Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations

Conference Paper · May 2019

DOI: 10.1117/12.2518868

3 authors, including:

Prakruti Bhatt
Tata Consultancy Services Limited
12 PUBLICATIONS **13** CITATIONS

SEE PROFILE

PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations

Prakruti V. Bhatt, Sanat Sarangi, Srinivasu Pappula

Prakruti V. Bhatt, Sanat Sarangi, Srinivasu Pappula, "Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations," Proc. SPIE 11008, Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV, 1100808 (14 May 2019); doi: 10.1117/12.2518868

SPIE.

Event: SPIE Defense + Commercial Sensing, 2019, Baltimore, Maryland, United States

Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations

Prakruti V. Bhatt, Sanat Sarangi, and Srinivasu Pappula

TCS Digital Farming Initiatives, Research and Innovation, Mumbai, India

ABSTRACT

Timely and accurate recognition of health conditions in crops helps to perform necessary treatment for the plants. Automatically localizing these conditions in an image helps in estimating their spread and severity, thus saving on precious resources. Automated disease detection involving recognition as well as localization helps in identifying multiple diseases from one image and can be a small step forward for robotic farm surveying and spraying. Recent developments in Deep Neural Networks have drastically improved the localization and identification accuracy of objects. We leverage the neural network based method to perform accurate and fast detection of the diseases and pests in tea leaves. With a goal to identify an accurate yet efficient detector in terms of speed and memory, we evaluate various feature extraction networks and detection architectures. The images used to train and evaluate the models are with different resolutions, quality, brightness and focus as they are captured with mobile phones having different cameras through a participatory sensing approach. The experimental results show that the detection system effectively identifies and locates the health condition on the tea leaves in a complex background and with occlusion. We have evaluated YOLO based detection methods with different feature extraction architectures. Detection using YOLOv3 achieves mAP of about 86% with 50% IOU while making the system usable in real time.

Keywords: crop disease detection, digital farming, single shot detection, participatory sensing

1. INTRODUCTION

Out of many factors affecting crops, spread of diseases and pests is a major threat and needs real-time attention to save the crop. Automated and accurate identification of the diseases and pests, spraying on only the pest affected area and smart cropping of the detected area to estimate severity of the disease or pest in the farm can help in reducing costs and increasing yield. Recently, efforts have have been made to provide information for disease diagnosis online or on the edge devices like mobile phones with an aim to provide personalized advice to farmers based on the reported events. So, as a crucial step towards identifying and curbing disease/pest spread, we have distributed a field scouting mobile application through which farmers submit the images of diseased plants. The images collected by field scouting (a.k.a participatory sensing) which usually involves more than one person, come with added challenges of uncontrolled conditions like varying backgrounds, resolution, illumination, camera angle, occlusion and image quality. It is required to recognize and localize diseases or pests in the images that can have parts of plants affected with one or more conditions in the same image. Research on detection of diseases has been carried out for a long time and there are several techniques to identify plant properties as well as stress due to different conditions. These methods include laboratory based chemical analysis of the infected area of the plant, analysis on visible light, and assessment of hyperspectal or multispectral images.[1] However, if analysis on images taken using a normal (RGB) camera gives accurate results, it makes the solution real-time, computationally efficient and low cost. Recent advances in detection and localization methods based on detector categories like Region-based Convolutional Neural Network (RCNN),[2] You Only Look Once (YOLO),[3] Region-based Fully Convolutional Network (R-FCN)[4] and Single Shot Multibox detector (SSD)[5] have proven useful to get highly accurate results on such images.[6] An automated system to identify, localize and measure the severity of the disease or pest in tea plant images submitted by various farmers is presented in this paper. The aim is to obtain an efficient and accurate method to detect and localize affected areas in any part of the plant in

Further author information:
E-mails: prakruti.bhatt@tcs.com, sanat.sarangi@tcs.com, srinivasu.p@tcs.com

Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV
edited by J. Alex Thomasson, Mac McKee, Robert J. Moorhead, Proc. of SPIE Vol. 11008,
1100808 · © 2019 SPIE · CCC code: 0277-786X/19/$18 · doi: 10.1117/12.2518868

images of different quality in terms of illumination, resolution and focus on the region of interest. The related work in precision agriculture for disease identification and object detection is presented in the next section. We then explain in sections 3 to 5, the basic idea of object detection using Convolutional Neural Networks (CNN), its evaluation and the single shot detection method applied by us. Later sections explain the dataset, training methods and evaluation of tea diseases and pests in terms of identification and localization accuracy as well as efficiency of all models.

