Disease Detection in Tea Leaves: A Hybrid Model using YOLOv7 and DCNN

Md Zahidul Kabir
Assistant Programmer
Ministry of Shipping, Bangladesh
Secretariat, Dhaka.

Md Sourav Hossen
Co-Ordinator
Bangladesh Computer Council,
Dhaka, Bangladesh

Sumiya Kaisar Keya
Department of English,
Gopalganj Science & Technology
University, Bangladesh

ABSTRACT
Tea drinking has been a part of culture for thousands of years plays a crucial role in daily life in today's world. It is difficult to find a person who does not drink tea 2-3 times a day, especially in South Asia. So, this important drink in our daily lives should be fresh and risk-free. It is very important and significant to detect the tea leaf disease timely in order to more production of fresh and risk-free tea. To detect tea leaf disease, numerous conventional methods are established including CNN, Deep CNN, DNN, AX-Retina Net, improved DCNN, YOLOv5, YOLOv7, and multi-object image segmentation etc. This research proposed a solution of hybrid model using YOLOv7 and DCNN for tea leaf disease detection with an improved accuracy rate. Here, the strong object detection features of YOLOv7 were utilized for identifying the detected regions and the deep learning methodology of DCNN was employed for identifying disease type accurately. This proposed model was experimented with a customized dataset of approximate 5,000 image that is collected from Bangladesh tea research institute (BTRI) and forest department, Bangladesh (FDB) achieved a mAP of 97.6%, a Precision of 95.2%, and an Accuracy of 97.8% for detecting the tea leaf disease.

Keywords
Tea Leaf Disease, Leaf Disease Detection, Hybrid Model, Disease Classification, Diseased Region Detection, Bangladesh Tea Research Institute (BTRI), Forest Department, Bangladesh (FDB), You Only Look Once (YOLO), Deep Convolutional Neural Network (DCNN).

1. INTRODUCTION
Tea is currently a very important crop with ranks among the most favored beverages globally around the world. Also, it is the good source of vitamin C, vitamin B1, vitamin B6, carotene, and folic acid [1]. The Food and Agriculture Organization of the United Nations (FAO) reports that it is the second most consumed liquid worldwide, following water. Research over the years has highlighted its potential benefits, ranging from alleviating fatigue to promoting longevity [2]. Rich in antioxidants, tea is vital for enhancing the immune system and overall well-being. Furthermore, tea stimulates mental clarity, revitalizes the body, and enhances performance.

Tea plants are affected by various diseases and pests in adult, immature or seedling stages. Generally, tea plant diseases are caused by fungi and algae [3]. Sometimes the diseases are small in scale and sometimes they are widespread and cause great damage to tea. The main diseases and pests that attack the tea leaf are: wheel spot disease, die back disease, white star disease, leaf rot disease, brown and gray blight disease, blister disease, red rust disease, branch canker disease, horse hair blight disease, thread blight disease, charcoal stump rot, black rot, violet root rot, and purple root rot etc [3, 4]. It is very important to identify tea leaf diseases as soon as possible. In Bangladesh, in normal conditions, about 10-15% of tea yield is reduced due to diseases alone [1]. There is a widely used, reliable method for detecting plant diseases (see Figure 1) that also detects tea leaf diseases with a high degree of accuracy. Nevertheless, these existing models has limitations, like that some models are being effective for small areas only, and some have inadequate data availability.

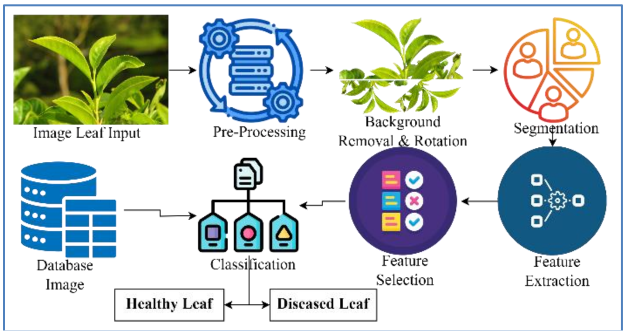

Figure 1: Plant Disease Detection Process.

This research proposed a hybrid model (see **Figure 3**) that combines YOLOv7 (you only look once version 7) and DCNN (deep convolutional neural network). The robust object detection features of YOLOv7 were utilized for identifying the detected regions, while the deep learning methodology of DCNN was employed for identifying disease type accurately. This research utilized a customized dataset collected from BTRI and FDB, comprising approximately 5,000 images of various types of diseased tea leaves, healthy tea leaves, and other plant leaf images (see **Figure 2**).

After that, the images were processed for making useable as dataset by doing the mechanism of resize the image (1), image duplicity removal, normalize the data, noise removal (4), extract the reason of interest (ROI), encode the label, convert image to tensor, and handling class balance. A powerful computer was set featuring an Intel Core i7 processor, complemented by 32 GB of RAM and 500 GB SSD, is located in room 816(A) of the Ministry of Shipping, Bangladesh. For the experiment, Python 3.8, CUDA for GPU acceleration, PyTorch for developing deep learning models, TensorFlow for constructing the DCNN model, OpenCV for image preprocessing and visualization, and YOLOv7 were installed on this machine. In this experiment, the proposed model achieved an improved accuracy rate of 97.8% while the standalone DCNN and YOLOv7 recorded accuracy rates of 94.7% and 96.4% respectively (see Table 2, see **Figure 7**).

