Proceedings of the Indian National Science Academy (2026) 92:1185–1203
https://doi.org/10.1007/s43538-025-00474-3

RESEARCH PAPER

Hybrid deep learning approach for early detection and classification of plant diseases from aerial view of agricultural lands

K. Sujatha[1] · N. P. G. Bhavani[2] · G. Rohini[3] · V. Srividhya[4] · A. Ganesan[5]

Received: 4 June 2024 / Accepted: 26 May 2025 / Published online: 28 June 2025
© Indian National Science Academy 2025

Abstract

This research presents a new combined deep learning system for effective and reliable identification of plant diseases in complicated agricultural environments. One of the most difficult jobs in agriculture is identifying plant diseases early on. Early disease detection in plants is crucial for increasing agricultural yield. With the application of machine learning and deep learning techniques, this issue has been resolved. Large crop farms can now detect plant illnesses automatically, which is advantageous as it reduces the monitoring time. The suggested approach consists of multiple important stages. To begin with, image quality of the agricultural lands is improved through preprocessing techniques like noise reduction, gamma correction and white balancing. Data augmentation is incorporated to expand the dataset and improve the generalization capacity of the model. Efficient methods such as EfficientDet, Squeeze Net, colour and shape based features, are included in feature extraction. The most relevant features are selected by a Hybrid Optimization Algorithm (HOA), which integrates Mother Optimization Algorithm (MOA), Teaching learning-based optimization (TLBO) and Improved Wild Horse Optimization (IWHO) to detect the various plant diseases like Bacterial Blight, Tungro, Blast and Brown spot. At last, a deep learning detector, which may include Recurrent Convolutional Neural Networks (RCNN) and Recurrent Neural Networks (RNN), Deep Neural Networks (DNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) identifies the location and type of plant diseases. Finally, HOA integrated with MOA, TLBO, IWHO feature selection and CNN classifier is proposed for identification of planst diseases from the hyper spectral images. Gradient Descent (GD) method is used for weight adjustment to tune the hyper parameters so as to reduce the loss function. It is also implemented to avoid over fitting and improve the overall generalization. This comprehensive approach depicts encouraging results in overcoming challenges in plant disease detection.

Keywords EfficientDet · Squeeze net · Plant diseases · MOA · TLBO and improved wild horse optimization · R-CNNs and recurrent neural networks

✉ K. Sujatha
sujathak73586@gmail.com

N. P. G. Bhavani
sbreddy9999@gmail.com

G. Rohini
rohinimukunthan@gmail.com

V. Srividhya
sripranav2007@gmail.com

A. Ganesan
dragmephd@gmail.com

[1] Department of EEE, Dr. M.G.R. Educational and Research Institute, Chennai, Tamil Nadu, India

[2] Department of ECE, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India

[3] Department of EEE, Saveetha Engineering College, Chennai, Tamil Nadu, India

[4] Department of EEE, Meenakshi College of Engineering, Chennai, Tamil Nadu, India

[5] Department of EEE, RRASE Engineering College, Chennai, Tamil Nadu, India

Introduction

The need for efficient analysis arises due to the daily accumulation of large amounts of aerial and satellite imagery; and presenting a new method in our letter that uses content-based techniques to automatically identify and annotate objects or areas in high-resolution aerial images (Ševo and Avramović 2016) and innovate Convolutional Neural Network (CNN) models for detecting important ground features in aerial disaster images, such as buildings, vehicles, vegetation, debris, and floods (Pi et al. 2020).The identification of objects in aerial images is difficult due to differences in size and direction, as well as a shortage of comprehensive standards, leading to limited advancements in this area (Ding et al. 2021). Presents a rapid, accurate vehicle detection system employing Adaptive Virtual Private Network (AVPN) to achieve precision for small objects, and a combined R-CNN for simultaneous identification of vehicle location and characteristics (Deng et al. 2017).Identifying various objects in remote sensing images is difficult, and deep learning methods such as Full Resolution Convolutional Network (FRCN) are showing potential despite facing application challenges (Deng et al. 2018).

When using Earth Vision, accurately identifying, and classifying objects in aerial photos typically involves the use of OBBs. However, this can lead to problems with uncertainty in defining the target, which can impact the efficiency and accuracy of the process (Wang et al. 2020a). Suggests the use of CNNs to examine Unmanned Aerial Vehicle (UAV) images, divided into tiles, for the classification of multiple labels using radial basis function networks and a unique multi-labelling layer, demonstrating encouraging outcomes (Zeggada et al. 2017). Presenting YOLOs, designed specifically for analysing multiple types of remote sensing imagery, providing a range of detection capabilities at different scales, the ability to detect small targets, predicting orientations, and featuring a unique mid-level fusion architecture (Sharma et al. 2020) and presenting "Focus-and-Detect," a two-stage approach to addressing the detection of small objects. This framework includes a clustering object detector guided by Gaussian distribution and specific prediction for focal regions. Additionally, it incorporates Incomplete Box Suppression to handle challenges related to object truncation (Koyun et al. 2022).Presenting a point-based estimator design that utilizes spatial data through point representation and a convolutional network to improve localization precision and differentiate recognition by aligning instances to achieve feature-to-region congruence (Fu et al. 2020).

Feature Fusion Deep Networks (FFDN), improves the detection of aerial images by using a combination of structural learning layer to enhance the spatial relationship between the objects. This has shown significant progress on UAV123 and UAVDT datasets (Long et al. 2019). A Network for Multi-Scale Object Detection in Aerial images (mSODANet) is a network specifically created for the purpose of identifying objects within aerial images. Hierarchical dilated convolutions are utilized to analyse context from various perspectives and scales, ultimately enhancing the ability to capture and detect visual data effectively (Chalavadi et al. 2022) and suggested the use of Adaptive Saliency Biased Loss to improve DNN object detectors. This method involves adjusting the weight of training examples and anchors based on saliency metrics at the image or anchor levels (Sun et al. 2020) and presenting DHRec encoding for objects with multiple orientations, arranging the coordinates of vertices, creating two horizontal boxes, and using area ratios to guide the decoding from predicted double horizontal rectangles (Nie and Huang 2022).Propose using a Scale Decoupling Module in the early layers to underline small objects while suppressing large ones, alongside Adaptive Anchor Matching Strategy and Sparse Nonlocal Attention to improve detection accuracy efficiently (Ma et al. 2023).

The main key elements of this paper are as follows:

- Preprocessing is carried out by Image enhancement techniques such as noise reduction via Bilateral Filtering, Gamma Correction via contrast adjustment, and White Balancing with Grey Card Reference via color normalization are applied to improve image quality and facilitate feature extraction.
- Data augmentation is carried out by the strategies, including geometric transformations (rotation, flipping, scaling).
- For to extract the Features from the image, EfficientDet and Squeeze Net; Color based features like Color Coherence Vector and Color Correlograms; and Shape based features are used.
- To choose the best features a hybrid Optimization Algorithm is designed by the combination of mother optimization algorithm, TLBO and Improved wild horse optimization with levy flight algorithm Deep Learning-Based Detection:

2. For to detect the deep learning-based detection, potentially constructed using region-based CNNs (R-CNNs), RNN and Bi-LSTM is used to localize and classify objects based on the selected features and the hyper-parameter tuning via hybrid deep learning techniques are applied to prevent over-fitting and improve generalization.

The study is integrated into various parts: The first section presents the introduction, the second section includes a literature review, the third section details the proposed

methods, the fourth section analyses the results and their significance, and lastly, the fifth section wraps up the research.

Literature review

In 2022, Wang et al. (Wang et al. 2022) proposed the Adaptive Recursive Feature Pyramid, which encompasses a recursive structure, an Efficient Global Context bottleneck, and a Discriminative Feature Fusion module. This design allowed for efficient utilization of semantic and location information across pyramid levels by using dense connections and adaptive weighting. Extensive testing on VisDrone2019 and DIOR datasets showed that ARFP outperformed existing models, with average performance gains of 2.7% and 2.8% respectively.

In 2020, Wang et al. (Wang et al. 2020b) created a new method for training and making inferences, significantly improving the accuracy of detecting objects in high-resolution aerial photographs. This method, different from straightforwardly regressing object orientations, used segmentation tasks to make more accurate predictions of rotated bounding boxes. Paired with a strategy that synthesizes images for data augmentation, it effectively tackled the challenges of data imbalance and achieved remarkable performance on the Dataset of Object deTection in Aerial images (DOTA) dataset through extensive experimentation.

In 2021, Han et al. (Han et al. 2021) presented the Single-shot Alignment Network comprising a Feature Alignment Module and an Oriented Detection Module (ODM). The FAM utilized a new Alignment Convolution method to improve anchoring and adjust features, while the ODM employed rotating filters to encode orientation, thereby enhancing the alignment between classification and localization. This approach also enhanced the identification of objects in extensive images, finding a more favorable balance between quickness and precision.

In 2024, Ge et al. (Ge et al. 2024) suggested an adaptable dynamic labelling method to enchance the detection of small objects by adjusting the allocation thresholds for samples according to the specific characteristics of each ground truth. This method, using dynamic IoU, improved training by adjusting sample sets, leading to a significant improvement in performance on AI-TOD and Levir-Ship datasets without increasing computational expenses. Our approach has exceeded the current standards in both efficiency and accuracy.

In 2024, Weng et al. (Weng et al. 2024) presented the Selective Frequency Interaction network, which is a new method for detecting aerial objects. It includes two main components: Selective Frequency-domain Features Interaction and Selective Frequency-domain Feature Extraction. The SFFE module excelled in extracting important frequency-domain information and it significantly improved the richness of feature maps at various frequencies.

In 2018, Xu et al. (Xu et al. 2018) presented a new framework for detecting moving objects, which improves accuracy by extracting selected feature points and predicting registration accuracy. Drawing on input from previous iterations, the inaccuracies are removed and incorporated a measurement to anticipate registration precision, providing guidance for incorporating supplementary indicators. Our study on feature extractors for registering aerial infrared images found that SURF and LDB are highly effective.