2. RELATED WORK

Several image processing based techniques along with different classification methods have been applied to identify plant stress, pests and diseases. Visible-light range images in different color formats, satellite and airborne hyperspectral[7] as well as multispectral images[8] which include Near Infrared (NIR)[9] and fluorescence spectroscopy[10] are either analysed individually or in combination for the task. Recently, deep CNN based classifiers have been shown to be highly accurate especially with uncontrolled image data as they automatically identify the features imperative for recognition. Input to a CNN based classifier is a raw image and output is the probability that the image belongs to the considered classes. A plant detection system based on a small CNN to detect diseases in cucumber leaves have been shown to achieve an average of 82.3% accuracy under a 4-fold cross validation strategy.[11] Transfer learning has been used quite effectively to adapt deep convolutional neural networks to classify tomato leaves into 6 classes of diseased and healthy conditions[12] and to distinguish between healthy and unhealthy Cassava leaves.[13] Mohanty et. al[14] have performed supervised leaf disease classification with 99.35% accuracy by fine tuning the top layer and 98.36% by training from scratch the CNN models with a dataset taken in ideal conditions of background and illumination. Using CNN based deep neural networks for feature extraction, the detection tasks can be accurately performed at a low cost using in-field RGB images captured on mobile phones. While these methods have successfully classified the images into respective categories, we also aim to locate every recognized disease in the plant image for which we leverage the deep learning based detectors. Baweja et. al.[15] have collected image data of sorghum plants using stereo imagery and applied faster-RCNN with features extracted from VGG-16 in order to detect, count and calculate the width of the plant stalks. They have obtained one bounding box for each stalk and individually segmented the stalk in order to measure its width. I. Sa et. al.[16] have adapted Faster-RCNN on Color (RGB) and Near Infrared (NIR) images for detection of 7 kinds of fruits and while achieving an F1-score of 0.838 for sweet pepper detection.

Although disease recognition methods exist, accurate classification and localization of disease in images taken in uncontrolled environments can still be a challenge. Moreover, the system should also be efficient enough to give real-time inferences preferably on the edge devices. With this goal, we propose and evaluate detection performance of diseases and pests in RGB images of tea leaves captured under such conditions using CNN's.

3. CNN BASED OBJECT DETECTION

The state-of-art method of object detection, on a high level involves the following steps: (a) getting anchors or potential bounding boxes that might contain an object, (b) obtaining features of these proposals using convolutional neural network, (c) classifying the resulting features using a classifier model, and (d) localizing them using regression on the bounding boxes of the proposed regions.

Enhancements at every step of the detection have been made since the method of using deep learning for object detection - OverFeat[17] - was proposed. The convolutional neural network used for feature extraction is usually addressed as the *backbone* or a *base network* while the detection network except the feature extractor is termed as the *meta-architecture*.[18] The current detectors can be broadly divided into two categories based on the method to obtain features of anchor boxes. Detectors based on SSD and YOLO are single stage or single shot detectors as they predict the boxes and the classes of the objects simultaneously in them using the same convolutional neural network, while those based on RCNN and RFCN are called 2-stage detectors as they generate class-agnostic anchor boxes (region proposals) using a CNN usually called proposal stage and then use another fully connected or a convolutional network to classify those regions. For object detection, the current methods used widely apply an image classifier to identify an object in candidate or proposed locations of a test image. In the meta-architectures where region proposals are used, typically, there is a collection of boxes overlaid

on the image at different spatial locations, scales and aspect ratios, These boxes are addressed as *anchors*, or *priors* or *default boxes*. These detection models are trained to predict (a) class for the object in each anchor and (b) co-ordinates of the box or an offset by which the anchor needs to be shifted to fit the groundtruth bounding box. YOLO, a single shot detection method has been shown to perform with equivalent accuracy of other state-of-art detection methods on the COCO[19] dataset but at much higher speed. So in order to obtain accurate and real-time disease detection, we evaluate the YOLO based method along with different backbone (feature extraction) architectures for disease detection.

4. EVALUATION METRICS

Classification and localization performance of the detection method is evaluated based on the metrics (mean Average Precision) mAP and (Intersection Over Union) IOU. IOU measures how good the prediction of the bounding box with the object detector is with respect to the real or ground truth object boundary. It indicates the overlap between predicted bounding box and the ground truth bounding box as in Equation 1 where union implies the area encompassed by both the predicted bounding box and the ground-truth bounding box. The higher this value, the higher is the localization accuracy. A prediction of bounding box of a detected object is correct if it has a certain acceptable value of IOU.

$$IOU = \frac{Area\ of\ overlap}{Area\ of\ union} \tag{1}$$

mAP indicates whether the classified object is correct. It is the mean of the Average Precision over all classes making sure a method has precision at all levels of recall. Once a threshold for IOU between anchor and the ground truth is designated, for that value, the detection is marked as correct or incorrect. After getting the true positive and false positive values, the precise-recall curve is computed, and then average precision[20] is evaluated which is basically the same as average of summation over N maximum precision p values at all N recall $\overline{r}$ values exceeding r as in Equation 2.