This paper is organized as follows. Section 1 is described the introduction of this research. Section 2 reviews the existing literature on tea leaf disease detection, highlighting key findings and identifying gaps in the current knowledge. Section 3 described the research methodology, including details about the data collection and processing in subsection 3.1, data augmentation in subsection 3.2, finally model implementation in subsection 3.3. Section 4 presented the results of the study. Finally, Section 6 concludes the paper by summarizing the key findings, and recommending directions for future research.

2. RELATED WORK

Currently, there are various models for diagnosing tea leaf diseases mainly are deep CNN [2, 8], YOLOv5, YOLOv7[5], BHC-YOLOV8[6], YOLOv8-RCAA [7], CNN [8], Deep CNN [2, 8], DNN [14], AX-Retina Net, improved DCNN, multi-convolution neural network [13], AI model, neural network ensemble [17], hybrid model [15], and multi-objective image segmentation [16]. The analysis of several model is presented in the T**able 1.**

CNN (convolutional neural network) and deep CNN (deep convolutional neural network) are very effective methods for diagnosing tea leaf diseases [9]. It extracts features such as color, texture, and pattern from leaf images through convolution layers. CNN basically first identifies the type of disease of a plant [9-11].

On the other hand, deep CNN has convolution and pooling layers in which activation functions (such as ReLU) [10, 12] help in creating complex patterns of image features and the pooling layer preserves important information in the feature map [2, 8, 13]. Finally, the fully connected layer detects diseases.

In the case of DNN (deep neural network) [14, 15], the pixel data of the image is taken as input. It creates complex patterns in many of its hidden layers and identifies the characteristics of leaf diseases in the image processing process [12, 14, 15]. In the case of DNN, the larger the database, the more accurate the results.

YOLO (you only look once) is a very effective method for diagnosing leaf diseases. It divides the input image into different grids and creates complex patterns of image features. Using the complex patterns of the grid [5-7], it identifies disease-related areas in each grid [6, 7]. Therefore, it is possible to accurately determine the diseased region and type of tea leaf disease in a short time [5-7].

BHC-YOLOv8 [6] is a model using boundaries, hierarchies, and context (BHC) to improve detection accuracy is the adaption of the YOLOv8 framework. Even in the presence of complex backgrounds, BHC-YOLOv8 enhances the model's capacity to detect subtle disease patterns and abnormalities on tea leaves by combining multi-scale feature extraction and hierarchical context analysis.

A DNN supplemented with hybrid pooling [15] is used in a new AI-based method combines many pooling approaches such as max pooling and average pooling. The cutting-edge technique increases the model's robustness to changes in leaf sizes, shapes, and disease patterns in addition to improving its accuracy in identifying different disease in tea leaves.

Multi-objective image segmentation [16] is a complex method for diagnosing leaf diseases. First, it identifies disease-related areas and extracts various features. Second, it divides the image into different parts and analyzes various complex features of each part [8, 16].

Neural network ensemble [17], that combines multiple deep learning models to enhance accuracy in disease classification. The ensemble approach captures a variety of tea leaf disease characteristics by utilizing the capabilities of many neural networks, enhancing the model's capacity to differentiate between healthy leaves and those afflicted with blight, scab, or other conditions. This model's drawback is that it takes longer than other single networks.

Table-1: Related Works Analysis

Reference	**Technique**	**Accuracy**	**Strength**	**Limitations**
Hu Gensheng [2]	DCNN	92.5	Capable to extract complex features. Comparatively high accuracy.	The larger the data, the greater the accuracy. Less qualitied image, less accuracy.
S. Gayathri [3]	Deep CNN	90.23	Extract complex features that provides high accuracy.	Needs a large, high-quality datasets. Image of less lighting and less resolution decrease accuracy.
Janibul Alam Soeb [5]	YOLOv7	97.3	Detect and classify the diseases in one step. Also, can handle large datasets efficiently.	Less focus on types of disease. Also lack of transparency in decision-making.
BaiShao Zhan [6]	BHC-YOLOV8	94.5	Capable to classify the types of disease accurately.	For new disease it shows less adaptability. As it is customized architecture with high cost to implement.
Jing Chen, Qi Liu [10]	CNNs	90.16	Capability of making powerful image features.	The larger the data, the greater the accuracy. Comparatively challenging to compile.
Saikat Dat ta [14]	Deep Neural Network	96.56	Provides better accuracy and scalable to different crops.	Lack of comprehensive field testing on tea-growing regions. Does not mention the challenges of the deployment.
Oidong heng [15]	DNN based hybrid pooling	92.47	Designed for accurate object detection with better feature extraction and handling of small objects.	It does not perform well on rare disease types and complex backgrounded image.

