Journal of Cases on Information Technology
Volume 24 • Issue 4

Cotton Leaf Disease Detection Using Instance Segmentation

Prashant Udawant, Mukesh Patel School of Technology Management and Engineering, SVKM's NMIMS, Mumbai, India*
https://orcid.org/0000-0002-1383-355X

Pravin Srinath, Mukesh Patel School of Technology Management and Engineering, SVKM's NMIMS, Mumbai, India

ABSTRACT

Cotton is one of the most important cash and fiber crops in India. Agricultural machine learning plays a very important role in this agricultural industry. In this paper, the use of an object detection algorithm, namely Mask RCNN, along with transfer learning is used to find out if it is a fit algorithm to detect cotton leaf diseases in practical situations. The model training accuracy is found to be 94% whereas total loss value is continuously decreasing as the number of optimized iterations are increasing.

KEYWORDS

Detectron 2, Instance Segmentation, Mask RCNN, Object Detection, Transfer Learning

1. INTRODUCTION

Plant diseases have a substantial influence on agriculture since they cause crop quality and hence productivity to decrease. When it comes to cotton, one of India's most significant crops, 40-50 million people rely on it for their livelihoods, either directly or indirectly. It has several applications, including cotton textiles, paper, clothing, and many more. Cotton's great vulnerability to illnesses and pests is, nevertheless, one of the most significant limitations in its production.

Disease detection in plants is one of the many applications of modern technologies in agriculture. And this certainly can help boost production as early detection of diseases is necessary so that the yields do not go to waste. A lot of research is going on Artificial Neural Networks (ANN's), Convolutional Neural Networks (CNN), and object detection algorithms to find a feasible solution to reduce crop failure. In this paper, Object detection is focused to get accurate results of the type of diseases present on cotton leaves.

Previous results have shown the use of controlled environment images to train the models made it practically impossible to use since they give very good accuracy in only controlled environment images and received very bad accuracy with images taken under different conditions. Hence, this paper deals with images with different backgrounds, having multiple cotton leaves so that it is practically feasible.

The goal of this study is to find if Mask RCNN is a good fit algorithm to detect diseases and pests on the cotton leaves.

DOI: 10.4018/JCIT.296721
*Corresponding Author

2. LITERATURE SURVEY

To understand what different methods were used to identifying diseases in different plants, a summary of the different papers need to be understood. On the Plant Village dataset, the authors (Tm, P., Pranathi *et al*., 2018) developed a method to categorize sick tomato leaves into ten distinct groups. Data gathering, data pre-processing, and classification are the three phases in their suggested technique. Because the dataset contained low-noise pictures, noise reduction was not necessary for the pre-processing stage, and the images were scaled to 60×60 resolution. The Z-score technique, which is the mean and standard deviation, was used to normalize the pictures. They utilized Convolutional Neural Networks (CNNs) using several deep learning architectures such as LeNet, AlexNet, and GoogleNet for classification. The LeNet architecture produced the greatest results, with an accuracy of 94.8 percent. However, because the pictures do not have any noisy backgrounds, this approach is computationally intensive, and its practical practicality is debatable.

In 2019 (Wang, Q. *et al*., 2019) researchers merged three deep convolutional neural networks, VGG169, ResNet50, and ResNet101, which are commonly used for extracting features, with a Faster RCNN structure to diagnose tomato illnesses. The entire procedure is broken down into four steps: The Deep Convolutional Neural Networks (DCNN) is the first, and it is used to extract feature maps from input pictures. The (Region Proposal Networks) RPN is generated in the second phase. The ROI (Region of interest) pooling stage involves RPN obtaining a proposed feature map of fixed size and DCNN obtaining the final feature map, which is then fed into the complete connection layer at the back for target detection and placement. The feature map is then fed into the complete connection layer, with the SoftMax layer utilized to precisely categorize the input. Simultaneously, the boundary box regression procedure is performed to determine the exact position of the tomato fruit. resnet101 is proven to have the best overall performance. However, due to the limited sample size utilized for training, the model in the article has a low detection accuracy. As a result, work on upgrading the network structure is required.