$$mAP = \frac{1}{N}\sum_{r=0}^{1}(max(p(\overline{r})))\ such\ that,\ \overline{r} \geq r \tag{2}$$

5. REAL-TIME DETECTION : YOLO

YOLO is a fully convolutional network (FCN) that simultaneously predicts multiple bounding boxes and class probabilities for those boxes. The image is divided into grid cells and each grid cell proposes potential bounding boxes and scores those boxes using convolutional features. The J. Redmon et. al[3] have framed the detection as a regression problem where the output is box co-ordinates, objectness score or confidence and the class probabilities for all the anchors of each grid cell of the image as shown in Fig. 1. If the image is divided into $S \times S$ cells then for each grid cell, B bounding boxes are predicted with their 4 co-ordinates, 1 confidence score of objectness and C class probabilities. The predicted output can be encoded as $S \times S \times (B \times (4+1+C))$. The confidence scores reflect how confident the model is that the box contains an object and how accurate the predicted box is. It can be defined as $Pr(Object) \times IOU_{pred}^{truth}$. If no object exists in the grid cell, the confidence score should be zero. The maximum confidence should be same as intersection over union (IOU) between the predicted box and the ground truth.

YOLO has evolved since its three versions in terms of accuracy, architecture and the training methods for performance on benchmark datasets Pascal VOC and COCO. Since there are major improvements in accuracy and architecture between YOLO and the later versions YOLOv2 and YOLOv3, we have considered the later versions for our study. YOLOv2[21] is a fully convolutional network. Instead of pooling, a convolutional layer with stride 2 is used to downsample the feature maps. This helps in preventing loss of low-level features due to pooling. As it is an FCN, YOLO is invariant to the size of the input image. Batch normalization is used over all the convolutional layers in YOLOv2 as it leads to significant improvements in convergence while eliminating the need for other forms of regularization. Apart from regularization, adding batch normalization increased the

accuracy of it by 2% on COCO dataset. The authors have used input size of the image as $416 \times 416 \times 3$. This size has been chosen to get an odd number of locations in the resulting feature map so that there is a single center cell among the grids. Having a single location or a grid cell centre helps by increasing efficiency as large objects usually tend to occupy center of the image. In YOLO, convolutional layers downsample the image by a factor of 32 so by using an input image of 416 we get an output feature map of 13×13. For every grid cell, the region proposals or anchor boxes are considered which will be used to predict the co-ordinates of bounding boxes and classify objects within. This methods predict for over 1000 boxes with anchors for each image.

Since the network learns to adjust the bounding boxes based on the anchor boxes, selecting the anchors as the priors from where the network starts, the authors run K-means clustering on the dimensions of the training set bounding boxes. The aim of using the anchor boxes is to get higher IOU scores for the predicted bounding boxes so the distance measure is based on average IOU between the bounding box b and the closest centroid c as shown in Equation 3.

$$d(b, c) = 1 - IOU(b, c) \tag{3}$$

Authors have chosen k=5 anchors for each grid cell as a good trade-off between model complexity and high recall based on the annotations of VOC and COCO datasets. YOLOv2 predicts the bounding box coordinates relative to the location of the grid cell (top left corner of the grid). Logistic activation is used to constrain the predicted values between 0 and 1. The network predicts 5 coordinates for each bounding box, t_x, t_y, t_w, t_h, and t_o. If the cell is offset from the top left corner of the image by (c_x, c_y) and the anchor has width p_w and height p_h, then the predictions correspond to Equation 4. Fig. 1 shows the transform from anchor to the bounding box where centre co-ordinates (b_x, b_y) of bounding box are calculated by applying sigmoid to predicted values and adding the corner points of corresponding grid cell while the dimensions b_w and b_h of the bounding box are calculated by applying a log-space transform to the predicted output dimensions and then multiplying with an anchor dimensions.

$$\begin{aligned} b_x &= \sigma(t_x) + c_x, \\ b_y &= \sigma(t_y) + c_y, \\ b_w &= p_w e^{t_w}, \\ b_h &= p_h e^{t_h}, \\ Pr(object) \times IOU(box, object) &= \sigma(t_o) \end{aligned} \tag{4}$$