Somnath Mukhopadhyay [16]	Multi-Objective image Segmentation	83	Input image is segmented different regions that provides high accuracy.	Advanced Algorithm and powerful hardware required for implementation.
Bikash Chandra Karmokar [17]	Neural Network Ensemble (NNE)	91	Combination of multiple neural networks. Strong capability to detect disease accurately.	Take more time compared to other single network. More significant resources for the implementation.

3. METHODOLOGY

The methodology of this research of tea leaf disease detection mainly consists of three steps like data processing, data augmentation and finally model implementation.

3.1 Data Processing

A customized dataset containing approximately 5000 leaf images including diseased, healthy and others plant leaf images (see **Figure 2**) ware collected from Bangladesh tea research institute (BTRI) and forest department, Bangladesh (FDB) was used for this research experiment. To make the dataset useable, a number of data processing procedures were carried out, including resize the image (1, 2), normalization (3, 4), image duplicity removal, noise removal (17), cropping region of Interest (ROI), impose filtering as needed, encode the label, dataset split & balancing, preparation of batch, convert image to tensor, and handling class balance [9, 17, 18]. Also, the dataset was made balanced by using additional technique like removing duplicity, oversampling, undersampling or synthetic data generation etc.

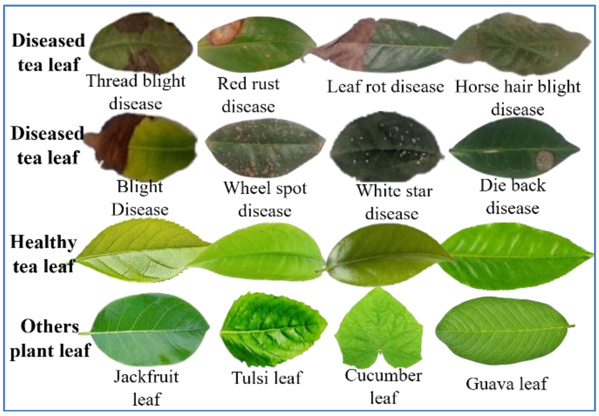

Figure 2: Tea leaf image collected from BTRI and FDB

This study emphasized the input image size of 224x224 pixel because image size for DCNNs is 224x224, 256x256, or 384x384 [8-9, 19]. So, equations (1, 2) were used for resizing the image from different size to 224x224.

$$Resized\ Image = f(I_{old}, x, y) \quad (1)$$

$$f(x,y) = \frac{(x_2-x)(y_2-y)}{(x_2-x_1)(y_2-y_1)} f(Q_{11}) + \frac{(x-(x_1)(y_2-y)}{(x_2-x_1)(y_2-y_1)} f(Q_{21}) \quad (2)$$

Where, f: Resize function, I_{old}: Original image, x, y: New image dimension, Q_{11}, Q_{21}: Neighbo-ring pixel value.

In order to make the image useable as a dataset, normalization [4, 18] is crucial. The collected images and input image ware normalized using the following equations (3, 4).

$$P_{norm} = \frac{p - \min(p)}{\max(p) - \min(p)} \quad (3)$$

$$P_{norm} = \frac{p - \mu}{\sigma} \quad (4)$$

Where, μ: The mean of dataset, σ: Standard deviation

In order to improve the experiment's outcome, the following equation (5, 6) were applied for filter and noise-removal procedure [4] in both collected and input image. For the purpose of eliminating noise from the input and dataset images, it was employed the Gaussian algorithm (5, 6), [20].

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{X^2+Y^2}{2\sigma^2}} \quad (5)$$

$$I_{filtered} = (I * G)(x,y) \quad (6)$$

Where, σ: Standard deviation, G(x, y): Gaussian kernel that is applied to the image I(x, y).

Next step was to convert disease label into numeric format (7) by cropping the region of interest (ROI) from the image (8). For detecting the disease accurately, it was very important to focus the specific part of the image. The dataset was split into training, validation, and test sets. Our split ratio is respectively 80%, 15% and 15%. Let our dataset is D, so the split dataset of D_{train}, D_{val}, and D_{test} were as follows (9):

$$Label = [0,0, \ldots .,1 \ldots .,0] \quad (7)$$

$$R = Image[(x_1, y_1), (x_2, y_2)] \quad (8)$$

$$D_{train} = \frac{85}{100} * D,\ D_{val} = \frac{15}{100} * D,\ D_{test} = \frac{15}{100} * D \quad (9)$$

A distinctive customized database with roughly 5,000 images including diseased, healthy and others plant leaf image was ready for the experiment after the entire process was finished. This customized dataset was utilized to experiment the proposed hybrid model using YOLOv7 and DCNN as well as the standalone YOLOv7 and DCNN methods for detecting tea leaf disease.

3.2 Data Augmentation

Data augmentation is a technique that plays the crucial role to extract the complex features from the dataset images [21]. There are a number of methods or processes for augmenting data, such as contrast modification, image rotation, image flipping, and image sealing. For a better experiment, rotation of θ degree angle in photos, flipped in both horizontally and vertically as needed, enlarged by a factor of S, and displaced them by (Δx, Δy) [9] ware utilized.

