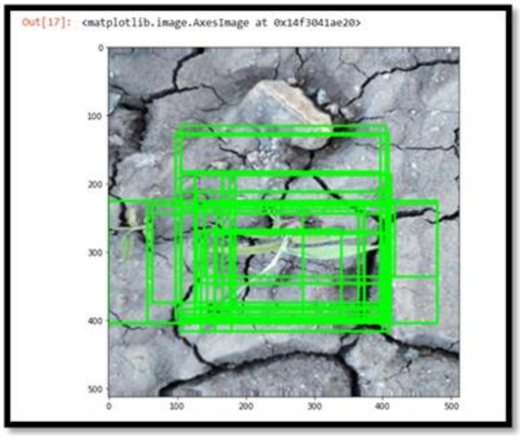


Fig. 3. IoU positive

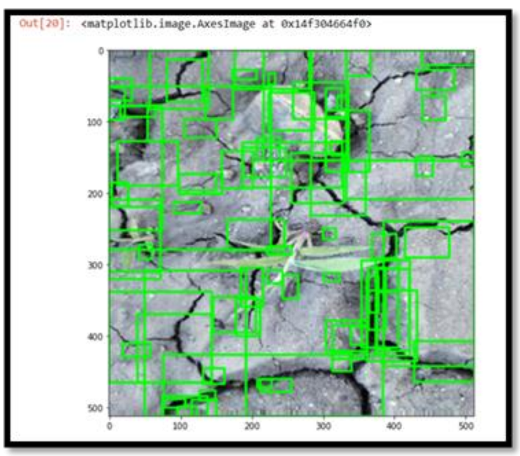


Fig. 4. IoU negative

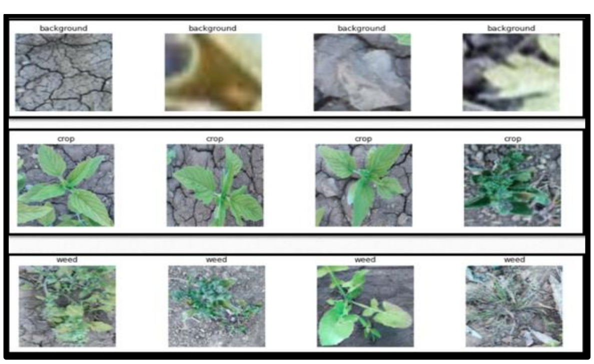


Fig. 5. Images of 3 classes: weed, crop and background

2. For CNN Fine Tuning which is one of the RCNN techniques, we use the above saved folders containing crop, weed, and background. By using Image Data Generator, which passes a new image at each and every iteration we will find the image which belongs to all the three classes: crop, weed and background. Now we construct the model architecture with VGG16, which is a CNN model. Using that model we calculate accuracy.
   - Accuracy = 97%

*CNN Fine Tuning:*

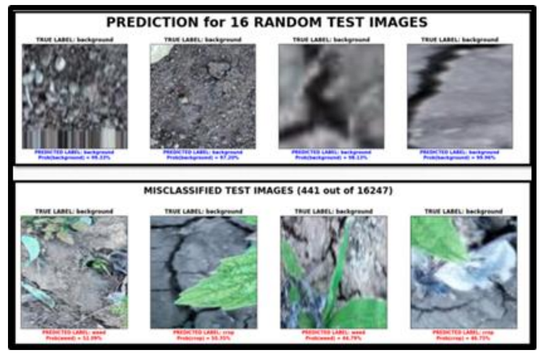


Fig. 6. Predicting random 16 images

In [8]: model.summary()

Model: "model"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_1 (InputLayer) | [(None, 224, 224, 3)] | 0 |
| block1_conv1 (Conv2D) | (None, 224, 224, 64) | 1792 |
| block1_conv2 (Conv2D) | (None, 224, 224, 64) | 36928 |
| block1_pool (MaxPooling2D) | (None, 112, 112, 64) | 0 |
| block2_conv1 (Conv2D) | (None, 112, 112, 128) | 73856 |
| block2_conv2 (Conv2D) | (None, 112, 112, 128) | 147584 |
| block2_pool (MaxPooling2D) | (None, 56, 56, 128) | 0 |
| block3_conv1 (Conv2D) | (None, 56, 56, 256) | 295168 |
| block3_conv2 (Conv2D) | (None, 56, 56, 256) | 590080 |
| block3_conv3 (Conv2D) | (None, 56, 56, 256) | 590080 |
| block3_pool (MaxPooling2D) | (None, 28, 28, 256) | 0 |
| block4_conv1 (Conv2D) | (None, 28, 28, 512) | 1180160 |
| block4_conv2 (Conv2D) | (None, 28, 28, 512) | 2359808 |
| block4_conv3 (Conv2D) | (None, 28, 28, 512) | 2359808 |
| block4_pool (MaxPooling2D) | (None, 14, 14, 512) | 0 |
| block5_conv1 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_conv2 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_conv3 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_pool (MaxPooling2D) | (None, 7, 7, 512) | 0 |
| flatten (Flatten) | (None, 25088) | 0 |
| dense (Dense) | (None, 4096) | 102764544 |
| dropout (Dropout) | (None, 4096) | 0 |
| dense_1 (Dense) | (None, 4096) | 16781312 |
| dropout_1 (Dropout) | (None, 4096) | 0 |
| dense_2 (Dense) | (None, 3) | 12291 |