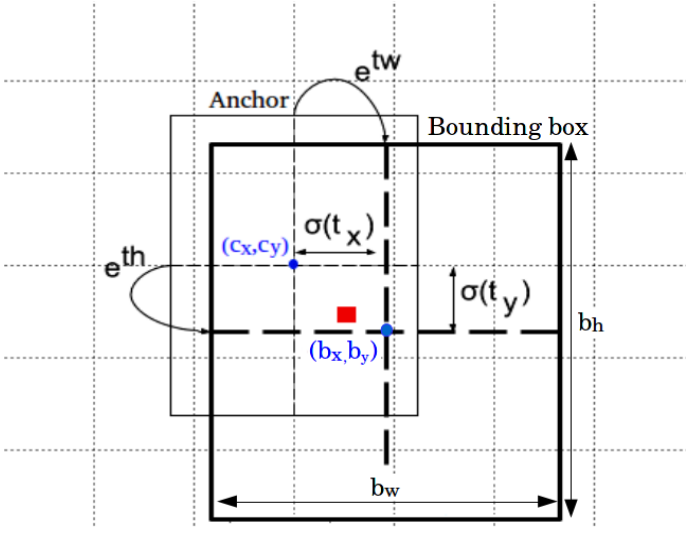


Figure 1. Co-ordinates of bounding box with respect to anchor and the grid cells

It can be seen that the feature map used in YOLOv2 is 13×13 as the convolution layer has a stride of 32. In order to localize smaller objects too, the authors add a passthrough layer that concatenates the features from an earlier layer at resolution of 26×26. Through the passthrough layer, higher resolution features are concatenated with the low resolution features by stacking adjacent features into different channels instead of spatial locations. This implies, the $26 \times 26 \times 512$ feature map becomes $13 \times 13 \times 1024$ which can easily be stacked with the original 13×13 features. The network can be resized to the dimension of the image we decide and continue training on the fly as the network is only based on convolutional and pooling layers. This means the same network can predict detections at different resolutions. YOLOv2 assigns only one bounding box for each ground truth object. If the bounding box proposals overlap a ground truth box by more than threshold of 0.5 IOU, it is dropped. The bounding box responsible for detecting the object will be the one whose anchor has the highest IOU with the ground truth box while training. Along with the meta-architecture of YOLOv2, the authors also propose a backend network with 19 convolutional layers and 5 maxpooling layers termed as "DarkNet-19". It requires 5.58 billion operations to process an image while achieving 72.9% top-1 accuracy and 91.2% top-5 accuracy on ImageNet. Apart from DarkNet-19, other backbone networks can also be used in YOLOv2 in order to get best feature representation and classification accuracy. Also, after the classification of each anchor (a large number of them), non-maximum suppression (NMS) is applied to ignore the redundant proposals. The idea behind the NMS is that the proposals are sorted by the confidence (objectness score), and then the proposals overlapping with a higher-scored proposal are ignored. The threshold of the overlap is typically defined as the Intersection Over Union (IOU) between two proposals.

5.1 YOLOv3

YOLOv3 uses DarkNet-53 as base network with blocks made up of successive convolution layers of 3×3 and 1×1 filter sizes and residual connections.[22] Authors show that DarkNet-53 achieves classification similar to that of ResNet-152 on ImageNet dataset but with twice the speed. YOLOv3 uses 9 anchor boxes generated using K-means clustering where $k = 3$ for each of the 3 scales. To predict the objectness and class of each box, the authors have used multi-label classification, using independent logistic classifiers for each class instead of softmax and used binary cross-entropy loss for training. We note that loss functions are squared errors instead of cross-entropy values in YOLOv2. Several convolutional layers are added to the base network (DarkNet-53) for detection thus making the YOLOv3 a 106 layered fully convolutional architecture. The last layer predicts a 3-d tensor encoding bounding box, objectness and class predictions $S \times S \times (B \times (4 + 1 + C))$ using logistic regression. YOLOv3 predicts $B = 3$ boxes at 3 different scales by extracting features from different scales like feature pyramid networks.[23] These feature maps at three different scales are extracted from layers having strides 32, 16, 8 respectively which implies with an input of 416×416, detections are done on scales 13×13, 26×26 and 52×52. Features are extracted from last two layers of these additional layers, upsampled two times and concatenated with the feature map from the base network. The former upsampled features give semantic information while the latter from backbone gives finer-grained information about the objects.

YOLOv3 achieves comparable results with the other detection methods for COCO dataset considering 0.5 IOU and at a speed that is three times higher than RetinaNet.[24] YOLOv3 with DarkNet-53 being a 106 layered architecture, is slower than the YOLOv2 but has higher accuracy and is still faster than other state-of-art methods of object detection. At the first level, image is down sampled by the network till 81 layers which has a stride of 32. So, the 416×416 image becomes $13 \times 13 \times 255$. One detection is made at this level. Feature map from layer 79 is connected to convolutional layer 84 and then upsampled by 2 times resulting into 26×26 feature map. This feature map is then depth concatenated with the feature map from layer 61. This concatenated feature map is again passed through some stacked 1×1 convolutional layers and at layer 94 the feature map of $26 \times 26 \times 255$ is used for detection. The features from layer 91 are passed through other convolutional layers till layer 97 and depth concatenated with feature map from layer 36 of the classifier network. This concatenated feature map is passed through the convolutional layers till the 106^{th} layer and then the detection is performed on this $52 \times 52 \times 255$ feature map. This way, the detection is performed at 3 different scales reducing the image using strides of 32, 16 and 8 on the output of the convolutional layers.