When the work or research is pattern recognition then it is very helpful to use the technique of convert image color to grayscale. It this research the following technique and equations (10, 11) were used for converting image color to grayscale [18]. Also, modified the pixel values for the sake of adjustment (10) of the brightness of image input and image dataset by using the following equations (10, 11)

$$I_{adjusted} = \alpha I + \beta \quad (10)$$

$$Grayscale\ Image = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (11)$$

Where, α: Contrast, β: Brightness control, R, G, B: Pixel value of red, green and blue color.

3.3 Model Implementation

The procedure of feature extraction, feature aggregation, diseased region identification, region of interest (ROI) extraction, disease type detection and finally disease detection were all followed by the proposed model (see **Figure 3**) for implementation.

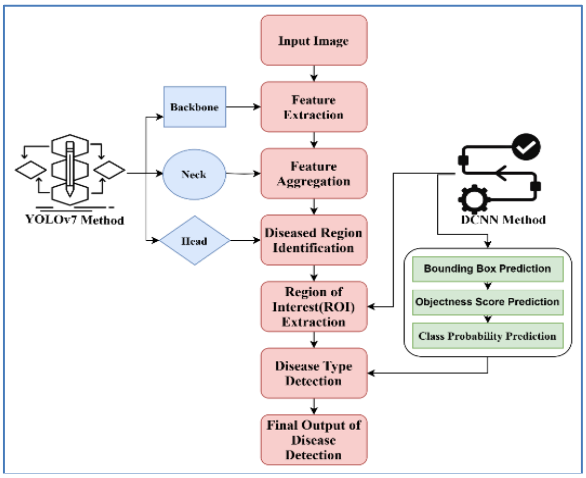

Figure 3: Disease Detection Process Using Proposed Model

Firstly, in order to execute the model, three different image resolutions were used as inputs: high, medium, and low.

Secondly, this step was for extracting feature (see **Figure 4**) of the input image using the YOLOv7 model [21]. Features extract in different perspectives like low-level, mid-level and high-level feature plays crucial role for detecting disease in different image sizes (see **Figure 4**). The feature of edges, texture and color with high resolution but fewer channel represented in feature map extracted in initial stage or low-level stage. After initial stage it's time to extract diseased region part and shapes of input image as mid-level feature with moderate resolution and better number of channels. Final stage was high-level feature like total object and different pattern extraction in feature map with low resolution and huge number of channels [5].

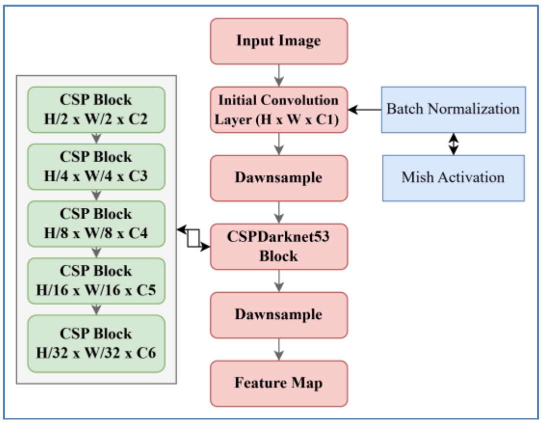

Figure 4: Feature Extraction Process using YOLOv7

For feature extraction (see **Figure 4**), there were several layers used in YOLOv7 [6]. First layer is convolution layer with batch normalization and mesh activation (14). In convolution layer, filters (karnels) are applied in input image for getting different features like edges, textures, and patterns. Batch normalization (12, 13) that is applied every convolutional layer works in the output that adjusting and scaling of the convolution layer [5, 6]. Then applied Mish Activation (14) because it was providing which helped in better feature representation like smoother activation, better gradient flow, faster convergence, and better performance. Equations follows here:

$$x' = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{12}$$

$$y = \gamma x' + \beta \tag{13}$$

$$Mish(x) = x.\tanh\,(\ln\,(1 + e^x)) \tag{14}$$

Where, μ: Mean of the input batch, σ^2: Variance of the input batch, ϵ: Small constant for numerical stability, γ: Learnable scaling, β: Shifting parameters, y: Normalized output.

After convolutional layer, the next steps are creating feature map using multiple CSP block. In this purpose, CSPDarknet53 architecture [6] that reduce computational redundancy and also improve gradient flow was used. In a CSP block, input feature map [19] is denoted in F_{in}. then split the input feature map into two parts of F1 and F2. For feature extraction F1 has passed by a series of blocks and F2 bypass these blocks. where *f*() represents the convolutional operations within the residual block. The outputs of the residual blocks and the bypass path are concatenated through the following equation (15).

$$\begin{aligned} F_1, F_2 &= f(F_{in}) \\ F_{res} &= f(F_1) + F_1 \\ F_{out} &= f(F_{res}, F_2) \\ F_{down} &= f_{stride=2}(F_{out}) \end{aligned} \tag{15}$$

By increasing the number of channels and decreasing the spatial dimensions of the feature map, the network is able to collect higher-level features.