In 2021, Xu et al. (Xu et al. 2021) tackled the issue of feature misalignment by introducing a streamlined single-shot detector that incorporates a pseudo anchor proposal module and a context-based feature alignment module. The PAPM enhanced the primary anchors for aligned positions, addressing spatial misalignment and imbalance issues. Following this, CFAM adjusted the sampling points of the convolution kernel using the main anchors, effectively resolving any misalignment between the kernel and the object and extracting powerful, aligned features.

In 2022, Xu et al. (Xu et al. 2022) created a new evaluation method known as Normalized Wasserstein Distance, combined with a Ranking-based Assigning approach, in order to enhance the identification of small objects. The NWD-RKA approach, which can be used with various anchor-based detectors, has replaced traditional IoU threshold-based methods, leading to substantial enhancements in label assignment and offering extensive training supervision. After conducting tests on four separate datasets, it was evident that NWD-RKA significantly improved the detection of small objects.

In 2022, Hu et al. (Hu et al. 2022) suggested a training method called Correction Maximization Training, which focuses on removing incorrect annotations from datasets before training a model. We developed a new noise filter called the Probability Differential as part of our strategy to accurately identify and correct errors in labels. After being cleaned, the detector was trained using the revised dataset. This method showed that it works well with popular object detectors like Faster RCNN and Retina Net, requiring few changes to hyperparameters when used with different datasets and models.

In 2020, Wang et al. (Wang et al. 2020a) suggested to change the method of OBB regression by instead predicting center-probability maps. This would help to minimize uncertainties related to identifying targets and background pixels. The use of the forecasted CenterMaps helped in creating OBBs. The Center Map OBB representation stood out for its simplicity and effectiveness. Additionally, we have incorporated a weighted pseudo segmentation-guided attention network to enhance the distinction between the desired

objects and the densely populated background. This also aids in predicting both horizontal bounding boxes and OBBs.

Problem statement

Depending on the advantages and disadvantages resulting illustrated in Table 1 from the previous work, the problem statement is identified. The major objective of this work is to detect the plant diseases from the aerial images captured by the surveillance drones used for incorporating smart technology in agriculture.

Proposed methodology

To enhance the initial quality and feature extraction from the images of the agricultural lands, a series of preprocessing steps are incorporated, including Bilateral Filtering for diminishing noise without blurring edges, Gamma Correction for optimizing contrast, and Grey Card Reference-based White Balancing for achieving accurate color representation. To mitigate the risk of overfitting and bolster the algorithm's ability to generalize across diverse data, we apply data augmentation techniques such as rotations, mirroring, and scaling. In the feature extraction phase, a dual approach is utilized as in Fig. 1 by combining EfficientDet and SqueezeNet for capturing intricate details alongside implementing Color Correlograms and Color Coherence Vectors for color attributes, and geometric metrics like Area, Perimeter, and Eccentricity for shape attributes. To ensure the selection of the most impactful features, a novel hybrid optimization algorithm is introduced by merging the MOA with an Enhanced Wild Horse Optimization (EWHO) integrated with TLBO, fine-tuning the feature selection process for optimal performance. For the object detection mechanism, we integrate a sophisticated deep learning framework that combines the strengths of region-based CNN, BiLSTM and Recurrent Neural Networks. This ensemble is meticulously designed to localize and categorize objects with high precision. Additionally, to optimize the model's accuracy and robustness, a unique combination of hybrid deep learning methodologies for hyperparameter optimization, effectively minimizing over fitting while maximizing generalization is employed across unseen plant images as in Fig. 2.

Step 1: Preprocessing

- Image enhancement techniques such as noise reduction via Bilateral Filtering, Gamma Correction via contrast adjustment, and White Balancing with Grey Card Reference via color normalization are applied to improve image quality and facilitate feature extraction.
- Data augmentation strategies, including geometric transformations (rotation, flipping, scaling), are employed to artificially expand the dataset and enhance the model's generalization capability.

Step 2: Feature extraction

- EfficientDet and Squeeze Net
- Color based features: Color Correlograms and Color Coherence Vector
- Shape based features: Area, Perimeter, Eccentricity

Step 3: Feature selection

- An algorithm that combines different optimization techniques is created to choose the most important features from the pool that has been extracted. The hybrid optimization algorithm is the combination of mother optimization algorithm, TLBO and Improved wild horse optimization.

Step 4: Deep learning-based detection

- A deep learning detector, potentially constructed using region-based CNNs, RNN and Bi-LSTM is employed to localize and classify the disease specific variations based on the selected features.

Table 1 Advantages and drawbacks of the previous research work

Author	Method	Advantages	Drawback
Xu et al. (Xu et al. 2021)	Aligned single-shot detection with pseudo anchor proposal	Fixes spatial misalignment; improves feature alignment robustly	Complexity in real-time application scenarios
Xu et al. (Xu et al. 2022)	Normalized Wasserstein Distance with Ranking-based Assigning	Enhances small object detection; optimizes label assignment	May require extensive computational resources
Hu et al. (Hu et al. 2022)	Correction Maximization Training with Probability Differential	Improves model accuracy	Initial filtering process can be time-consuming
Wang et al. (Wang et al. 2020a)	Center-probability maps for OBB regression	Simplifies OBB creation; reduces target-background confusion	Limited to specific types of object detection tasks

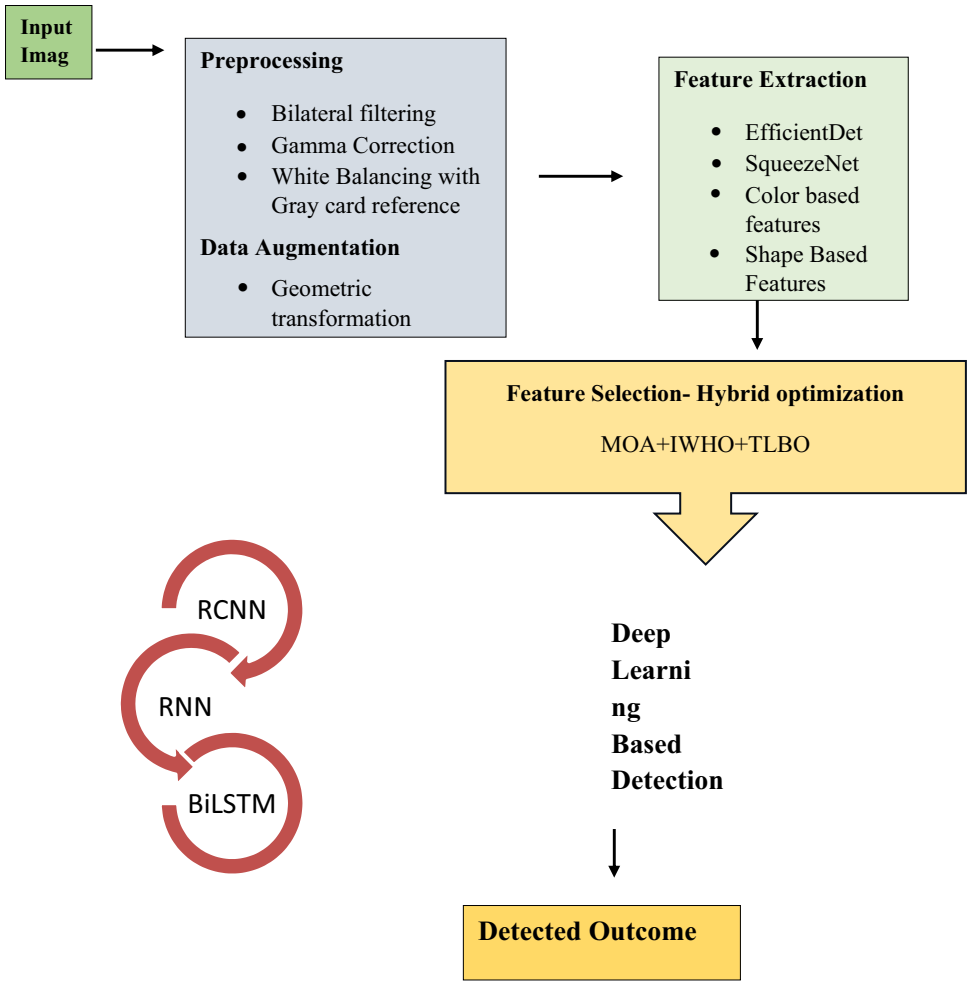


Fig. 1 Overall architecture

- Hyper-parameter tuning via hybrid deep learning techniques are applied to prevent overfitting and improve generalization.

Preprocessing

During the initial processing stage, improve the quality of the image by employing various enhancement techniques such as utilizing Bilateral Filtering to suppress noise, implementing Gamma Correction to optimize contrast, and enhancing color accuracy with the use of a Grey Card Reference for white balancing. Furthermore, enhance the dataset and the model's flexibility by utilizing data augmentation techniques.

Bilateral filtering

The Bilateral filter is considered a peak choice for spatial domain filtering, with the main goal of removing noise from images. It effectively targets and removes additive Gaussian noise by accurately predicting the value of a given pixel. This is achieved by considering both the spatial proximity and intensity similarity of pixels within a localized vicinity. This method of filtering uses Gaussian distribution equations to determine weights, considering the geometric and photometric connections between pixels.