6. DETECTION OF HEALTH CONDITION WITH TEA IMAGES

Human participatory sensing enabled through mobile phones and related web services offer a powerful mechanism to collect and analyze relevant data for use in studying and providing solutions based on inferences of the submitted data. Farmers in different regions submit events with images, audio and meta-data during the entire cultivation cycle of the crop from sowing to harvest. This data is used for creating personalized advisory systems for challenges related (but not limited) to crop diseases, pests, weeds and use of correct seeds and chemicals. We collected tea crop images infected with certain diseases and pests of which Tea Mosquito Bugs (TMB) and Red Spider Mites (RSM) are most reported incidents. This being a crowd sourced system, the quality, exposure and resolution of the images submitted are quite different and challenging. Standard data augmentation tricks like changes in brightness, saturation, cropping, rotations, shifts have been used to increase the dataset. A total of 250 images for each pest occurrence were collected. The number of labelled bounding boxes in these images is 478 for TMB and 689 for RSM. Using data augmentation techniques, we increased the dataset to 2000 images for each category.

6.1 Feature Extraction and Classification

Various deep convolutional networks with different architectures and improvements in the accuracy and speed as a result have been proposed for image classification. The classification models are evaluated in terms of the number of floating point operations, depth, number of parameters, sparsity influence the computational speed and memory requirement. Considering the trade-off between speed and accuracy, we have taken MobileNet, Inception v2 and ResNet-101 architectures in our experiments for feature extraction as these are different in terms of architectures. VGG-16,[25] Inception v2 [26] and Resnet-101[27] have been the state-of-art CNNs in classification and detection challenges. MobileNet is designed for efficient inference in various mobile vision applications as depthwise separable convolutions used in it result into the classification accuracy similar to VGG-16 on ImageNet within about 30 times lesser computational and memory requirement. DarkNet-19 and DarkNet-53 are fully convolutional networks, proposed with YOLOv2 and YOLOv3 respectively and have acheived reasonable classification accuracy and speed.[22] We have evaluated these backbone networks for their accuracy in classification of different health conditions in tea leaf images. Table 1 shows the performance of these classifiers on the tea leaf dataset.

Table 1. Comparison of Backbone Networks with tea leaf dataset

Backbone	**Accuracy(%)**	**Parameters (mn)**	**Flops (bn)**	**Memory (MB)**
DarkNet-19	74.1	15.7	7.29	202
MobileNet	71.1	3.3	1.2	17
Inception v2	73.9	10.17	5.7	95
ResNet-101	76.4	42.60	19.7	170
DarkNet-53	77.2	61.5	18.7	248

6.2 Training

The detection model is trained with the images from dataset consisting of images of the affected tea leaves and their corresponding bounding box and class annotations in XML files. The bounding box values used for training are saved in YOLO format, i.e. the X-Y coordinate of center, width and height of the bounding box. The X value of center coordinate and width of bounding box are normalized by the width of image while Y value of center and height of bounding box are normalized by height of the image.

The feature extractor backbones used for detection are the ones pre-trained with the ImageNet dataset. Training procedure used for YOLOv2 and v3 is same except for the loss functions. The loss function is L2 (sum of squared error loss) for box prediction for YOLOv3 while cross-entropy loss is used for getting objectness confidence and class predictions. All loss functions in YOLOv2 are sum of squared errors. The dataset was divided in 80-20 proportion for training and validation multiple times and the test accuracy was measured. The input dimension of the network is defined to be $416 \times 416 \times 3$ and the image is downsampled by 32. As YOLO is a FCN, we trained the network while changing the image input dimensions in multiple of 32 from $160 \times 160 \times 3$

to $608 \times 608 \times 3$ because the image sizes vary in the collected dataset as they have been captured using different mobile phones. All the experiments were performed on the GPU - GTX1050Ti. Anchors for both the versions were calculated using K-means clustering. For training the detection architecture, we use batch size of 2 and Adam optimizer with a clipping of parameter gradients to a maximum norm of 0.001. Starting learning rate selected is $10e-4$ and is reduced by a factor of 0.1 on plateau. The graph of loss function for detection task at the last layer is shown in Fig. 2 where it can be seen that the model converges. The loss function for YOLO is sum of loss functions for bounding box, objectness score and class predictions.