Prasad proposed (Prasad, S. *et al*., 2014) that the disease be controlled and damage reduced by measuring plant leaf damage levels and thereby forecasting plant damage levels. The first stage is Leaf acquisition, which necessitates a constant consistent background. The disadvantage of the recommended approach is that it necessitates the use of basic background and the cleaning of the leaf to remove dust particles, sand, or dirt to avoid mispredictions. The second step is divided into many parts, the first of which is Frame selection, in which entropy is used to determine the image's statistical unpredictability using the Gray image. Using the unsupervised clustering technique L*a*b* colour space model-based k-means leaf disease segmentation methodology, this work was created for basic camera-based mobile devices and categorised the pictures into three clusters: background, diseased, and non-diseased.

(Prasad, S. *et al*., 2013) Employed the k-mean approach to find homogeneously textured clusters in unsupervised L*a*b colour texture-based picture segmentation, which has a cheap processing cost and a high accuracy rate with little changes. A mobile vision system's colour texture is the best attribute for grouping leaf disease gathered in a continuous backdrop. Unsupervised colour picture segmentation recovers homogeneous regions from an image using low-level information like as intensity or texture characteristics. It's known as a bottom-up segmentation method. Before k-mean clustering, the images are averaged. And three of the clusters are the same as the ones mentioned in the preceding paragraph. The approach has a high level of accuracy, and the whole turnaround time on a mobile device is about 1 second (best case). This technique has the problem of requiring a clean background, which is difficult for a farmer to capture.

Based on biological growth of cotton plants, it affects the collection of cotton leaf of different classes in same region. So in primary level healthy and unhealthy identification of leaf is done in this paper (Udawant, P. *et al*., 2019). Transfer learning technique can be used to identify the region of diseased portion. Based on the affected region the percentage of healthy and unhealthy diseased portion has been identified (Udawant, P. *et al*., 2021).

The authors (Warne, P. P. *et al*., 2015) developed a technique for identifying cotton diseases with an SVM classifier by collecting images, pre-processing them by filtering noisy data, and segmenting the image using k-means clustering with colour as the segmenting characteristic by assessing the L*a*b color-space. The classifier has an accuracy rate of 98 percent.

(Teng, C. *et al*., 2011) Segmented and classified pictures of plants in the field in 2011. It also estimates and reconstructs three-dimensional (3D) plant information. For 3D reconstruction, the system requires more than one picture of the plant. In addition, the segmented leaves must be in the image's center. This method is complicated, and farmers are unable to use it effectively.

The authors (Rehman, Z *et al*., 2021) presented a system for detecting and classifying apple fruit diseases. A hybrid contrast enhancement approach is used to improve the image quality in the beginning. Then they're sent to Mask RCNN for classification, as well as Resnet-50 retraining. The data in this example came from Plant Village, and a selection strategy based on Kapur's entropy and an MSVM approach was created to acquire the features for the final classification.

The authors (Liu, B. *et al*., 2018) presented a unique identification technique based on DCNN to identify apple leaf diseases. Image rotation and Principal component analysis jittering were used to increase inadequate pictures for training (augmentation). GoogleLeNet's inception is used to boost the feature extraction capabilities of a new CNN built on AlexNet. The accuracy attained in the experiment was 97.62 percent.

3. PROPOSED METHODOLOGY

3.1 Dataset Collection

One of the most significant shortcomings of current plant disease detection is the classification part, where leaves with multiple diseases cannot be identified because the training was done on only controlled environment images, which means that all images were taken as a single leaf with solid non-confusing backgrounds and only one disease. When it comes to the practical method to identifying illnesses on the leaf, they stop working effectively due to the reasons described above, as well as a lack of excellent plant leaf datasets, with the PlantVillage dataset accounting for the majority of current successes.

To make this a reality, a high-quality camera smartphone was used to capture about 2000 photos of cotton leaves from several farms in Maharashtra with various backdrops. This dataset contains pictures of leaves affected by various diseases and pests, with numerous leaves in the background, which might represent real-world scenarios in which this application could be utilized. Experts classified and confirmed the illnesses depicted in the photographs.

3.2 Data Annotation

The data annotation was completed with the aid of a program called LabelImg, which allows you to create bounding boxes around sick areas and pests, and the annotations were stored in Pascal VOC format (XML annotations).