If the input image 'f' is noisy, then the output can only be produced as given in Eq. 1 through spatial domain filtering below:

$$H_1(X) = K_D^{-1}(X) \iint f(\varepsilon), e(\varepsilon, X) d\varepsilon \tag{1}$$

Here,$e(\varepsilon, X)$ represents the geometric distance between the central point X and its neighboring points $\prime\varepsilon\prime$. Furthermore, when focusing solely on the variations in intensity within a small local area, the filtering process generates an output defined by Eq. 2.

$$H_2(X) = K_R^{-1}(X) \iint f(\varepsilon) \cdot S(f(\varepsilon) f(X) d\varepsilon \tag{2}$$

$S(f(\varepsilon)f(X)$ indicates the photometric distance between the centre x and its neighboring points $\prime\varepsilon\prime$. Integration of

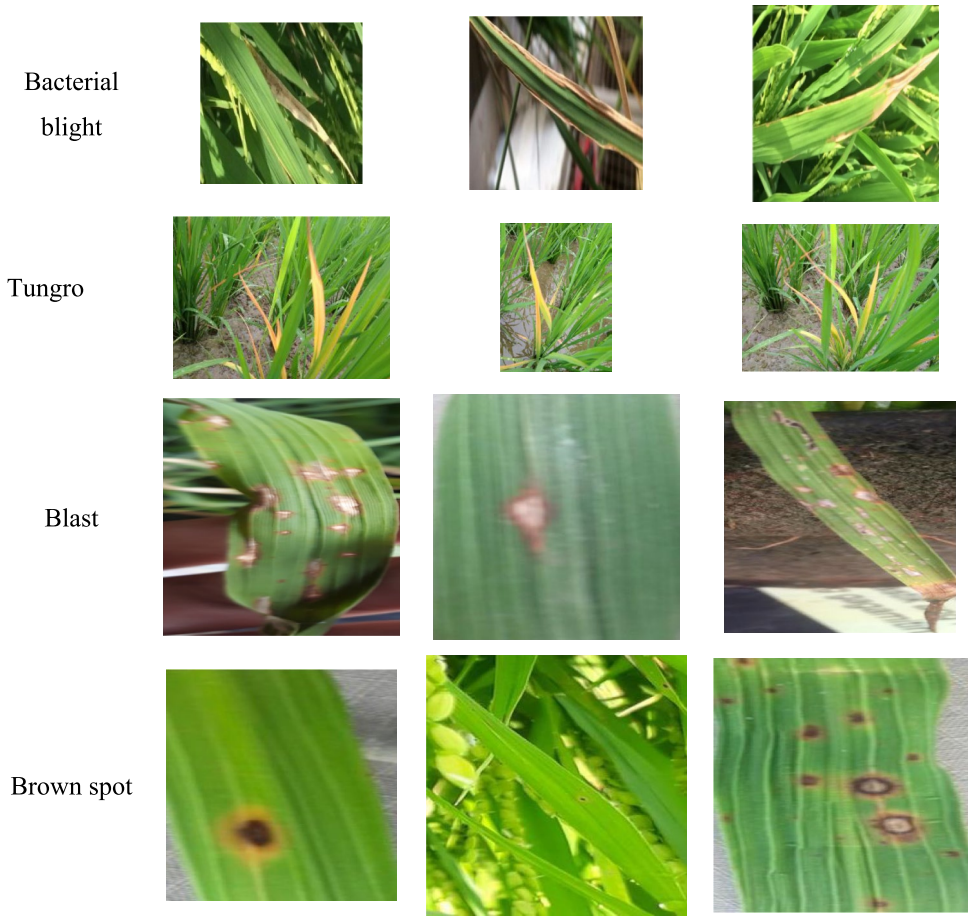


Fig. 2 Images of the plants affected by various diseases

the spatial and intensity distances of pixels within a local neighborhood, yields an output as specified in Eq. 3.

$$H(X) = K^{-1}(X) \iint f(\varepsilon) \cdot e(\varepsilon, X) \cdot S(f(\varepsilon) f(X) d\varepsilon \tag{3}$$

$K(X)$ represents the normalization constant as shown in Eq. 4.

$$K(X) = \iint e(\varepsilon, X) \cdot S(f(\varepsilon) f(X) d\varepsilon \tag{4}$$

In the standard bilateral filter designed to mitigate additive Gaussian noise, both the geometric distance e and the photometric distance S are modeled to adhere to Gaussian distributions, as described in Eqs. 5 and 6 respectively.

$$e(\varepsilon, X) = \exp\left(\frac{-1}{2}\left(\frac{d(\varepsilon, X)}{\sigma_d}\right)^2\right) \tag{5}$$

$$S(\varepsilon, X) = \exp\left(\frac{-1}{2}\left(\frac{d(f(\varepsilon) f(X)}{\sigma_R}\right)^2\right) \tag{6}$$

Euclidian distance is denoted as $\prime d \prime$.

Gamma correction

The process known as gamma correction adjusts the brightness levels of an image through a non-linear transformation. By altering the gamma value, one can achieve different degrees of image enhancement. Similar to gamma correction, logarithmic transformations are effective in brightening an image's dark regions, enhancing contrast especially in low-light areas. Such transformations are particularly valuable for emphasizing details in darker regions. When the gamma value is set below 1, this technique expands the range of darker pixels while compressing the range of brighter pixels, based on a fundamental power law in Eq. 7

$$S = CR^{\gamma} \tag{7}$$

The positive constant is $Candy\gamma$, The above equation can be represented in an alternative form given in Eq. 8.

$$S = C(R + \varepsilon)^{\gamma} \tag{8}$$

Gamma correction, also known as gamma adjustment, is a technique used to alter the luminance of an image. The principle behind this is to modify the input signal based on a power-law relationship, which can be adjusted through the

gamma value. This value can range from 0 to infinity, influencing how the image's brightness is adjusted. A gamma of 1 implies a linear relationship between input and output, maintaining the original brightness levels. When gamma is set below '1', the image becomes brighter, as this setting biases the adjustment towards higher, more luminous output values. Conversely, a gamma value above 1 darkens the image by favouring lower output values, thus making it suitable for correcting images that appear too bright or too dark.

White balancing with gray card reference

White balance in digital photography is typically achieved by using a white piece of paper as a reference under the subject's ambient light. This method involves adjusting the camera settings to ensure that white appears truly white, maintaining the neutrality of all other colours in the photo. Essentially, white balance is used to correct the colour temperature of a photograph, ensuring that the colours are rendered accurately based on the lighting conditions at the scene. On the other hand, colour balance involves adjusting the colours to compensate for colour casts caused by varying lighting conditions. This process might not always be straightforward, as different algorithms handle the nuances of colour correction in diverse ways. The objective of colour balancing can vary: Some techniques aim to replicate the scene's colours as faithfully as possible, while others might focus on enhancing the aesthetic appeal of the image. This variation can sometimes complicate the understanding and implementation of effective colour or white balancing techniques in photography. A gray card serves as a crucial tool in video production and photography, offering a standard middle gray reference which aids in achieving consistent exposures and accurate colour settings. This neutral gray tool has an even spectral reflectance, making it ideal for use with reflective light meters. Beyond its role in exposure measurement, a gray card also acts as a key reference point for setting white balance, helping cameras adjust to the colour temperature of different lighting conditions. Whether utilized during a shoot for in-camera white balance adjustments or in post-processing to fine-tune colour balance, a gray card is essential for maintaining colour fidelity and exposure accuracy in visual media.

Data augmentation—geometric transformations (rotation, flipping, scaling)

Data augmentation techniques, including geometric transformations such as rotation, scaling and flipping are applied to enhance the diversity of the training set.

Scaling

Image scaling involves adjusting the dimensions of a plant image by enlarging or reducing it according to specific scale factors. This resizing process utilizes equations that map the original pixel coordinates (X, Y) to new coordinates $(X\prime, Y\prime)$ in the resized plant image. The scaling factors, $S_X and S_Y$ dictate the proportion of change along the image's width (rows) and height (columns), respectively. These factors determine how each pixel in the original image is positioned and represented in the scaled version, affecting the overall size (Eqs. 9 and 10) of the output image.

$$X' = X \cdot S_X \tag{9}$$

$$Y' = Y \cdot S_Y \tag{10}$$

Interpolation is employed during image scaling to refine object edges and preserve the aspect ratio when the scale factors for both dimensions, $S_X and S_Y$, are equal. This technique enhances model robustness by improving performance across different image sizes. Upscaling involves expanding an image and then cropping it back to its original dimensions. Conversely, when downscaling, the image is reduced while maintaining its original size, with the extra space filled using techniques such as nearest neighbor to ensure no loss of visual information.

Rotation

Image rotation, widely used in data augmentation for computer vision, manipulates an image by rotating it through an angle ranging from 0 to 360 degrees, thereby generating diverse training samples. This transformation is guided by equations that map original pixel coordinates (X, Y) to new positions $\left(X', Y'\right)$ in the rotated frame, with θ representing the rotation angle in radians and $(C_X and C_Y)$ denoting the center of the image. This process enriches the dataset and enhances the model's ability to generalize across varied orientation.

$$X' = \left(X - C_X\right)\cos\theta - \left(Y - C_Y\right)\mathrm{Sin}\theta + C_X \tag{11}$$

$$Y' = \left(X - C_X\right)\cos\theta - \left(Y - C_Y\right)\mathrm{Sin}\theta + C_Y \tag{12}$$

Like image scaling, image rotation can also create empty spaces in the image, which are typically filled using interpolation methods to ensure a seamless visual transition and maintain image quality. The expression for image rotation is illustrated in Eq. 11 and 12 respectively.

- Flipping

  Image flipping depicted in Eq. 13, involves swapping the left and right sides of an image. In hand gesture recognition, typically only horizontal flipping is applied. The process for horizontal flipping is described by a formula where (h, Y) represent the original pixel coordinates, h' is the flipped pixel position in the image, and w denotes the image's width.

$$h' = (w - 1) - h \tag{13}$$

For horizontal flipping, each row of pixels in the image is reversed from left to right. This method does not require any interpolation as it simply mirrors the image across its vertical axis.

Feature extraction

Feature extraction techniquess employed include EfficientDet and SqueezeNet for image analysis, while colour-based attributes such as Colour Coherence Vector and Colour Correlograms, alongside shape-based features like area, perimeter, and eccentricity, are utilized to enhance object detection and classification accuracy.

- EfficientDet

  EfficientDet stands out due to its innovative architecture, drawing on the strengths of EfficientNet as its backbone to capture complex features from challenging datasets, such as those involving various forest fire scenarios. At the core of EfficientDet's structure is the bi-directional feature pyramid network (Bi-FPN), an enhanced version of PANet that acts as the model's neck. Bi-FPN improves the integration of multi-scale features through efficient, bidirectional feature fusion. It incorporates learnable weights that evaluate the importance of different input features, enabling a more parameter and computation-efficient fusion process compared to the standard PANet utilized in Yolov5.EfficientDet also employs a compound scaling method, similar to that of EfficientNet, which strategically scales the resolution, depth, and width across the backbone, the feature network, and the prediction modules for both boxes and class data. This methodical scaling enhances the model's performance and efficiency, adjusting to the computational resources at hand and allowing for incremental accuracy enhancements as more resources become available. By integrating a unique mix of backbone, neck, and head components, EfficientDet excels at processing and learning diverse information, potentially surpassing the capabilities of models like Yolov5, as demonstrated in Fig. 3.