Thirdly, this step is for aggregating feature using in YOLOv7 model [6, 7, 22]. The feature map from CSPDarknet53 [20, 21] is aggregate and refine of different layer of features that provide better quality network model from detecting disease in the leaf. The purpose of feature aggregation was to improve the quality of feature map and provide better performance. Here used PANet (Path Aggregation Network) [6] for aggregating feature in different stage of network. In this network it followed top-down pathway from higher level to lower-level feature that make strong ability for disease detection and the bottom-up pathway is reverse of top down. The final output was the combination of top-down and bottom-up pathway that provides the facility of preservation both high level information and low-level information with different size and complexation [6, 7].

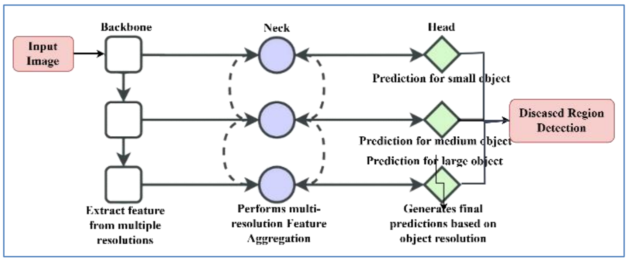

Figure 5: YOLOv7 Architecture for diseased area detection

Fourthly, this step is for the detection of diseased region using existing architecture (**see Figure 5)** of YOLOv7 [5-7]. In this stage the input image was separated into a grid of cells that were used to map the feature maps [6]. Each cell represents a group of anchor boxes and anchor box represents input image object dimensions, which were predetermined shapes. By adjusting to the typical aspect ratios of the objects in the dataset, they assist the model in producing bounding boxes (see **Figure 6**) that were more accurate.

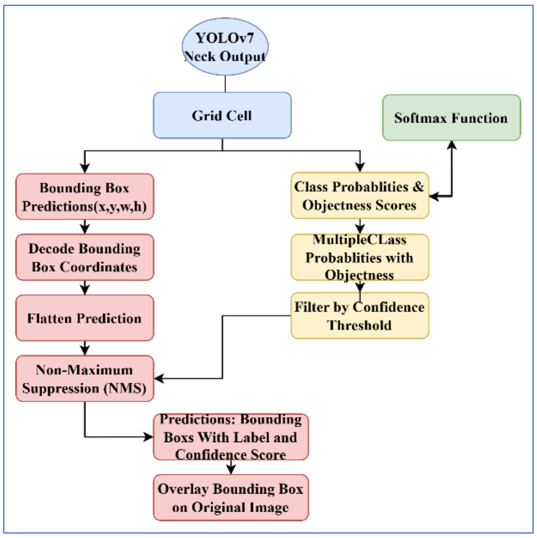

Figure-6: Diseased Region Prediction in YOLOv7

YOLOv7 anticipated a number of crucial components for each anchor box that were necessary for the creation of bounding boxes [7]. These consist of the box's width, height, and center coordinates (x, y) with respect to the grid cell [6]. The predetermined dimensions of the anchor box were used to refine these predictions. The model could precisely localize objects thanks to the bounding box coordinates that were produced, which indicated the diseased area location within the image. Multiple bounding boxes were predicted by each grid cell, and these predictions could be adjusted further during training to match the dataset's ground truth bounding boxes (see **Figure 6**). A bounding box was predicted by each feature map cell in relation to its grid location [5, 7]. The expected bounding box for a cell located at the feature map was followed by following equations (16-19):

$$b_x = i + \sigma(t_x) \quad (16)$$
$$b_y = j + \sigma(t_y) \quad (17)$$
$$b_w = e^{t_w}.w_{prior} \quad (18)$$
$$b_h = e^{t_h}.h_{prior} \quad (19)$$

Where, b_x, b_y: Center coordinates of the bounding box, b_w, b_h: Width, height of the bounding box, t_x, t_y, t_w, t_h: Predicted offsets, σ: Sigmoid function, w_{prior}, h_{prior}: Prior width and height.

Each grid cell predicted the item class and the objectness score in addition to the bounding box coordinates (16-19). In the meantime, the probability that a detected region was present in the bounding box was indicated by the objectness score [6, 7]. This score lowers false positives and increases detection accuracy by assisting the model in eliminating predictions when there was little chance that the detected region be present. The boxes were ranked according to their objectness ratings, and the box with the highest score was retained while ones with a large overlap were discarded.

$$p_c = \frac{e^{t_c}}{\sum_{k=1}^{k} e^{t_k}} \quad (20)$$
$$o = \sigma(t_o) \quad (21)$$

Where, k: Number of classes, t_o: Predicted objectness value.

The ultimate result was produced by the detection head of YOLOv7comprises a collection of bounding boxes [20], each linked to a specific class label, a confidence score, and the coordinates of the detected region [5-7] (see **Figure 5, 6**). This output was utilized by YOLOv7 to recognize and categorize detected region present in the image, indicating their locations and assigning appropriate labels [5, 6]. Through the integration of feature aggregation in the neck of YOLOv7, precise bounding box predictions, and objectness score (20, 21) YOLOv7 attains rapid and accurate diseased region detection. The entire process of diseased region prediction using proposed hybrid model were shown in **Figure 6**.