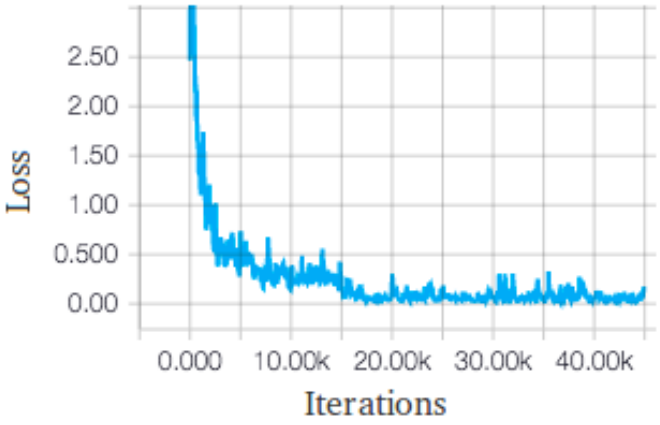


Figure 2. Loss function in YOLOv3 with DarkNet-53

6.3 Accuracy and Inference time

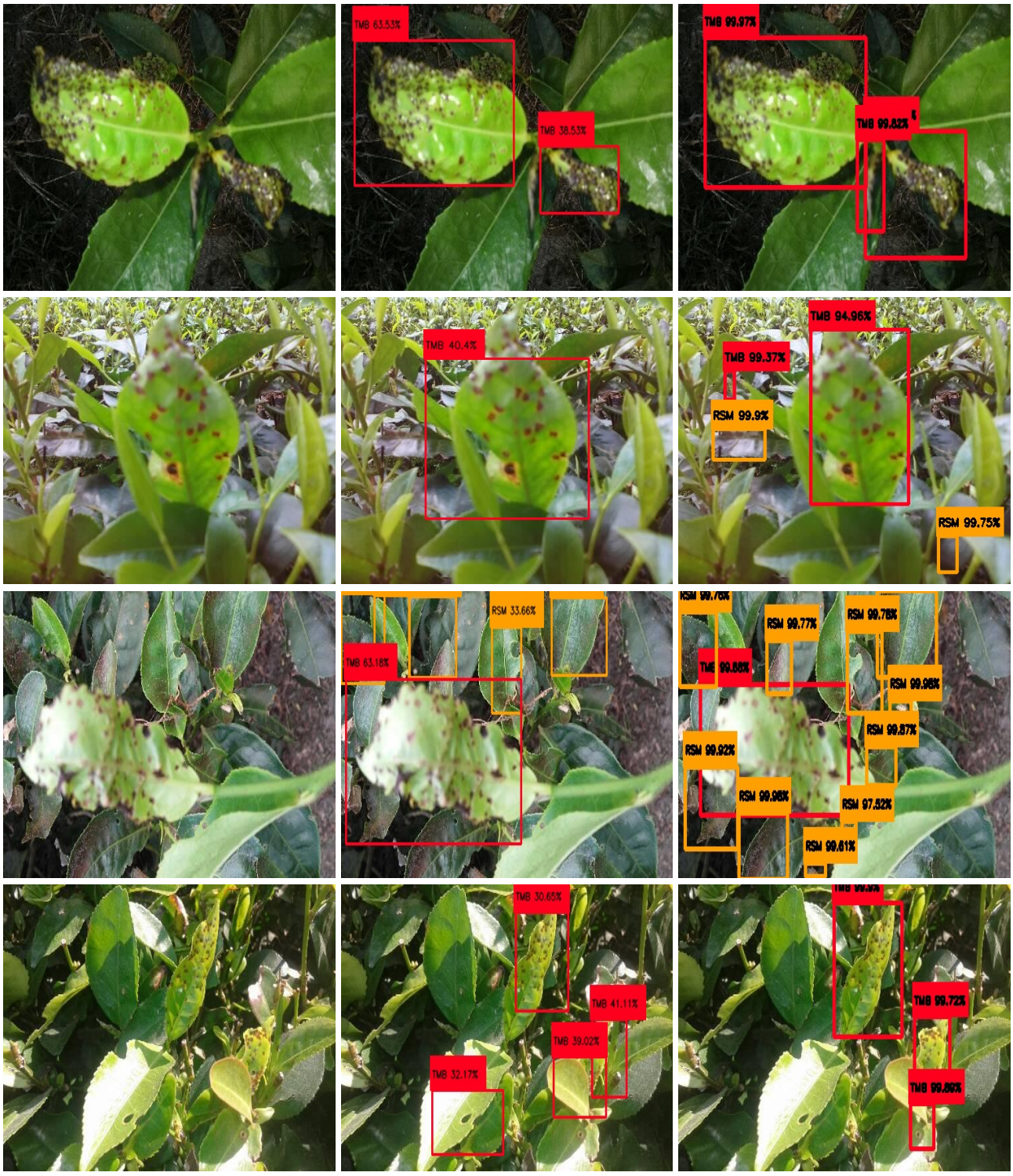
It is usually observed that there is a trade-off between performance accuracy and speed in case of classification and object detection methods. Increase in the depth of the classification network DarkNet-53 has increased the detection capability of YOLOv3 especially with smaller objects, but at the same time made it slower than YOLOv2 that uses DarkNet-19. The detection of pest affected leaves that are smaller and occluded is more accurate in YOLOv3, while it is missed in the detection results of YOLOv2 which can be seen in Fig. 3. These results are obtained using DarkNet-19 as backbone with YOLOv2 and DarkNet-53 for YOLOv3. The bounding boxes on detection results in red are for TMB and those in yellow are for RSM conditions.