- SqueezeNet

  SqueezeNet is meticulously crafted to minimize its parameter footprint, implementing several effective strategies. Primarily, it replaces standard 3x3 filters with smaller 1x1 filters, which are less computationally intensive. Additionally, it reduces the number of input channels to 3x3 filters, and strategically delays downsampling until later in the network, allowing for larger activation maps earlier in the processing sequence. A distinctive feature of SqueezeNet is its "Fire Layers," each constructed of "Fire Modules," illustrated in Figure 4. Within each Fire Module, a squeeze layer equipped solely with 1x1 filters channels its output to an expand layer that merges both 1x1 and 3x3 filters. This architecture enables the adjustment of three key hyperparameters: s1x1, e1x1, and e3x3, where s1x1 indicates the number of 1x1 filters in the squeeze layer, and e1x1 and e3x3 are the counts of 1x1 and 3x3 filters in the expand layer, respectively. This design ensures the model remains highly efficient in its

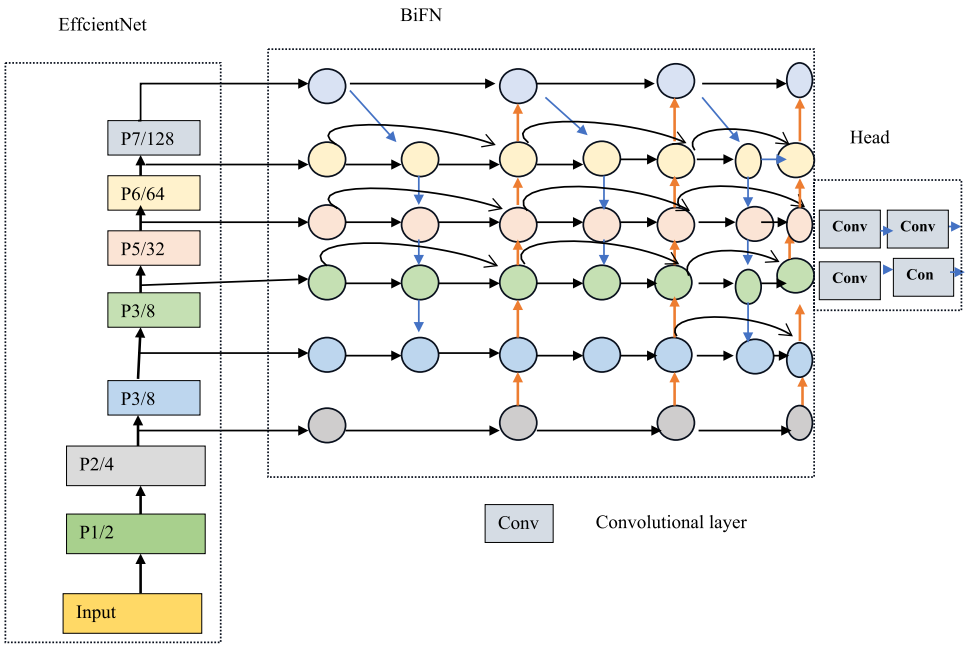


Fig. 3 EfficientDet

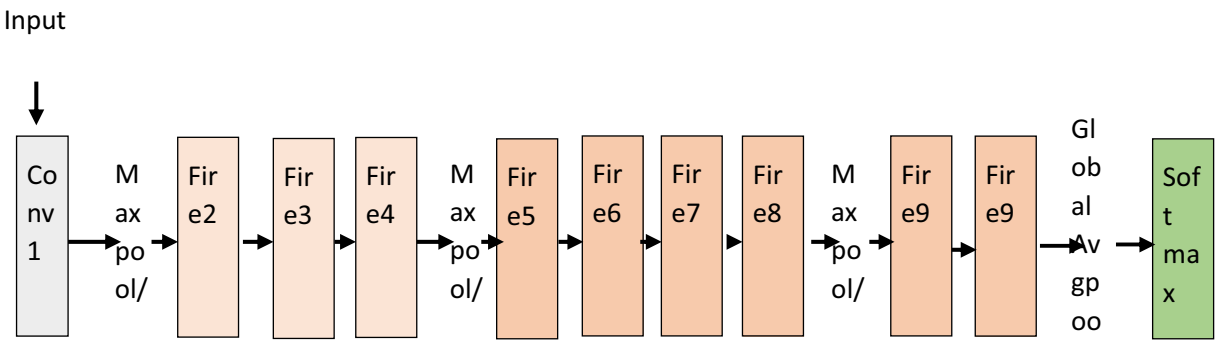


Fig. 4 Squeeze Net

operations, SqueezeNet achieves impressive performance despite its small size and has been widely adopted across various applications.

- Colour based features

  Colour-based features such as Colour Coherence Vector and Colour Correlograms are utilized to analyse and describe the distribution and relationship of colours within an image.

- Colour coherence vector

  The Vector of Colour Coherence extends traditional colour histograms by splitting the colour representation into two components: Coherent and incoherent vectors. Coherence classification hinges on a predefined threshold, τ. If the size of a color region surpasses τ, it is classified as coherent; otherwise, it is deemed incoherent. This approach helps in distinguishing between large, uniform colour areas and scattered, non-uniform ones.

  Initially, apply a blurring technique to mitigate excessive separation by averaging the values of each pixel with those of its eight surrounding neighbours. Subsequently, construct both coherent and incoherent vectors using a predetermined threshold value, τ, to categorize regions based on their size and uniformity.

  Define α_N as the count of coherent pixels and β_N as the count of incoherent pixels for each color segment. Each color segment is then represented by the sum $\alpha_J + \beta_J$. Consequently, the colour coherence vector for the image is formulated using these sums for each colour in Eq. 14.

$$< (\alpha_1 + \beta_1), \ldots (\alpha_N + \beta_N) > \tag{14}$$

  The comparison of images I and I′ using the colour coherence vector involves calculating the similarity or differences between their vectors as in Eq. 15, 16 and 17 respectively.

$$ccv_I =< (\alpha_1 + \beta_1), \ldots (\alpha_N + \beta_N) > \tag{15}$$

$$ccv_{I'} =< (\alpha'_1 + \beta'_1), \ldots (\alpha'_N + \beta'_N) > \tag{16}$$

$$\Delta g = \sum_{N=1}^{N} \left|(\alpha_N - \alpha'_N)\right| + \left|(\beta_N - \beta'_N)\right| \tag{17}$$

- Colour correlograms

  The colour correlogram presents a nuanced approach to analysing images by examining both the frequency of colour appearances and the spatial relationship between different colours. It is essentially a three-dimensional histogram where two dimensions represent a pair of pixel colours, and the third dimension captures the spatial separation between them. At the core of a colour correlogram is a matrix indexed by pairs of colours, with each cell in the matrix (for a colour pair (C_I, C_J) and a predefined distance 'D') revealing the likelihood of encountering a pixel of color C_J at the specified distance d from a pixel of color C_I within the image. This method offers a comprehensive way to understand the intricate spatial interplay of colours in an image's composition using Eq. 18.

$$\gamma^{(K)}_{C_I,C_J} = P_1 \in i^{pR}_{C_I \in i}\left[P_2 \in i_{C_J} \middle| |P_1 - P_2| = K\right] \tag{18}$$

  When considering the full scope of colour correlations across a plant image, the pairings between colours (C_I, C_J) from a set of M distinct colors, and distances K within a range from $1 to D$, the calculation for a complete color correlogram can become computationally intensive, growing in complexity to $O(M^2D)$ This exhaustive approach, which measures the distance $|P_1 - P_2|$ between any two pixels P_1 and P_2, is often streamlined into a more manageable form known as the color autocorrelogram. Unlike its more complex counterpart that

accounts for every possible colour pair, the colour auto-correlogram focuses solely on the spatial correlation of matching colours. This simplification significantly reduces the computational load to ($O(MD)$), offering a more efficient yet still effective method for capturing the essence of spatial colour relationships in an image as in Eq. 19.

$$\alpha_{C_I}^{(K)}(i) = \gamma_{C_I,C_J}^{(K)}(i) \tag{19}$$

Choosing a larger d in colour correlogram analysis heightens computational and storage demands, while a smaller d may simplify features too much. Although quantizing colours (M) to 64–256 broadens analysis, it also increases complexity. The color auto-correlogram, though resource-intensive, excels in image retrieval, making correlograms essential for detailed image analysis.

- Shape based features

  Shape-based features such as area, perimeter, and eccentricity are crucial for analysing and classifying geometric forms. The area measures the space contained within the shape, the perimeter quantifies the length around it, and eccentricity evaluates how much the shape deviates from being a perfect circle.

- Eccentricity

  Eccentricity given in Eq. 20, is a scalar metric describing the shape of an ellipse based on its second moments, matched to a particular region. It calculates the ratio between the focal distance (distance between the ellipse's foci) and the length of its major axis, with values ranging from 0 to 1.

$$ecc = \sqrt{1 - \left(\frac{B^2}{A^2}\right)} \tag{20}$$

  Here, A is the semi-major axis, B is the semi-minor axis, and C is the distance from the center to a focus.

- Area

  Area represents the total count of pixels that form a specific region, where pixels are either part of the shape (typically represented as 1) or not (typically represented as 0), the formula to calculate the area of a shape is given by Eq. 21.

$$area = \sum_{I=1}^{M}\sum_{J=1}^{N} i(I,J) \tag{21}$$

  $i(I,J)$, shows the pixel value at the I^{th} row &J^{th} column of the image matrix, where $=i(I,J) = 1$ if the pixel belongs to the shape and $i(I,J) = 0$ otherwise .$MandN$ are the dimensions of the image in terms of rows and columns, respectively.