Total params: 134,272,835
Trainable params: 119,558,147
Non-trainable params: 14,714,688

Fig. 7. VGG16 model architecture

After calculating the accuracy some of the random 16 images from the dataset are taken and classified as the true and predicted label.

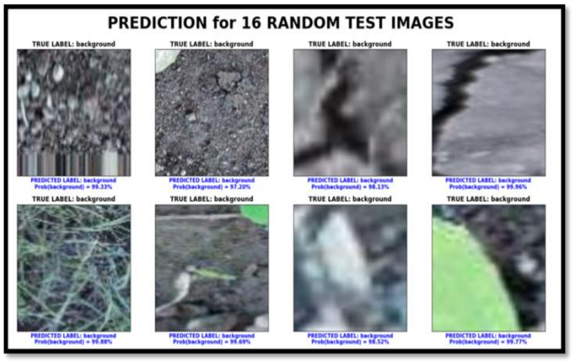


Fig. 8. Prediction of 16 random test images

Same as the first point, some random images are mis-classified. Mis-classified images are nothing but actually they are predicted as crop or weed but they might be background.

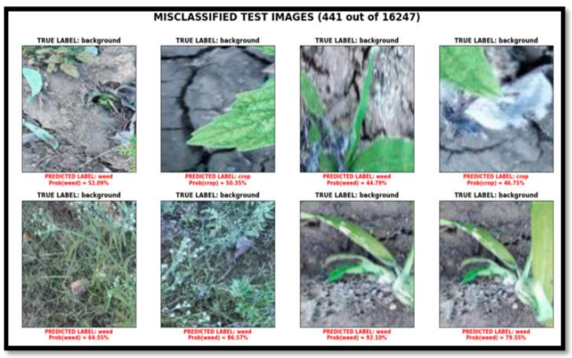


Fig. 9. Misclassified test images (441 out of 16247)

**loading model without last two Fully connected layers**

```
In [10]: model_without_last_2FC = tf.keras.models.Model(model.inputs,model.layers[-5].output)

In [11]: model_without_last_2FC.summary()

Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 4096)              102764544
=================================================================
Total params: 117,479,232
Trainable params: 102,764,544
Non-trainable params: 14,714,688
_________________________________________________________________
```

Fig. 10. Loading model without last two fully connected layers

3. For CNN+SVM which is another RCNN technique. Here mainly the mis-classified images in CNN Fine Tuning are classified. For CNN+SVM, we will extract the images from ground truth labeled images which may be crop or weed and also, we will extract images from negative examples. Then we will prepare the data for SVM using feature vectors and we will find the accuracy of our model. We will check for an image randomly and we will find the accuracy of the image which gives the accuracy of all the three classes (crop or weed or background).
4. Finally, for performing detection, we will load the previously trained model and again we will calculate IoU. Then, we will take one image for performing detection and we will find the output.

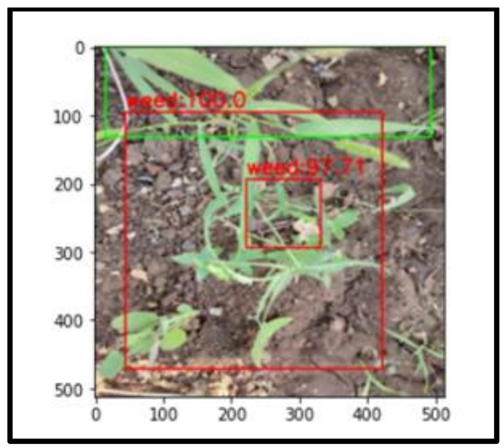


Fig. 11.

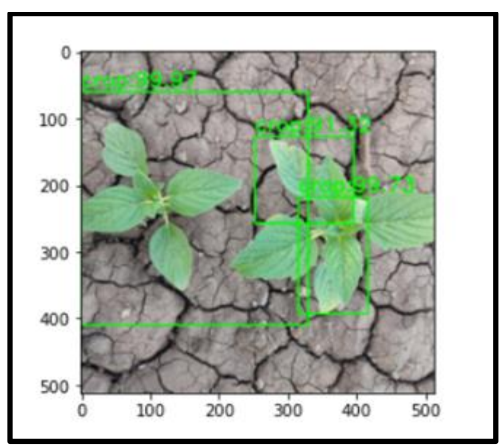


Fig. 12.