Fig. 4 shows the precision-recall curves for detection of both the classes of pests at 50% IOU obtained by the YOLOv3 method. The individual average precision for TMB is 86.65% while that for RSM is 84.69%. With different set of test images from the whole dataset, the mAP varies by about 5.2% and 3% on an average for YOLOv2 and YOLOv3 respectively. This variation is mainly due to various levels of severity and clarity of the affected regions in the test images.

Table 2 shows that significant improvement is obtained in detection accuracy at 50% IOU with YOLOv3 architecture and DarkNet-53 backbone. Moreover the inference time in the same table suggest that the system is still real-time though there is minor increase in inference time of YOLOv3 due to deeper backbone compared to DarkNet-19 and MobileNet.

Table 2. Evaluation of Detection methods

YOLO versions	Backbone	mAP	mAP 50	mAP 75	Time (ms)
YOLOv2	DarkNet-19	46.6	68.4	46.4	102
	MobileNet	48.9	68.4	45.6	87
	Inception v2	56.7	81.6	58.3	230
	ResNet-101	58.4	82.7	61.1	310
YOLOv3	DarkNet-53	60.3	85.7	61.9	120



(a) Tea leaves (b) YOLOv2 (c) YOLOv3

Figure 3. Column (a): Tea leaf test images, Column (b) YOLOv2 detection result, Column (c) YOLOv3 detection result for Tea Mosquito Bugs (TMB) and Red Spider Mites (RSM)

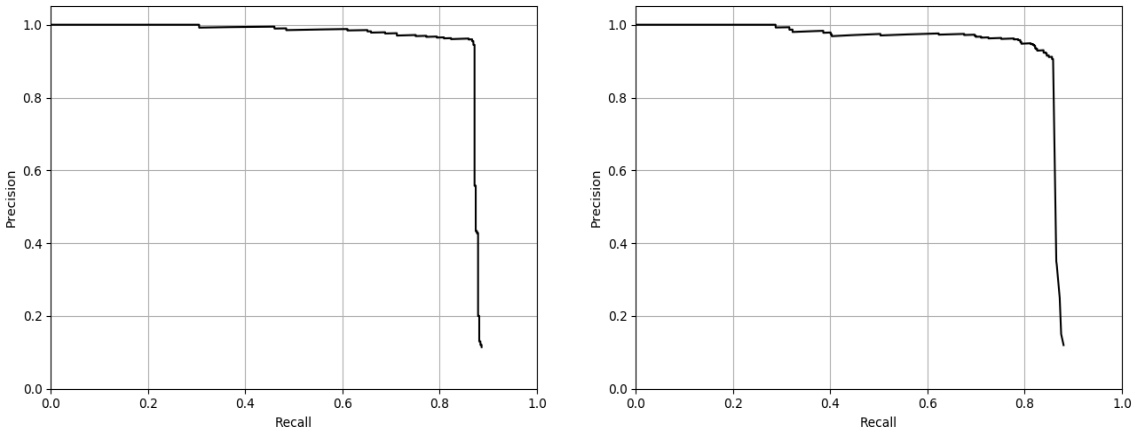


Figure 4. Precision-Recall (PR) curves at IOU=50%. (left) Tea Mosquito Bugs with Area Under Curve: 86.65%, (right) Red Spider Mites with Area Under Curve 84.69%