- Perimeter

  The perimeter of a shape in image analysis is calculated by summing the distances between consecutive pairs of edge pixels that outline the shape. For shapes represented in digital images, this typically involves adding the lengths of the line segments connecting these adjacent border pixels. The formula for the perimeter P of a shape can be expressed in Eq. 22.

$$peri = \sum_{K=1}^{n} D(P_K, P_{K+1}) \tag{22}$$

  $P_K and P_{K+1}$ are the consecutive pixels on the border of the shape. 'D' is the distance.

Feature selection (Hybrid optimization algorithm)

A hybrid optimization algorithm that combines the features of Teaching–Learning Based Optimization, Mother Optimization Algorithm, and Wild Horse Optimization can lead to an innovative approach that leverages the diverse strengths of these individual strategies. This integration aims to enhance global exploration and local exploitation capabilities, making it robust for complex optimization problems. Below is a step-by-step development of this hybrid model, incorporating key elements and equations from TLBO, MOA, and WHO.

Step 1: Initialization

A population-based meta-heuristic optimization algorithm operates by evolving a set of candidate solutions over multiple iterations. The solutions are conceptualized as vectors within a defined problem space, with their collective configurations forming a matrix (Eq. 23). At the inception of the algorithm, this population matrix is populated according to an initialization formula (Eq. 24). Each vector, representing a unique solution, specifies potential values for the decision variables of the problem. Throughout the optimization cycle, the collective capabilities of these vectors are harnessed to navigate and converge towards an optimal solution efficiently. The initial population was divided into multiple groups, indicated by x_I. Each group includes a leader known as a stallion, making the total number of stallions. The remainder of the population, consisting of foals and mares, was evenly distributed across these groups.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_I \\ \vdots \\ x_n \end{bmatrix}_{n\times M} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,J} & \cdots & X_{1,M} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ X_{I,1} & \cdots & X_{I,J} & \cdots & X_{I,M} \\ \vdots & \cdots & \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,J} & \cdots & X_{n,M} \end{bmatrix}_{n\times M} \quad (23)$$

$$X_{I,J} = LB_J + rand(0,1)\cdot\left(UB_J - LB_J\right), I = 1,2,\ldots,n; J = 1,2,\ldots,M \quad (24)$$

'x', denotes the matrix that holds all the candidate solutions, where n is the total number of solutions in the population, and M denotes the count of decision variables for each solution. Each solution in the matrix,x_I, is the *Ith* candidate, described by a vector$(X_{I,1},\ldots X_{I,J},\ldots X_{I,M})$, where $X_{I,J}$ is the *Jth* variable of the ith solution. The function *rand*(0,1) is utilized for random number generation uniformly distributed between 0 and 1. The domain of each decision variable, j, is defined by its lower bound,LB_J, and upper bound,UB_J.Consider a scenario where there are M different subjects available to N learners (representing the population size, with learners numbered$K = 1toN$). During each teaching–learning cycle, labelled$I$, the variable $M_{J,I}$ represents the average performance in a specific$J$. In this model, the most adept and knowledgeable learner across all subjects during a cycle is designated as the teacher. This top performer is denoted by $X_{total-Kbest,I}$, representing the highest achieving learner or the 'teacher' for that specific cycle.

Step 2: Fitness evaluation

Calculate the fitness of each candidate solution using the objective function for all populations represented in Equation 25.

$$f = f\left(X_I\right) \quad (25)$$

Step 3: Teaching phase

The teacher strives to enhance overall class knowledge, yet learner gains depend on both teaching quality and learner calibre. Differences in performance between the teacher's results and the average outcomes in each subject are calculated using Equation 26.

$$difference_mean_{J,I} = R_I\left(x_{J,Kbest,I} - t_f M_{J,I}\right) \quad (26)$$

In subject, $x_{J,Kbest,I}i$ s the top learner's score, acting as the teacher. The Teaching Factor (TF), either 1 or 2, randomly modifies means, influenced by a random number R_I between [0, 1], depicted in Eq. 27.

$$t_f = Round[1 + Rand(0,1)\{2-1\}] \quad (27)$$

During the teacher phase, the solution is updated using the Equation 28

$$x'_{J,K,I} = x_{J,K,I} + difference_mean_{J,I} \quad (28)$$

The updated variable $x'_{J,K,I}$ is retained if it improves performance, serving as the basis for the learner phase progression.

Step 4: Combine MOA and WHO search mechanisms

Following the TLBO teaching phase, merge the Multi-Objective Algorithm (MOA) with the Wild Horse Optimization (WHO) strategy to enhance search space exploration and exploitation efforts.

In the "Education" phase of the MOA approach, the best population member, termed the "mother," guides significant changes in others' positions for enhanced global search and exploration, as per Equation 29. Successful position changes improving objective scores are updated using Equation 30.

$$X^{p1}_{I,J} = X_{I,J} + Rand(0,1)\cdot\left(m_J - Rand(2)\cdot X_{I,J}\right) \quad (29)$$

$$x_I = \begin{cases} x_I^{p1}, f_I^{p1} \le f_I \\ x_I, else, \end{cases} \quad (30)$$

In the MOA's first phase, the mother's position m_J influences each member x_I's position $X_{I,J}$, generating a new position x_I^{p1}, with dimensions $X^{p1}_{I,J}$, guided by random functions *rand*(0,1) and *rand*(2) to assess changes using objective value f_I^{p1}.

Equation 31 is used for modelling the grazing behaviour, simulating how the population optimizes resource utilization during the search process.

$$x^J_{I,g} = 2z\cos(2\pi rz)\times\left(stallion^J - x^J_{I,g}\right) - stallion^J \quad (31)$$

The equation depicts the current position of foal/mare group members influenced by the stallion's position, a random factor from -2 to 2, and adaptive dynamics.

Step 5: Learner phase

During the learner interaction phase of the algorithm, students enhance their understanding by engaging with peers. Two learners, 'p' and 'q', are randomly chosen, ensuring they have distinct knowledge levels from the recent teaching phase. Here, $x'_{Total-p,I} \neq x'_{Total-q,I}$ represent the new knowledge states of learners, 'p' and 'q', respectively, differentiated from their previous states, $x_{total-p,I}$ and $x_{total-q,I}$. This process stimulates peer learning, where students benefit from the diverse insights of others, given by Eq. 32 and Eq. 33.

$$x''_{J,p,I} = x'_{J,p,I} + R_1\left(x'_{J,p,I} - x'_{J,q,I}\right),\ if\ x'_{total-p,I} > x'_{total-q,I} \quad (32)$$

$$x''_{J,p,I} = x'_{J,p,I} + R_1\left(x'_{J,q,I} - x'_{J,p,I}\right),\ ifx'_{total-q,I} > x'_{total-p,I} \quad (33)$$

Adopt $x''_{J,p,I}$ it improves the objective function value.

Step 6: Update leaders and solution

Apply the selected best fitness values to update the positions or solutions in the population using Equation 34.

$$X_I = X_I + R \cdot \left(f_{Best,I} - X_I\right) \tag{34}$$

X_I, is the current solution of the *Ith* individual. '*R*' is a random number typically in the range [0, 1]. This formulation allows the incorporation of the best solutions from multiple optimization techniques, improving overall performance by leveraging the strengths of each algorithm, and dynamically updating the WHO stallions based on the relative success of their subgroups.

Step 7: Local and Global exploitation

In the MOA phase, mothers inspire their children to enhance their capabilities, leading to better local search and exploitation abilities by implementing minor adjustments to the positions of population members. This upbringing phase is simulated by generating a new position for each member, reflecting children's personality development as depicted in Equation 35. Should the new position yield an improved objective function value, the prior position of the member is updated to this new one as outlined in Equation 36.

$$X_{I,J}^{p3} = X_{I,J} + (1, -, 2, \cdot, Rand, (, 0, 1,)) \cdot \frac{UB_J - LB_J}{T} \tag{35}$$

$$x_I = \left\{ \begin{array}{c} X_I^{p3}, f_I^{p3} \leq f_I \\ x_I, else \end{array} \right) \tag{36}$$

In the third phase of MOA, calculate X_I^{p3} for each population member, updating positions based on iteration counter t and random factors.To mimic the mating behaviour seen in horses, where foals are separated before puberty and integrated for breeding, the algorithm utilizes the formula in Eq. 37.

$$x_{g,K}^{P} = crossover\left(x_{g,I}^{Q}, x_{g,J}^{Z}\right), I \neq J \neq K, Q = Z = end$$

$$crossover = mean \tag{37}$$

In the horse optimization algorithm, the position of horse *Pi* n group *K*, denoted as $x_{g,K}^{P}$ is determined by combining the positions of horse *Q* in group *I* and horse *Z* in group *J*. The crossover probability in this algorithm is labelled as *PC*, remains fixed throughout the process.

Step 8:*Termination*

Repeat the Steps from 3 to 7 until a stopping criterion is met, such as maximum number of iterations or convergence is achieved.

Step 9: Output of the best solution

Identify and outcomes of the best solution from the final population. By integrating these strategies, the hybrid algorithm provides a comprehensive approach to tackling various optimization challenges, effectively combining the structured learning of TLBO, the maternal guidance of MOA, and the social dynamics of WHO. This model aims to efficiently explore and exploit the search space, adjusting dynamically to find optimal solutions in diverse problem landscapes.

Deep learning-based detection

In deep learning-based detection, a sophisticated detector might be developed utilizing recurrent neural networks, RCNN, and bidirectional long short-term memory networks. This detector is designed to accurately localize and classify objects by leveraging carefully selected features. To enhance the model's performance and prevent over fitting, hyper-parameter tuning is implemented using a blend of deep learning techniques. This strategic tuning not only helps in fine-tuning the model but also significantly boosts its ability to generalize across different datasets.

- RCNN

  R-CNN architecture is leveraged for object detection in images, particularly for scenarios where an image contains multiple objects, which traditional CNNs struggle with due to their design for single-object classification. R-CNN operates through a two-step process: Initially, it employs selective search to identify potentially diseased regions within a plant image. Each of these proposed regions, amounting to about 2000 per image, is then processed individually by separate CNN models to extract features. This network architecture features five convolutional layers, with the count of filter response maps escalating from 96, to 256, twice 384, and finally back down to 256.