3.3 Augmentation

Traditional Augmentation techniques were used to augment the images containing only the Jassid Attack and Jassid Attack Effect because the number of instances of these was very less as compared to other diseases which would affect the model training to incorrectly detect diseases. Various kinds of transformations like mirroring, rotating, and shifting were applied to the training dataset.

3.4 Detection of Object

Deep Neural Network-based object identification algorithms were used to identify many diseases in a single leaf when the backgrounds of the photos were complex. Object detector techniques are

Figure 1. Bounding boxes on diseased cotton leaf parts using Label-Img tool

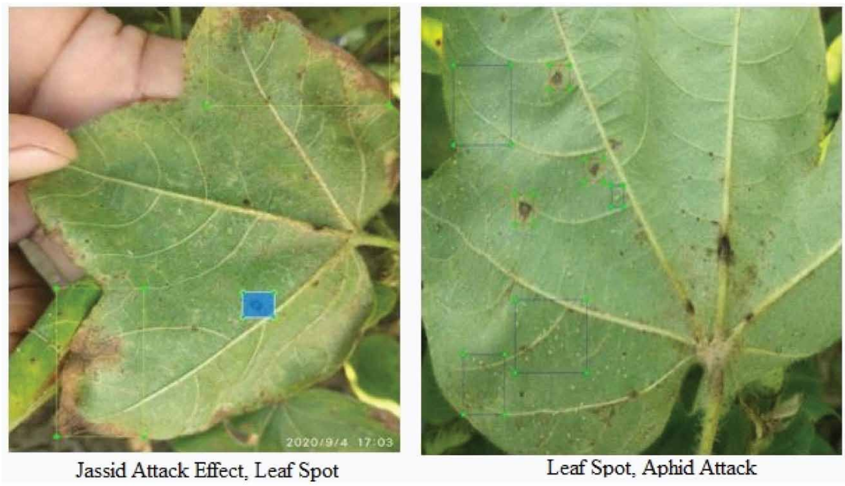


Table 1. Table consisting of the classes(diseases) and the number of bounding boxes per class

Class Number	Class Name (Disease Name)	Number of Bounding Boxes
1.	Aphid Attack	2763
2.	Jassid Attack	1333
3.	Whiteflies	2403
4.	Leaf Spot	2849
5.	Jassid Attack Effect	1615

divided into two categories: single-stage and two-stage object detectors. Object detection is treated as a simple regression problem using single state detectors, which learn the class probabilities and bounding box coordinates from an input image. Region Proposal Networks (RPN) are used in the first stage of two-stage object detectors to establish regions of interest and provide region recommendations for object classification and bounding-box regression.

Two state-of-the-art algorithms, Single Shot Detector (SSD)(single-stage) and Mask RCNN (He, K. *et al*., 2017) (two-stage), were used on the dataset, which was split into 70 percent for training, 15 percent for testing, and 15 percent for validation. Over the course of 30,000 iterations, the SSD model was trained using a batch size of 32 and a learning rate of 0.0001. The model was pre-trained on the COCO dataset, and the backbone architecture was Resnet101. This object detection model was created from the ground up. The bounding box values X-min, Y-min, X-max, Y-max, and labels were first transformed from the XML annotations into a single CSV file. The file was then transformed to TFrecords, a format that combines labels, pixel values, and bounding box coordinates into a single package.

The SSD model's results aren't up to par, and even after 30,000 cycles of training, the diseases are barely discernible. The loss created at the end of the training was around 0.4, and it remained consistent even as the number of steps increased.

Detectron 2 is the next-generation program that incorporates state-of-the-art object recognition algorithms rather than starting from scratch with the Mask RCNN model. It is based on the Mask RCNN benchmark and is a rework of Detectron. Detectron 2 supports object detection, semantic segmentation, instance segmentation, panoptic segmentation, and keypoint detection. The model was trained using a batch size of 64 and a learning rate of 0.01 for the first 500 iterations, then reduced to 0.001 and trained for another 42,000 iterations to obtain a loss of 0.03. The backbone architecture used was Resnet-50. The architecture was pre-trained on the COCO dataset, allowing for a considerably faster and more efficient outcome. Model weights were frozen and preserved for future usage after the training. The entire project was completed with Google Colab and an Nvidia GPU Tesla T4.