Fifthly, this step is to identify the type disease using DCNN model [2]. The architecture of a Deep Convolutional Neural Network (DCNN) generally comprised multiple layers, each tailored to extract distinct features from the input image. The initial layers, referred to as convolutional layers [2], utilized filters on the input feature map to capture fundamental characteristics such as edges, textures, and colors (22), [10]. As the network progresses deeper into its structure, the filters evolved in complexity, identifying more advanced features such as shapes, patterns, and areas indicative of disease symptoms.

Subsequent to the convolutional layers [2, 20] pooling layers were employed to diminish the spatial dimensions of the feature map, allowing the network to concentrate on the most pertinent features while reducing computational demands. Ultimately, the fully connected layers [2, 20] were tasked with producing the final classification [10], which indicated the type of disease present or confirms the health status of the leaf (22-25), [4]. The DCNN involved fine-tuning the network's weights using the custom database of leaf images and implementing optimization methods to reduce classification errors (25).

The classification of leaf diseases using DCNN adheres to a supervised learning framework, wherein the model was trained on the custom dataset of leaf images. The dataset generally includes images of healthy leaves alongside those affected by various diseases, such as blight, or fungal infections. The following equation were used in different DCNN layers like equation (22) was in convolutional layer, (23) was in pooling layer, (24) was in fully connected layer and (25) is in softmax function.

$$Z_{i,j} = \sum_{m=1}^{k}\sum_{n=1}^{k} x_{i+m-1,j+n-1}.K_{m,n} + b \quad (22)$$
$$y_{i,j} = \max_{m,n}\{x[i.s+m, j.s+n]\} \quad (23)$$
$$y = W.x + b \quad (24)$$
$$P_{(p=i|x)} = \frac{\exp(z_i)}{\sum_{j=1}^{C}\exp(z_i)} \quad (25)$$

Where, X: feature map, K: kernel of k * k size, B: Bias term, $Z_{i,j}$: Output feature map, *s:* Pooling window slides over the input, *W:* weight matrix, *x:* Input vector, z_i: Score for class *i*, *c: T*otal number of classes.

Sixthly, this study used the loss function (26-29) in both YOLOv7 and DCNN model for performance optimization, handling multiple objects, achieving the desired performance. The loss function integrated losses related to localization, objectness, and classification [23]. The localization loss quantified the disparity between the predicted bounding boxes and the actual ground truth bounding boxes [23]. The objectness loss imposed a penalty on inaccurate predictions regarding the presence of objects. The classification loss quantified the disparity between the predicted class probabilities and the actual class probabilities [23, 24].

$$L_{loc} = \sum_i[(x_i - x'_i)^2 + (y_i - y'_i)^2 + \left(\sqrt{w_i} - \sqrt{w'_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{h'_i}\right)^2] \quad (26)$$
$$L_{obj} = \sum_i o_i \log(o'_i) + (1 - o_i)\log(1 - o'_i)] \quad (27)$$
$$L_{cls} = \sum_i \sum_{c=1}^{k} y_{ic} \log(p'_{ic}) \quad (28)$$

Where, y_{ic}=1 if the *i*-th object belongs to class *c*, and otherwise

y_{ic}=0.

The overall loss was calculated as a weighted combination of the aforementioned components (29) [23].

$$L = \lambda_{loc}L_{loc} + \lambda_{obj}L_{obj} + \lambda_{cls}L_{cls} \quad (29)$$

Where, λ_{loc}, λ_{obj}, λ_{cls} are weighting factors.

YOLOv7and DCNN employed the Adam optimizer (30), [25] during its training process.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2){g_t}^2$$
$$m'_t = \frac{m_t}{1-\beta_1^t}$$
$$v'_t = \frac{v_t}{1-\beta_2^t}$$
$$\theta_t = \theta_{t-1} - \alpha\frac{m'_t}{\sqrt{v'_t}+\epsilon} \quad (30)$$

Where, m_t, v_t: First and second moment estimates, g_t: Gradient at step t, α: Learning rate, ϵ: Small constant for numerical stability.

4. EXPERIMENT RESULT

For environment setup, A powerful computer was established featuring an Intel Core i7 processor with 32 GB of RAM and 500 GB SSD, was located in room 816(A) of the Ministry of Shipping, Bangladesh. For the experiments, Python 3.8, CUDA for GPU acceleration, PyTorch for developing deep learning models, TensorFlow for constructing the DCNN model, OpenCV for image preprocessing and visualization tasks, and YOLOv7 were installed on this device.

Within the test comes about for tea leaf disease discovery employing a hybrid model using YOLOv7 and DCNN approach, the demonstration illustrated improvement in accuracy and efficiency compared to other conventional models. The YOLOv7 design was utilized for its diseased region identification capabilities, empowering the exact localization of diseased region on tea leaf. The DCNN, on the other hand, was utilized for its classification capacities, guaranteeing exact detection of various types of disease. The experiment results shown that the proposed hybrid model showed accomplished the accuracy is 97.8% on the test set, beating standalone YOLOv7 and DCNN based approaches (see Table 2) (see Figure 7). The demonstration moreover shown a low false-positive rate, reducing the chances of misclassification. The integration of YOLOv7 and DCNN permitted for an adjustment between speed and accuracy. The following measures are commonly used to assess the tea leaf disease detection model's performance.