  In this architecture, every convolutional layer is succeeded by a rectified linear unit (ReLU) to introduce nonlinearity and a spatial max pooling operation to decrease the spatial dimensions of the feature maps. Designed to accommodate images of different sizes, the network employs a Region of Interest (RoI) pooling layer to convert the feature maps from each proposed region into uniform, fixed-size vectors before they proceed to the fully connected layers. This standardization is crucial for ensuring consistent inputs for the subsequent stages, which include softmax classification and bounding-box regression. These layers are initialized with weights from a zero-mean Gaussian distribution, with standard deviations of 0.01 for classification and 0.001 for regression, and biases initialized at zero. Individual learning rates are set at 1 for weights and 2 for biases, alongside a global

learning rate of 0.001 to optimize the training process efficiently. To combat overfitting, dropout techniques are applied to the fully connected layers. The network culminates in two parallel output layers designed for class prediction and bounding box adjustments as in Fig. 5.

- RNN

  Recurrent Neural Networks revolutionize sequence predictions by leveraging a unique architecture where outputs from prior steps are inputs to subsequent steps. Unlike traditional neural networks, where inputs and outputs function independently, RNNs rely on historical context, essential for tasks like predicting the next word in a sentence. This contextual awareness is managed through a hidden layer that maintains a 'memory state,' essentially storing relevant past input information. This memory enables RNNs to process sequences effectively, using shared parameters across all inputs and hidden layers. This uniform application of parameters across the sequence simplifies the network's complexity and enhances its ability to handle tasks where historical data influences current output.

- Bi-LSTM

  Bidirectional LSTM incorporates two LSTM layers in one, linking both to the output layer to improve learning long-term dependencies and, consequently, boost model performance. Research has demonstrated that bidirectional networks outperform standard models in various applications, notably in estimating blood pressure. BiLSTM utilizes a forward and a backward LSTM layer, depicted in an illustrative layout, which processes inputs in both chronological and reverse order. This dual approach allows for richer contextual understanding and more accurate predictions. The outputs of both LSTM layers are merged using a sigmoid function into a single output vector, reflecting the estimated value for the next time step. Each layer in the BiLSTM is stacked, increasing network depth which has been shown to effectively address long-term dependency challenges in sequential data. Moreover, introducing residual connections, which add outputs from earlier layers to those of deeper layers, can further enhance performance by facilitating deeper network training without the degradation problem often seen with increased depth.

- Fully connected layer

  The architecture involves passing the combined feature vector from earlier layers through a sequence of three fully connected layers. The ReLU activation function is used in the first two layers to add non-linearity, and the final layer uses SoftMax activation to create probability distributions for various text sentiment categories. To combat over fitting, the model incorporates L2-norm regularization with a coefficient of 0.1 and dropout with a rate of 0.30 in the fully connected layers. These techniques help ensure that the model generalizes well to new, unseen data by penalizing complex models and randomly deactivating certain neurons during training.

- SoftMax

  SoftMax is a crucial function in CNNs that converts the raw output scores from the network—also known as logits—into a probability distribution across potential class labels. This is achieved by first normalizing the logits so their total sum becomes 1. Each element in the vector is exponentiation and then divided by the total sum of these exponentiation values, ensuring each element falls between 0 and 1. These elements now represent the probabilities that the input image belongs to each class.

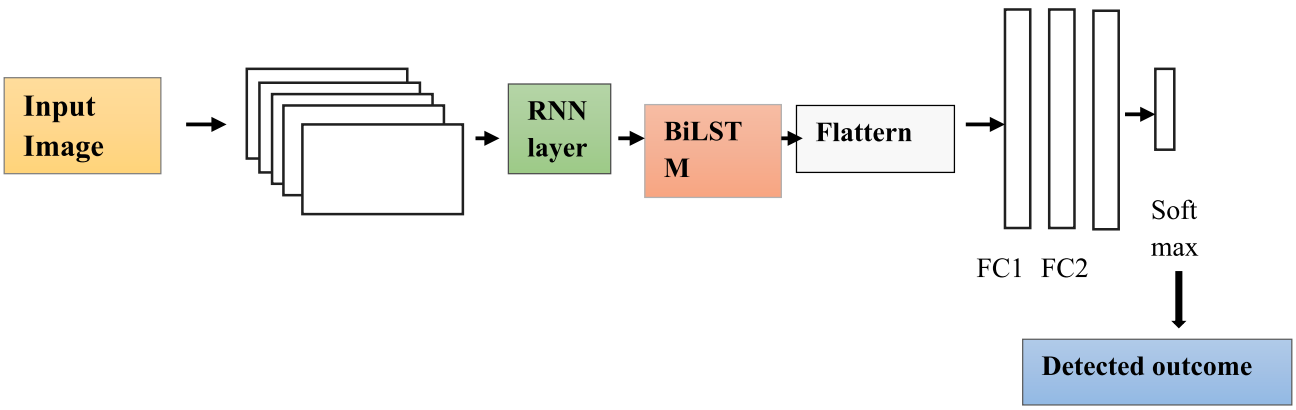


Fig. 5 Deep learning-based detection model

By providing a probability for each class, the SoftMax function enables the CNN to make more precise and confident predictions about the category of the input image. Moreover, the following figures from Fig. 6a–e denotes the process carried out by the suggested approach for detecting the diseased regions from aerial images.

Results and discussion

Experimental setup

This section presents the findings and discussions derived from the conducted research. This study utilizes PYTHON to execute the proposed model, employing performance indicators such as NPV, MCC, FNR, FPR, precision, F-measure, sensitivity, specificity, and accuracy. A hybrid optimization model enhances accuracy. It also explores accuracy loss and density variations at learning rates of 0.7 and 0.8. The proposed method is a combination of feature selection with hybrid optimization of MOA + IWHO + TLBO and CNN classifier. GD is used for tuning the hyper parameters like learning rate, batch size and momentum.

Discussions

Table 2 delineates the comparative performance metrics for various models, each set at a learning rate of 0.7, momentum of 0.81 and a batch size of nearly 75% of the total data set. The RNN model records an accuracy of 97.4048%, with precision at 81.5789%, sensitivity at 98.4127%, and specificity at 97.2816%. Additionally, it scores an F-measure of 89.2086%, an MCC of 88.2472%, an NPV of 98.8801%, a FPR of 2.7184%, and a false negative rate of 1.5873%. The DNN model exhibits an accuracy of 95.8478%, precision of 73.4940%, sensitivity of 96.8254%, specificity of 95.7282%, F-measure of 83.5616%, MCC of 82.2480%, NPV of 94.5960%, FPR of 4.2718%, and FNR of 3.1746%. The CNN model displays superior performance with an accuracy of 97.9239%, precision of 84.9315%, sensitivity of 98.4127%, specificity of 97.8641%, F-measure of 91.1765%, MCC of 90.3209%, NPV of 99.8020%, FPR of 2.1359%, and FNR of 1.5873%. The Bi-LSTM model shows an accuracy of 97.5779%, precision of 85.5072%, sensitivity of 93.6508%, specificity of 98.0583%, F-measure of 89.3939%, MCC of 88.1460%, NPV of 99.2141%, FPR of 1.9417%, and FNR of 6.3492%. The proposed model surpasses all with an accuracy of 99.1349%, precision of 95.3125%, sensitivity of 98.8254%, specificity of 99.4175%, F-measure of 99.0630%, MCC of 95.5809%, NPV of 99.8925%, FPR of 0.5825%, and FNR of 1.1746%.

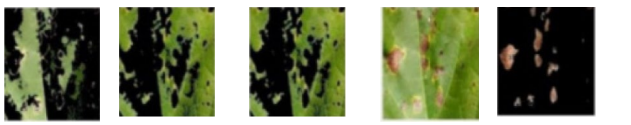
Table 2 Performance measures of proposed model with learning rate 0.7, momentum of 0.8 and 75% of the total dataset for training

Metrics	RNN	DNN	CNN	Bi-LSTM	PROPOSED
Accuracy	0.974048	0.958478	0.979239	0.975779	0.991349
Precision	0.815789	0.73494	0.849315	0.855072	0.953125
Sensitivity	0.984127	0.968254	0.984127	0.936508	0.988254
Specificity	0.972816	0.957282	0.978641	0.980583	0.994175
F-Measure	0.892086	0.835616	0.911765	0.893939	0.99063
MCC	0.882472	0.82248	0.903209	0.88146	0.955809
NPV	0.988801	0.94596	0.99802	0.992141	0.998925
FPR	0.027184	0.042718	0.021359	0.019417	0.005825
FNR	0.015873	0.031746	0.015873	0.063492	0.011746

In Table 3, the assessment of various models at a learning rate of 0.8, momentum of 0.87 and batch size of nearly 75% of the total data set is displayed. The RNN model achieves an accuracy of 97.9221%, with a precision of 85.7143%, sensitivity of 98.8464%, specificity of 97.6261%, an F-measure of 92.3077%, MCC of 91.4765%, NPV of 98.6547%, FPR of 2.3739%, and FNR of 1.1326%. The DNN model registers an accuracy of 96.8831%, precision of 80%, sensitivity of 95.2121%, specificity of 96.4392%, F-measure of 88.2186%, MCC of 87.8358%, NPV of 95.5547%, FPR of 3.5608%, and FNR of 2.9656%. The CNN model reports an accuracy of 98.4416%, precision of 88.8889%, sensitivity of 98.6155%, specificity of

Table 3 Performance measures of proposed model with learning rate 0.8, momentum of 0.87 and 75% of the total dataset for training

Metrics	RNN	DNN	CNN	Bi-LSTM	Proposed
Accuracy	0.979221	0.968831	0.984416	0.976623	0.997403
Precision	0.857143	0.8	0.888889	0.862105	0.979592
Sensitivity	0.988464	0.952121	0.986155	0.943655	0.995215
Specificity	0.976261	0.964392	0.982196	0.983294	0.997033
F-Measure	0.923077	0.882186	0.941176	0.914286	0.989691
MCC	0.914765	0.878358	0.934378	0.905326	0.988274
NPV	0.986547	0.955547	0.971325	0.99427	0.997655
FPR	0.023739	0.035608	0.017804	0.026706	0.002967
FNR	0.011326	0.029656	0.015125	0.051654	0.010357