Because it is the most commonly used metric in object detection, the mean average precision was chosen as the performance metric. The value for you (Intersection over Union) was set to 0.5. IoU is a numerical metric for determining how closely ground truth and projected boxes match as shown in equation 1:

$$IoU = \frac{Intersection\ \ area}{Union\ \ area} \tag{1}$$

As a result, regions with an IoU score of more than 0.5 are True Positive (TP, detecting needed items), while regions with a score less than 0.5 are False Positive (FP). This number ranges from 0 to 1, with 1 being the best result. Class c's Average Precision is derived as indicated in equation 2:

$$AP = \frac{\#TP\left(c\right)}{\#TP\left(c\right) + \#FP\left(c\right)} \tag{2}$$

Figure 2. Flowchart for the Mask RCNN model

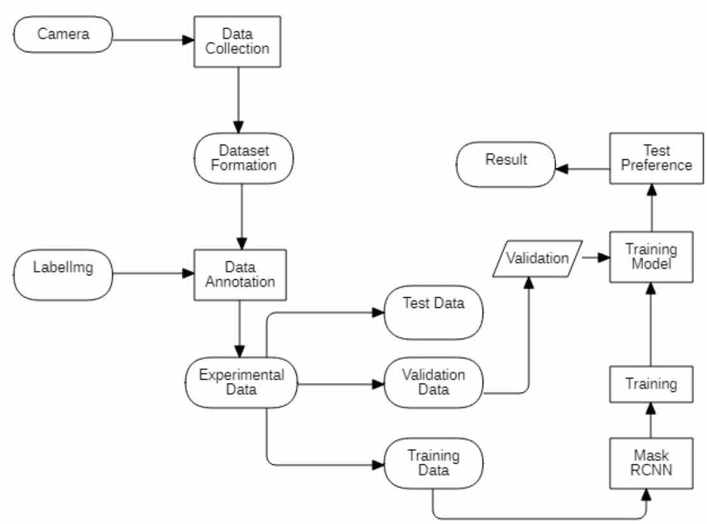


Table 2. Object Detection Accuracy

Type of Object Detection	Algorithm	Backbone Architecture	Accuracy
Two-Stage	Mask RCNN	Resnet-50	94

The mean of the Average Precision of all classes is the Mean Average Precision (MAP) which is calculated as shown in equation 3:

$$mAP = \left(\frac{1}{n}\right) * \sum_{c=1}^{c=n} AP\left(c\right) \tag{3}$$

- **n:** The number of classes.
- **AP(c):** The AP of class c.

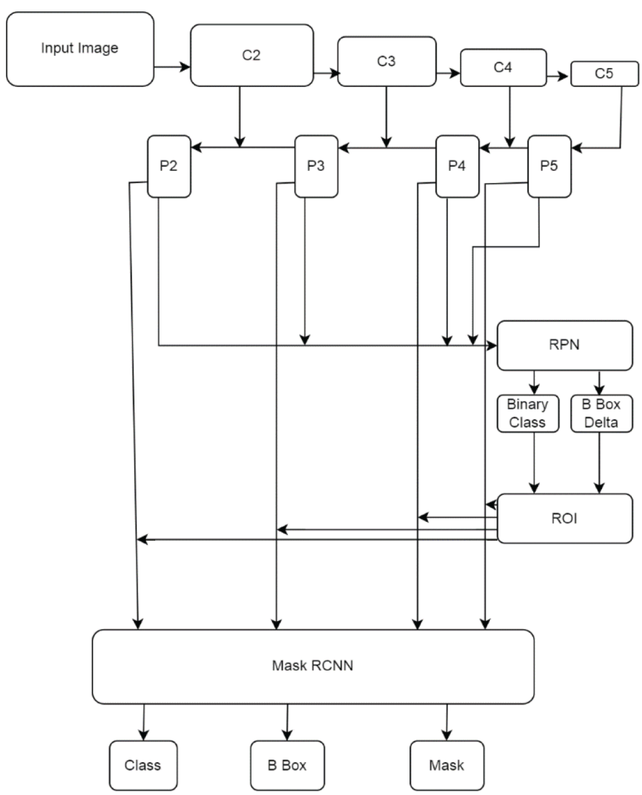
Table 2 shows that Mask RCNN accuracy. In general, two-stage methods reach great accuracy rates than single-stage methods, but two-stage methods are slower.