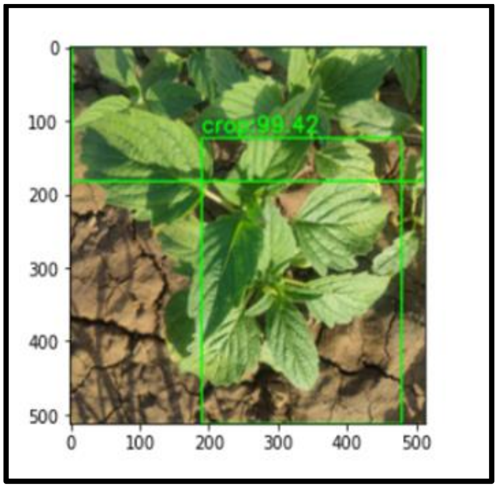


Fig. 13.

## 5. Architecture Model

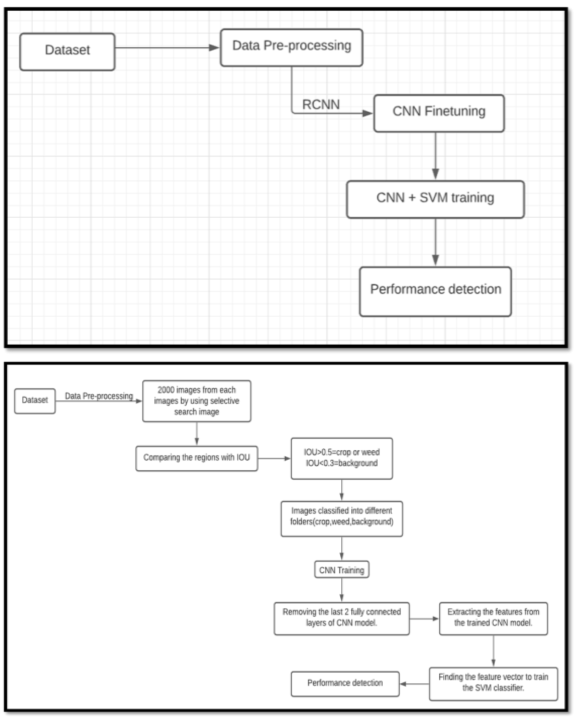


Fig. 14. Architecture model

## 6. Experiments Conducted

This project is conducted using the Anaconda Jupyter Notebook. I have implemented it in a Windows 11 PC which has 12GB RAM with CPU of Intel I5 10th gen processor and 8GB quad core processor.

## 7. Literature Survey

In [1] Richy Johnson, Thomas Mohan, Sara paul implemented the paper weed detection and removal based on image processing. Here the identification of weed and removal of weed are performed through image processing. The system focuses on reducing human labor and also reduces time for identifying and removing weed.

In [2] Sadia Murawat, Amish Quershi, Saleha Ahmad, Yousaira Shahid implemented this paper using SVM. In this paper some features like color are used for distinguishing crop and weed.

In [3] Arvinth S, Balakrishnan A, Harikrishnan M, Jeydheepan J implemented a Convolutional Neural Network. This system identifies the weed with the help of CNN and identifies the weed in the picture by utilizing neural organization.

## 8. Future Works

I will work to provide a Boundary Box Regression Model. Boundary Box Regression is the most popular technique to predict boxes or boundaries around an image to identify if it is a weed or crop or background.

## 9. Conclusion

Though many works are done related to weed identification using many machine learning models and artificial intelligence they had problems in identifying the image and detecting the weed. This paper uses the Regional Convolution Neural Network which helps me to detect the weed by using an image data generator. The use of VGG16 model in CNN Fine Tuning which has given better accuracy and reduced overfitting.

By this I conclude that RCNN gives better results compared to any other model for identification of weed.

## References

[1] Rincy Johnson, Thomas Mohan, Sara Paul, "Weed Detection and Removal based on Image Processing" International Journal of Recent Technology and Engineering, Volume 8, No. 6, March 2020.

[2] Sadia Murawat, Amish Quershi, Saleha Ahmad, Yousaira Shahid "Weed Detection Using SVM's" Engineering Technology & Applied Science Research, Volume 8, No. 1, 2018.

[3] Arvinth S, Balakrishnan A, Harikrishnan M, Jeydheepan J, "Weed Detection Using Convolution Neural Network" International Research Journal of Modernization in Engineering Technology and Science, Volume 3 No. 3, March 2021.