(a). Denoised image (b). Contrast enhancement (c). Contrast enhancement (d). White balancing (e) Detection of diseases from the Plant Imagery

Fig. 6 Detecting the diseased regions from aerial images

Table 4 Performance measures for the proposed model using GD hyper parameter tuning, with learning rate 0.7, momentum of 0.81 and 75% of the total dataset for training

Metrics	RNN	DNN	CNN	Bi-LSTM	Proposed
Accuracy	0.978221	0.968731	0.983416	0.966623	0.987403
Precision	0.854143	0.822	0.888779	0.862215	0.987592
Sensitivity	0.988364	0.952221	0.985155	0.942655	0.985215
Specificity	0.975261	0.963392	0.982596	0.983594	0.997133
F-Measure	0.923177	0.882286	0.941276	0.924286	0.989694
MCC	0.914755	0.878558	0.934478	0.905526	0.988374
NPV	0.986647	0.956547	0.972325	0.991427	0.995655
FPR	0.023729	0.035618	0.017814	0.026716	0.002957
FNR	0.011316	0.029626	0.015225	0.052654	0.020357

Table 5 Performance measures of proposed model using GD hyper parameter tuning with learning rate 0.8, momentum of 0.87 and 75% of the total dataset for training

Metrics	RNN	DNN	CNN	Bi-LSTM	Proposed
Accuracy	0.979221	0.968831	0.984416	0.976623	0.997403
Precision	0.857143	0.8	0.888889	0.862105	0.979592
Sensitivity	0.988464	0.952121	0.986155	0.943655	0.995215
Specificity	0.976261	0.964392	0.982196	0.983294	0.997033
F-Measure	0.923077	0.882186	0.941176	0.914286	0.989691
MCC	0.914765	0.878358	0.934378	0.905326	0.988274
NPV	0.986547	0.955547	0.971325	0.99427	0.997655
FPR	0.023739	0.035608	0.017804	0.026706	0.002967
FNR	0.011326	0.029656	0.015125	0.051654	0.010357

98.2196%, F-measure of 94.1176%, MCC of 93.4378%, NPV of 97.1325%, FPR of 1.7804%, and FNR of 1.5125%. The Bi-LSTM model notes an accuracy of 97.6623%, precision of 86.2105%, sensitivity of 94.3655%, specificity of 98.3294%, F-measure of 91.4286%, MCC of 90.5326%, NPV of 99.4270%, FPR of 2.6706%, and FNR of 5.1654%. The proposed model excels with an accuracy of 99.7403%, precision of 97.9592%, sensitivity of 99.5215%, specificity of 99.7033%, F-measure of 98.9691%, MCC of 98.8274%, NPV of 99.7655%, FPR of 0.2967%, and FNR of 1.0357%. Tables 4 and 5 are incorporated to illustrate the performance of Gradient Descent (GD) based hyper parameter tuning.

Figure 7a–i, displays the performance metrics for the proposed model at two different learning rates of 0.7 and 0.8. Excluding FNR and FPR, the model outperforms others in all metrics, attributed to the integration of hybrid optimization and deep learning techniques. Table 4 and 5, illustrates that almost equivalent results are obtained for the performance measures while using varied learning rates and momentum values on 75% of the total data set.

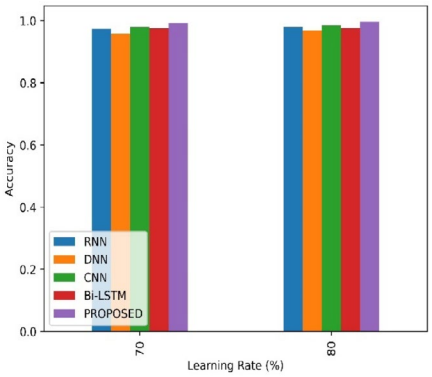
Figure 8 shows how accuracy and loss vary at 0.7 as learning rate. The graph highlights accuracy in blue and validation accuracy (Val accuracy) along with its corresponding loss in orange.

Figure 9 displays the fluctuations in accuracy and loss at a learning rate of 0.8. In the graph, blue represents accuracy while orange indicates validation accuracy (Val accuracy) and its associated loss.

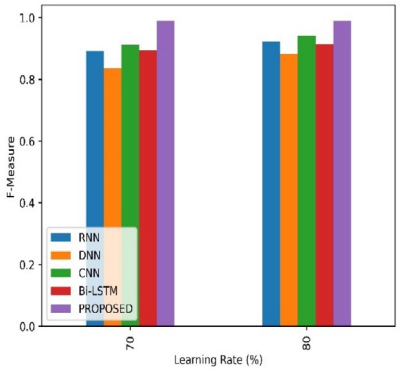
Figure 10 depicts density plots for actual and predicted values at learning rates of 0.7 and 0.8. In both graphs, the actual values are indicated in orange, while the predicted values are shown in purple. Table 6 and 7, depicts the accuracy values for two different hyper parameter models used for training the deep learning models using GD. Optimal accuracy is achieved for both the sets of hyper parameters which are tuned using GD.

Conclusion

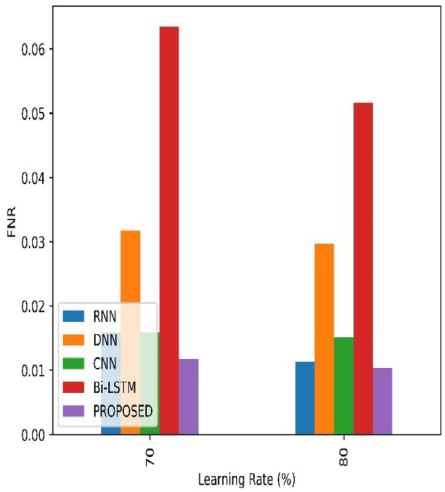
This study presents a cutting-edge hybrid deep learning strategy designed to overcome the challenges presented by traditional monitoring methods, especially in complex and computationally limited scenarios. By incorporating advanced image enhancement techniques like Bilateral Filtering, Gamma Correction, and White Balancing, along with extensive data augmentation, the approach substantially improves image quality and increases the dataset size, enabling more precise feature extraction for accurate plant disease detection. The methodology leverages EfficientDet and Squeeze Net to extract robust features, and introduces innovative descriptors such as the Color Coherence Vector and shape-based metrics to form an extensive feature set. At the heart of this system is a tailored hybrid optimization algorithm that combines the TLBO with an enhanced version of the wild horse optimization algorithm, is deployed for selecting the most pertinent features for plant disease detection. The integration of sophisticated deep learning architectures, including region-based CNNs, RNNs, and Bi-LSTMs, with strategic hyper-parameter optimization, GD for weight adjustment, feature Selection with hybrid optimization of MOA + IWHO + TLBO and CNN classifier ensures not only high accuracy but also effective generalization to new scenarios in categorizing and detecting the plant diseases. The future potential for this technology is significant. Prospective studies could investigate its application to real-time scenarios in dynamic environments like autonomous driving and surveillance, where fast and dependable object detection is essential. Further advancements could include exploring more complex deep learning models and additional optimization strategies to further improve both the precision and the speed of the object detection system, extending the frontiers of current capabilities in the field.



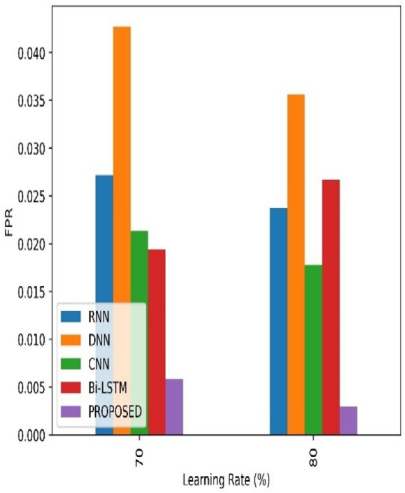
(a) Accuracy



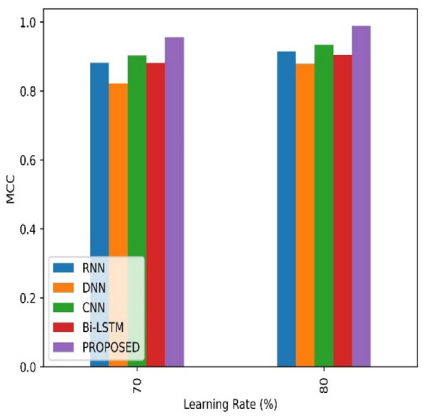
(b) F-measure



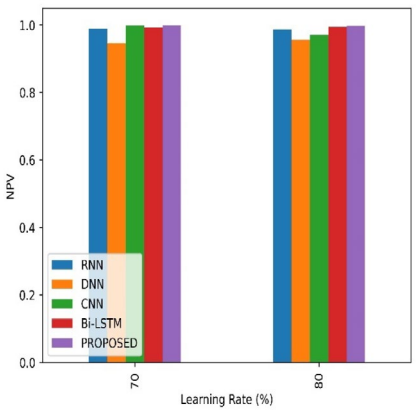
(c) FNR



(d) FPR



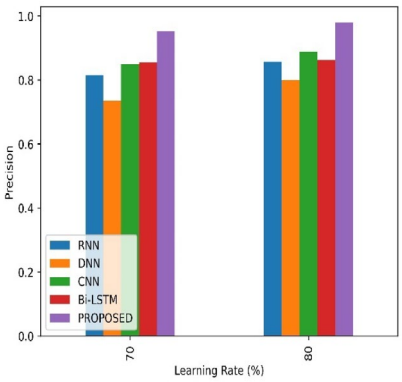
(e) MCC



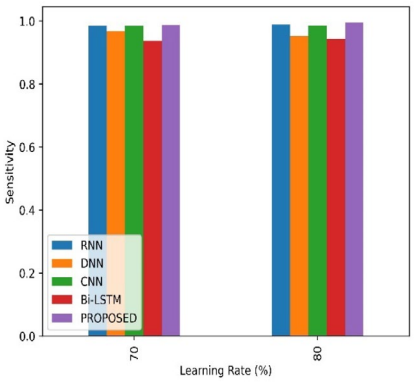
(f) NPV

Fig. 7 **a–i** Metrics Performance of proposed with learning rates 0.7 and 0.8

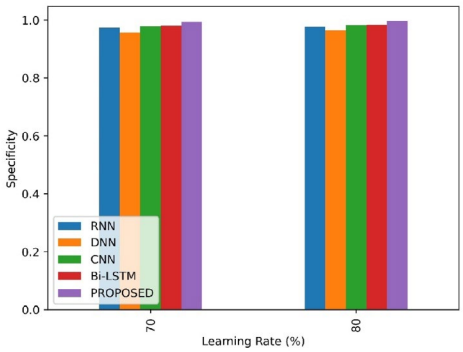
Fig. 7 (continued)