Recall: Recall is the percentage of positive instance that are correctly identified (32), [8]

Precision: Precision is the percentage of positive predictions that are originally correct (33), [8]

F1 Score: The F1-Score offers a fair assessment of the model's performance since it is the harmonic mean of precision and recall. (34), [8]

Mean Average Precision (mAP): For object detection tasks like YOLOv7, mAP is a popular metric. After determining the average precision (AP) for every class, it takes the mean of all the classes. The following equation of Accuracy, Precision, Recall, mAP and F1 score are used for experiment result calculations (31), [4, 9].

$$Accuracy(A) = \left(\frac{tp+tn}{ts} * 100\right)\% \quad (31)$$

Where, $ts = tp + tn + fp + fn$

$$Recall(R) = \left(\frac{tp}{tp+fn} * 100\right)\% \quad (32)$$

$$Precesion(P) = \left(\frac{tp}{tp+fp} * 100\right)\% \quad (33)$$

$$F1\ Score = 2\frac{P*R}{P+R} \quad (34)$$

Where, tp: True Positive, ts: Total Sample, tn: True Negative, fp: False Positive, fn: False Negative.

Table 2: Experiment Result

Model Name	Accuracy	Precision	Recall	mAP	F1 Score
YOLOv7 (Disease Region Identify)	N/A	96.5	94.2	98.2	95.34
DCNN (Disease Classification)	97.3	96.8	95.5	N/A	96.1
Hybrid model using YOLOv7 and DCNN	**97.8**	**95.2**	**94.5**	**97.6**	**94.8**

With high precision, recall, and accuracy, the proposed model for tea leaf disease detection demonstrated outstanding performance in both diseased region detection and disease classification (see Table 2). It can identify diseased region and disease classify simultaneously so the inference time of proposed model is less than the standalone in YOLOv7 and DCNN.

This research assessed several metrics, including accuracy, precision, recall, mean Average Precision (mAP), and F1 Score (31-34) as part of the experimental outcomes. Specifically, the performance of YOLOv7 in detecting diseased regions yielded precision, recall, mAP, and F1 Score values of 96.5, 94.2, 98.2, and 95.34, respectively (see Table 2). Furthermore, the Deep Convolutional Neural Network (DCNN) demonstrated accuracy, precision, recall, and F1 Score values of 97.3, 96.8, 95.5, and 96.1, respectively, in accurately identifying disease types (see Table 2). Finally, the experiments revealed that the proposed hybrid model achieved accuracy, precision, recall, F1 Score, and mAP values of 97.8, 95.2, 94.5, 97.6, and 94.8, respectively (see Table 2).

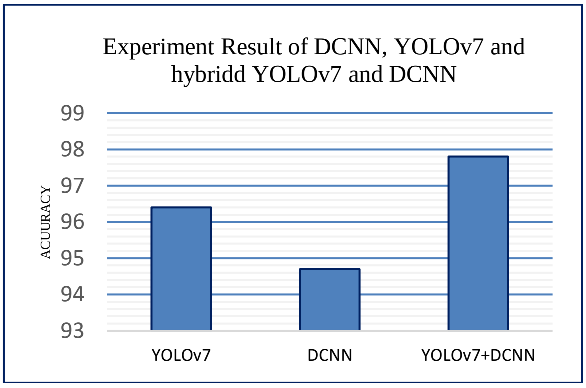

Figure 7: Graphical presentation of the experiment result

This research experimented the performance of two well-known models, YOLOv7 and DCNN, for detecting diseases in tea leaves using customized datasets. The accuracy rates obtained were 96.4% and 94.8%, respectively, while the proposed hybrid model achieved an accuracy rate of 97.8%. (see **Figure 7**)

5. CONCLUSION

A major step forward in tackling the difficulties involved in precise and effective plant disease identification is the incorporation of hybrid model of YOLOv7 and DCNN in leaf disease detection research. With the help of YOLOv7's diseased region identification and DCNN's strong disease type detection capabilities, this hybrid model performs better at identifying diseased areas, and a variety of leaf symptom patterns. Despite YOLOv7's superior speed and localization capabilities, its accuracy is improved by DCNN's deep feature representation, which makes it ideal for intricate situations where accuracy of 97.8% is crucial. In light of this, the hybrid model of YOLOv7 and DCNN becomes an effective instrument for the early and accurate identification of leaf diseases, with great promise for raising agricultural output and promoting sustainable farming methods. Overall, this combination strategy opens the door for future developments in plant health monitoring technology while also improving the dependability of automated disease detection systems.

This system can be modified to identify diseases in other crops, enhancing global food security as the need for efficient and sustainable agricultural methods increases. Small-scale farmers may find the answer more accessible if real-time, on-field disease monitoring systems are made possible by developments in edge computing and IoT devices. Large-scale crop health surveillance may also be made easier by combining this hybrid model with drones or satellite data, and ongoing advancements in the YOLOv7 and DCNN architectures may further increase accuracy and lower processing costs. Researchers, agronomists, and legislators working together may potentially result in the creation of extensive disease control platforms that promote precision agriculture and data-driven decision-making globally.