4. ARCHITECTURAL DESIGN

Mask RCNN is a Faster RCNN (Ren, S. *et al*., 2016) extension. Faster RCNN delivers the bounding box coordinates and class labels for each object in a given image, whereas Mask RCNN also returns object masks in addition to bounding box coordinates and class names.

The ResNet50 architecture is utilized in this study to extract features from pictures in Mask RCNN, which is then applied to a Region Proposal Network (RPN), which predicts whether or not an object is present in the region, as shown in figure 3. Stage 2 involves transmitting these regions

Figure 3. Architectural overview of Mask R-CNN



across a fully connected network layer to anticipate the class label and boundary boxes. After that, the IoUs are calculated. Then, to the existing architecture, a mask branch can be added, yielding a segmentation mask for each object-containing region (figure 4).

5. RESULTS

All of the photographs in figure 5 are the outcomes of the testing. They depict pests like aphids, jassids, and whiteflies, as well as illnesses such as Cotton Leaf Spot and Jassid Attack Effect. These are the after effect on the leaves after sucking pests, Jassid.

Figure 4. Mask RCNN for instance segmentation

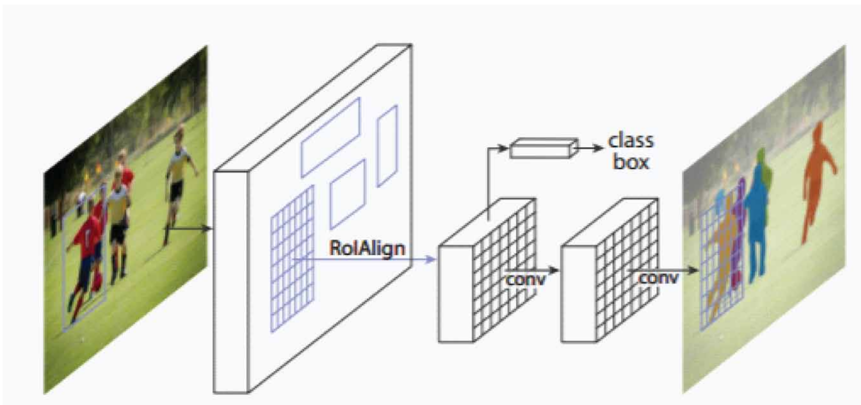
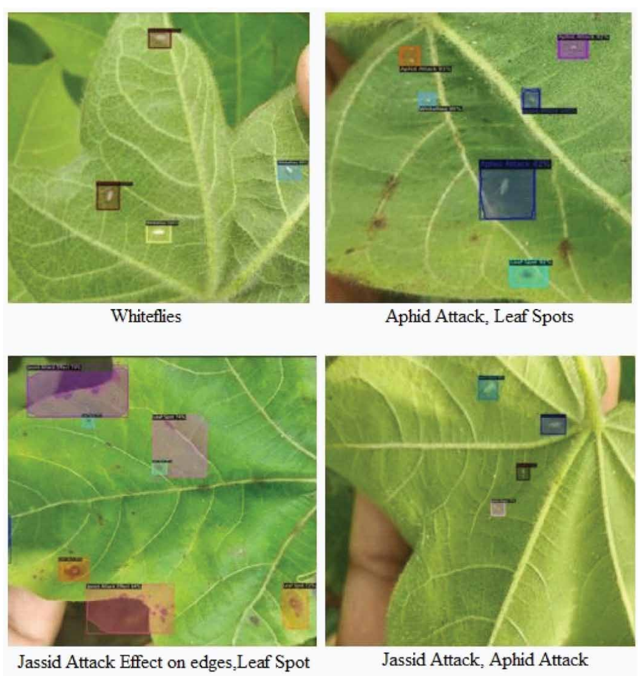


Figure 5. Showing the detection of pests and diseases on leaves by Mask R-CNN



An accuracy metric is used to interpretably measure the algorithm's performance. The accuracy of a model is usually calculated as a percentage after the model parameters have been defined. It's a metric for determining how close the model's forecast is to the actual data. The model's training accuracy is around 94 percent, as illustrated in figure 6. Out of the 200 photos tested, 180 photographs were found to have the proper diseases present, as well as high confidence levels (above 0.5).