7. CONCLUSIONS

In this paper, we have proposed application of single shot detection in tea leaves that helps in reducing the disease or pest detection time through automation. From the results, it can also be realized that only classification cannot reliably help when there are occurrences of multiple pest or disease conditions in a single image. Apart from automated identification, such a system is a good step towards automated and selective spraying, crop yield prediction based on occurrences and lesser dependence on time consuming lab tests or human diagnosis. YOLOv3 is more accurate in classification as well as localization when compared to YOLOv2 which is the fastest. It can be seen that there is a trade off between speed, accuracy and model size between different detection methods and the backend models. Our model can detect the affected areas in different lighting and resolution conditions due to variety of such conditions considered while training the models. The time, model size and accuracy analysis assures that the single shot detection method satisfactorily perform well in real-time as well as on the edge detection. Though the performance needs to be validated with more number of pests and diseases, the model can be refined with the increase in reported events and associated images. Our goal was to find a suitable and real time method to reliably detect tea health conditions. This study can be extended to other crops and health conditions like deficiency, diseases and pests with a focus to improve results and also measure the severity.

8. ACKNOWLEDGEMENT

We thank Tata Global Beverages Limited and Amalgamated Plantations Private Limited for their support in the data collection process for this work.

REFERENCES

[1] Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., Villa, P., Stroppiana, D., Boschetti, M., Goulart, L. R., et al., "Advanced methods of plant disease detection. a review," *Agronomy for Sustainable Development* **35**(1), 1–25 (2015).

[2] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," in [*Advances in neural information processing systems*], 91–99 (2015).

[3] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 779–788 (2016).

[4] Dai, J., Li, Y., He, K., and Sun, J., "R-FCN: Object detection via region-based fully convolutional networks," in [*Advances in neural information processing systems*], 379–387 (2016).

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "SSD: Single shot multibox detector," in [*European conference on computer vision*], 21–37, Springer (2016).

[6] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in [*IEEE CVPR*], (2017).

[7] Du, Q., French, J. V., Skaria, M., Yang, C., and Everitt, J. H., "Citrus pest stress monitoring using airborne hyperspectral imagery," in [*Proceedings of IEEE International Geoscience and Remote Sensing Symposium*], **6**, 3981–3984 (2004).

[8] Franke, J. and Menz, G., "Multi-temporal wheat disease detection by multi-spectral remote sensing," *Precision Agriculture* **8**(3), 161–172 (2007).

[9] Wang, X., Zhang, X., and Zhou, G., "Automatic detection of rice disease using near infrared spectra technologies," *Journal of the Indian Society of Remote Sensing* **45**(5), 785–794 (2017).

[10] Belasque Jr, J., Gasparoto, M., and Marcassa, L. G., "Detection of mechanical and disease stresses in citrus plants by fluorescence spectroscopy," *Applied Optics* **47**(11), 1922–1926 (2008).

[11] Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., and Iyatomi, H., "Basic investigation on a robust and practical plant diagnostic system," in [*15th IEEE International Conference on Machine Learning and Applications*], 989–992 (2016).

[12] Bhatt, P., Sarangi, S., and Pappula, S., "Comparison of CNN models for application in crop health assessment with participatory sensing," (2017).

[13] Ramcharan, A., Baranowski, K., McCloskey, P., Ahamed, B., Legg, J., and Hughes, D., "Transfer learning for image-based cassava disease detection," *Frontiers in plant science* **8**, 1852 (2017).

[14] Mohanty, S. P., Hughes, D. P., and Salathé, M., "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science* **7**, 1419 (2016).

[15] Baweja, H. S., Parhar, T., Mirbod, O., and Nuske, S., "Stalknet: A deep learning pipeline for high-throughput measurement of plant stalk count and stalk width," in [*Field and Service Robotics*], 271–284, Springer (2018).

[16] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C., "Deepfruits: A fruit detection system using deep neural networks," *Sensors* **16**(8), 1222 (2016).

[17] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y., "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229* (2013).

[18] Huang, J., Rathod, V., Sun, C., Zhu, M., Balan, A. K., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K., "Speed/accuracy trade-offs for modern convolutional object detectors," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , 3296–3297 (2017).

[19] Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., "Microsoft COCO: Common objects in context," in [*ECCV*], (2014).

[20] Hoiem, D., Chodpathumwan, Y., and Dai, Q., "Diagnosing error in object detectors," in [*European conference on computer vision*], 340–353, Springer (2012).

[21] Redmon, J. and Farhadi, A., "YOLO9000: Better, faster, stronger," in [*2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 6517–6525 (2017).

[22] Redmon, J. and Farhadi, A., "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767* (2018).

[23] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S., "Feature pyramid networks for object detection," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 2117–2125 (2017).

[24] Lin, T., Goyal, P., Girshick, R. B., He, K., and Dollár, P., "Focal loss for dense object detection," in [*IEEE International Conference on Computer Vision*], 2999–3007 (2017).

[25] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556* (2014).

[26] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., "Rethinking the inception architecture for computer vision," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 2818–2826 (2016).

[27] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 770–778 (2016).