6. REFERENCES

[1] Md. Riyadh Arefin, Md Ismail Hossain 2022. Present Status and Future Prospects of Tea production and Research on Varietal Improvement in Bangladesh.

[2] Hu Gensheng, Yang Xiaowei, Zhang Yan, Wan Mingzhu 2019. Identification of tea leaf diseases by using an improved deep convolutional neural network

[3] S. Gayathri, D.C.Joy Winnie Wise, P. Baby Shamini, N. Muthukumaran 2020. Image Analysis and Detection of Tea Leaf Disease using Deep Learning.

[4] Candra Nur Ihsan, Nova Agustina, Muchammad Naseer, Harya Gusdevi, Jack Febrian Rusdi, Ari Hadhiwibowo, Fahmi Abdullah 2024. Comparison of Machine Learning Algorithms in Detecting Tea Leaf Diseases

[5] Md. Janibul Alam Soeb, Md. Fahad Jubayer, Tahmina Akanjee Tarin, Muhammad Rashed Al amun, Fahim Mahafuz Ruhad, Aney Parven, Nabisab Mujawar Mubarak, Soni Lanka Karri & Islam Md. Meftaul 2023. Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)

[6] BaiShao Zhan, Xi Xiong, Xiaoli Li and Wei Luo1 2024. BHC-YOLOV8 : improved YOLOv8-based BHC target detection model for tea leaf disease and defect in real-world scenarios

[7] Jingyu Wang, Miaomiao Li, Chen Han and Xindong Guo 2024. YOLOv8-RCAA: A Lightweight and High-Performance Network for Tea Leaf Disease Detection

[8] Ananthakrishnan Balasundaram, Prem Sundaresan, Aryan Bhavsar, Mishti Mattu, Muthu Subash Kavitha, Ayesha Shaik 2025. Tea leaf disease detection using segment anything model and deep convolutional neural networks

[9] Samia Mehnaz[1] Md. Touhidul Islam 2025. Rice Leaf Disease Detection: A Comparative Study Between CNN, Transformer and Non-neural Network Architectures

[10] Jing Chen, Qi Liu, Lingwang Gao 2019. Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model.

[11] Hafijur Rahman, Iftekhar Ahmad, Parvej Hasan Jon, Abdus Salam & Md. Forhad Rabbi 2024. Automated detection of selected tea leaf diseases in Bangladesh with convolutional neural network

[12] Sakshi Pandey, Kuldeep Yogi, Ayush Ranjan 2024. Disease Detection In Rice And Wheat Leaves: A Comparative Study On Various Deep Learning Techniques

[13] Hu Gensheng, Fang Min. Using a multi-convolutional neural network to automatically identify small-sample tea leaf diseases

[14] Saikat Datta, Nitin Gupta 2023. A Novel Approach For the Detection of Tea Leaf Disease Using Deep Neural Network

[15] Qidong Heng, Sibo Yu, Yandong Zhang 2024. A new AI-based approach for automatic identification of tea leaf disease using deep neural network based on hybrid pooling

[16] Somnath Mukhopadhyay, Munti Paul, Ramen Pal & Debashis De 2021. Tea leaf disease detection using multi-objective image segmentation

[17] Bikash Chandra Karmokar, Mohammad Samawat Ullah, Md. Kibria Siddiquee and Kazi Md. Rokibul Alam 2024. Tea Leaf Diseases Recognition using Neural Network Ensemble

[18] Selim Hossain, Rokeya Mumtahana Mou, Mohammed Mahedi Hasan, Sajib Chakraborty, M. Abdur Razzak. Recognition and detection of tea leaf's diseases using support vector machine

[19] Yang Li, Yuheng Lu, Haoyang Liu, Jiahe Bai, Chen Yang, Haiyan Yuan, Xin Li, Qiang Xiao, 2024. Tea leaf disease and insect identification based on improved MobileNetV3

[20] Xiaoxiao SUN, Shaomin MU, Yongyu XU, Zhihao CAO, Tingting SU. Image Recognition of Tea Leaf Diseases Based on Convolutional Neural Network

[21] Xianze Yao, Haifeng Lin, Di Bai, and Hongping Zhou 2024. A Small Target Tea Leaf Disease Detection Model Combined with Transfer Learning

[22] Faruk Ahmed, Md. Taimur Ahad, Yousuf Rayhan Emon. Machine Learning-Based Tea Leaf Disease Detection: A Comprehensive Review

[23] Hang Zhaoy, Orazio Galloy, Iuri Frosioy, and Jan Kautzy 2016. Loss Functions for Neural Networks for Image Processing

[24] Juan R. Terven, Diana M. Cordova-Esparza, Alfonzo Ramirez-Pedraza, Edgar A. Chavez-Urbiola 2023. Loss Function and Metrics in Deep Learning. A Review

[25] B. Soujanya, T. Sitamahalakshmi 2020. Optimization with ADAM and RMSprop in Convolution Neural Network (CNN): A Case study for Telugu Handwritten Characters.