When the confidence score of detection that is anticipated to identify a ground-truth is less than the threshold (in example, threshold = 0.5), the detection counts as a false negative. (FN). As the number of training steps grew, the false-negative curve reduced (see Figure 7). As a result, the model was well-trained to be able to detect the object with high confidence.

Only if three requirements are met, then a detection called true positive (TP): confidence score > threshold; predicted class matches ground truth class; predicted bounding box has an IoU larger than a threshold (instance threshold value= 0.5) with ground truth. A false positive occurs when one of the last two conditions is violated (FP). As training progressed, the false-positive curve dropped, allowing the object to be correctly identified with a high confidence score (figure 8).

The loss value indicates how well or poorly a model performs after each optimization iteration. A loss, unlike accuracy, is not a percentage, as illustrated in figure 9. It's the total number of mistakes committed in each training or validation set for each example. After 40k cycles, the loss has decreased to around 0.3.

Figure 6. Mask RCNN accuracy

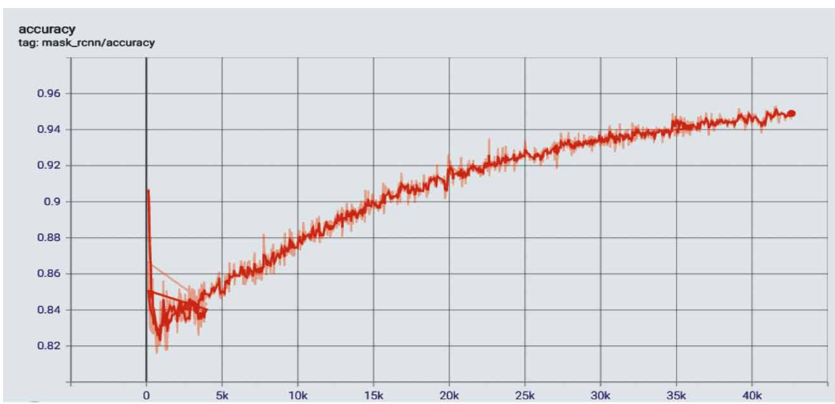


Figure 7. False Negatives while training

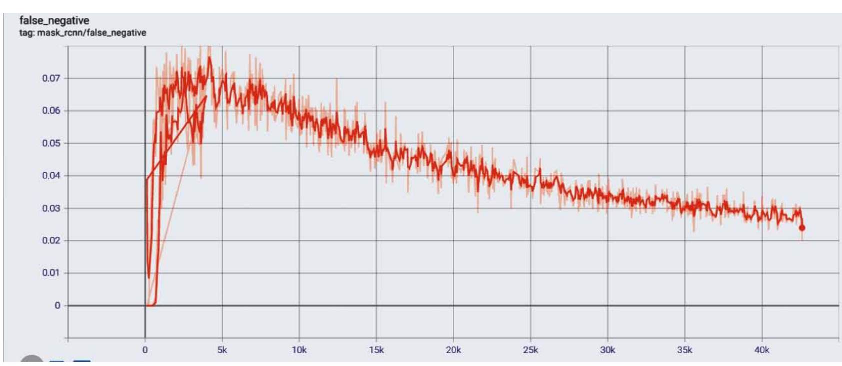


Figure 8. False Positives while training

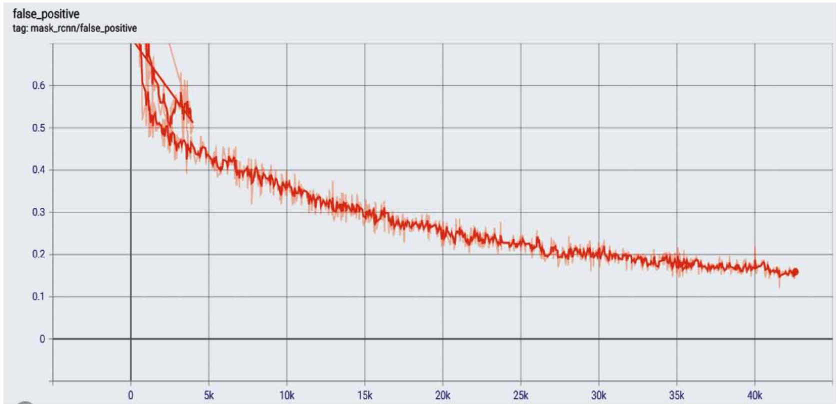
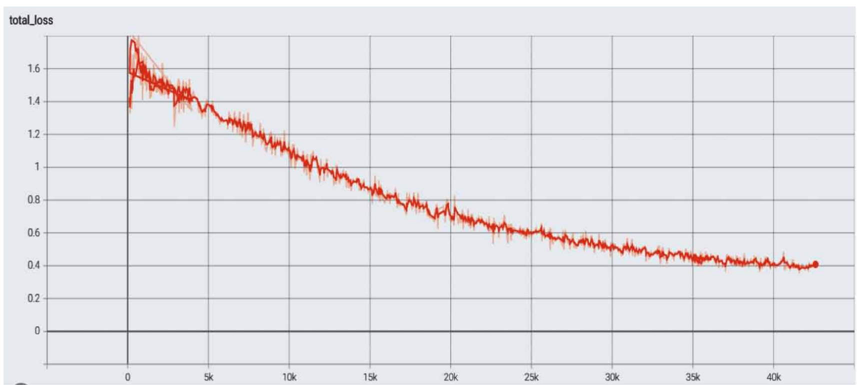


Figure 9. Total Loss



6. CONCLUSION AND FUTURE WORK

Detecting the minuscule-sized pests on leaves and also identifying the diseased portion of the leaves is a big challenge. Transfer learning was used to develop the model to detect pests and diseases on cotton leaves irrespective of background very easily and with good precision. The result of the model detects the name of the disease or pest present on the diseased region and it has been correctly identified. For future work, more images have to be collected to include more number diseases along with characteristics like soil type, temperature, while also using GANs (Goodfellow, I *et al.*, 2014) for augmentation to improve data for training.

REFERENCES

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. et al. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969). IEEE.

Liu, B., Zhang, Y., He, D., & Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, *10*(1), 11. doi:10.3390/sym10010011

Prasad, S., Peddoju, S. K., & Ghosh, D. (2013, October). Unsupervised resolution independent based natural plant leaf disease segmentation approach for mobile devices. In *Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop* (pp. 1-4). doi:10.1145/2528228.2528240

Prasad, S., Peddoju, S. K., & Ghosh, D. (2014, September). Mobile mixed reality based damage level estimation of diseased plant leaf. In *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies* (pp. 72-77). IEEE. doi:10.1109/ITME.2019.00176

Rehman, Z. U., Khan, M. A., Ahmed, F., Damaševičius, R., Naqvi, S. R., Nisar, W., & Javed, K. (2021). Recognizing apple leaf diseases using a novel parallel real-time processing framework based on MASK RCNN and transfer learning: An application for smart agriculture. *IET Image Processing*, *15*(10), 2157–2168. doi:10.1049/ipr2.12183

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. doi:10.1109/TPAMI.2016.2577031 PMID:27295650

Teng, C. H., Kuo, Y. T., & Chen, Y. S. (2011). Leaf segmentation, classification, and three-dimensional recovery from a few images with close viewpoints. *Optical Engineering (Redondo Beach, Calif.)*, *50*(3).

Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., & Koolagudi, S. G. (2018, August). Tomato leaf disease detection using convolutional neural networks. In *2018 eleventh international conference on contemporary computing (IC3)* (pp. 1-5). IEEE.

Udawant, P., & Srinath, P. (2019). Diseased portion classification & recognition of cotton plants using convolution neural networks. *International Journal of Engineering and Advanced Technology*, *8*(6).

Udawant, P., & Srinath, P. (2021). Leaf Diagnosis Using Transfer Learning. In *Applied Information Processing Systems* (pp. 235–246). Springer.

Wang, Q., & Qi, F. (2019, August). Tomato diseases recognition based on faster RCNN. In *2019 10th International Conference on Information Technology in Medicine and Education (ITME)* (pp. 772-776). IEEE. doi:10.1109/ITME.2019.00176

Warne, P. P., & Ganorkar, S. R. (2015). Detection of diseases on cotton leaves using K-mean clustering method. *International Research Journal of Engineering and Technology, 2*(4), 425-431.