(g) Precision



(h) Sensitivity



(i) Specificity

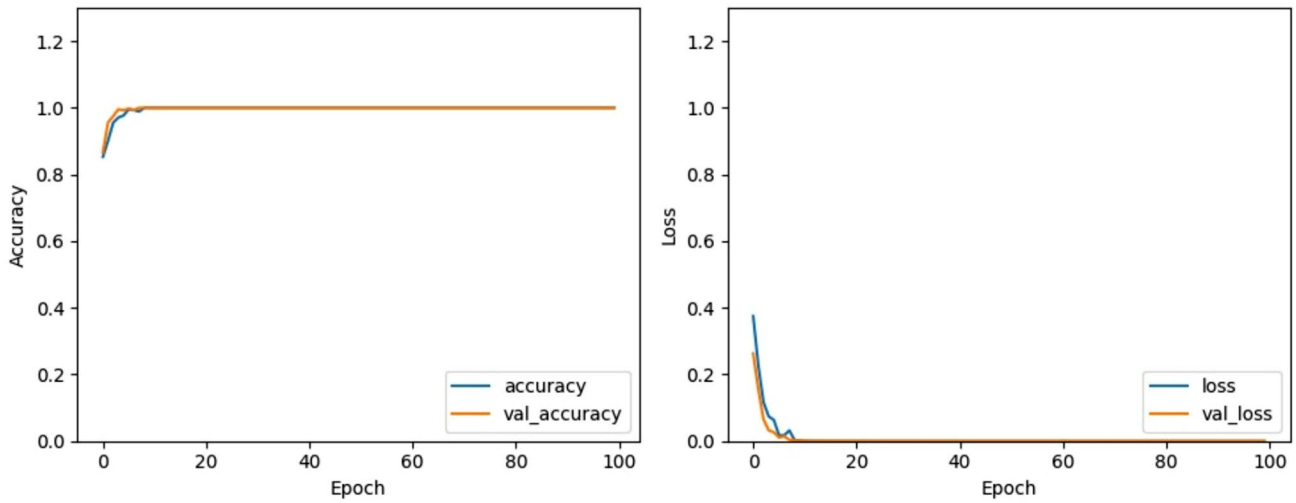


Fig. 8 Accuracy- loss with learning rate 0.7

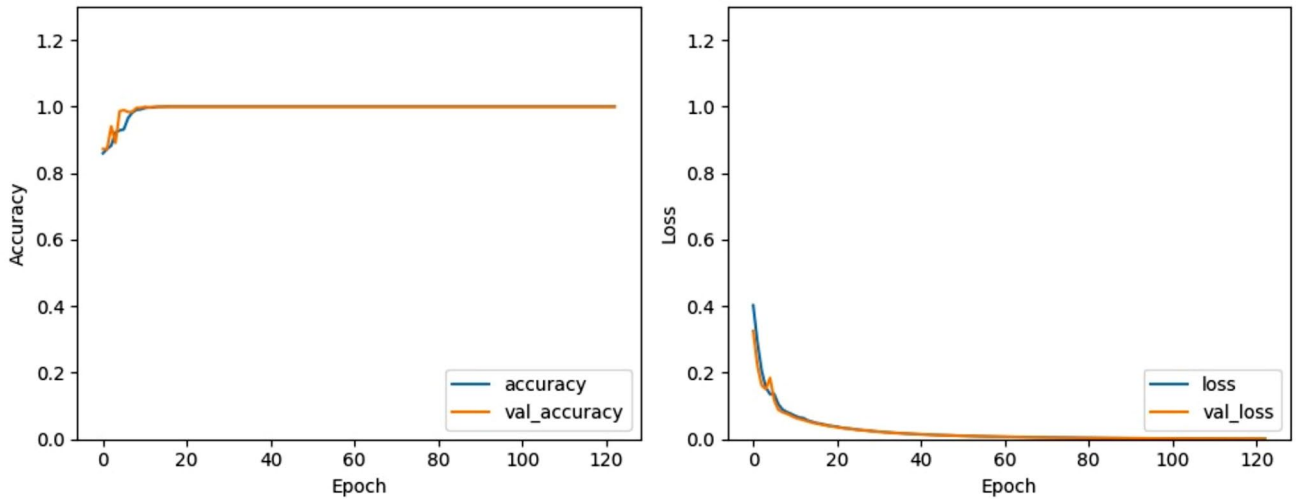


Fig. 9 Accuracy loss with learning rate 0.8

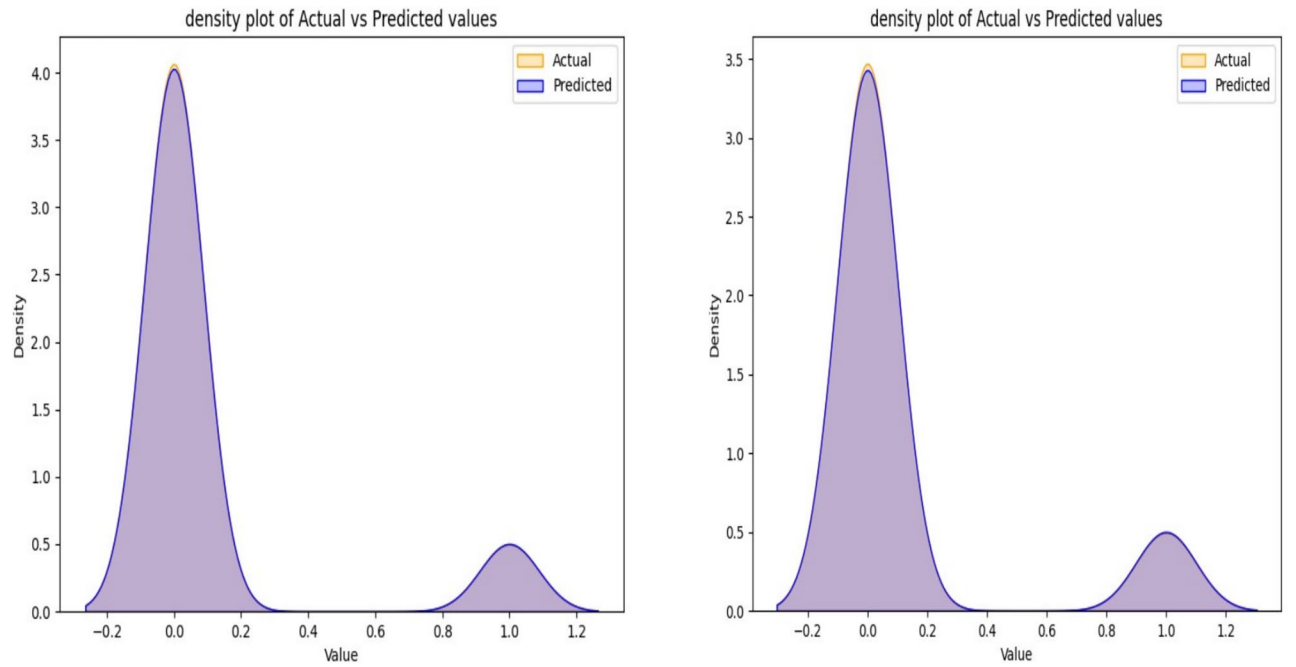


Fig. 10 Density plot with learning rate 0.7 and 0.8

Table 6 Hyper parameters for the various Deep learning models using GD

Hyper parameters	Accuracy				
	RNN	DNN	CNN	Bi-LSTM	Proposed
Learning rate = 0.8	0.978	0.967	0.983	0.975	0.987
Batch Size = 75%	0.856	0.821	0.879	0.872	0.962
Loss = 0.0001	0.964	0.961	0.975	0.944	0.994
Activation function = Softmax	0.972	0.963	0.985	0.985	0.994
Momentum = 0.87	0.933	0.883	0.942	0.924	0.979

Table 7 Hyper parameters for the various Deep learning models using GD

Hyper parameters	Accuracy				
	RNN	DNN	CNN	Bi-LSTM	Proposed
Learning rate = 0.7	0.979	0.968	0.984	0.976	0.997
Batch Size = 75%	0.857	0.8	0.888	0.862	0.979
Loss = 0.0001	0.988	0.952	0.986	0.943	0.995
Activation function = Softmax	0.923	0.882	0.941	0.914	0.989
Momentum = 0.81	0.914	0.878	0.934	0.905	0.988

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability The data is taken from the open source data base, https://www.kaggle.com.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

References

Chalavadi, V., Jeripothula, P., Datla, R., Ch, S.B.: mSODANet: a network for multi-scale object detection in aerial images using hierarchical dilated convolutions. Pattern Recogn. **126**, 108548 (2022)

Deng, Z., Sun, H., Zhou, S., Zhao, J., Zou, H.: Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **10**(8), 3652–3664 (2017)

Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., Zou, H.: Multi-scale object detection in remote sensing imagery with convolutional neural networks. ISPRS J. Photogramm. Remote Sens. **145**, 3–22 (2018)

Ding, J., Xue, N., Xia, G.S., Bai, X., Yang, W., Yang, M.Y., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: Object detection in aerial images: a large-scale benchmark and challenges. IEEE Trans. Pattern Anal. Mach. Intell. **44**(11), 7778–7796 (2021)

Fu, K., Chang, Z., Zhang, Y., Sun, X.: Point-based estimator for arbitrary-oriented object detection in aerial images. IEEE Trans. Geosci. Remote Sens. **59**(5), 4370–4387 (2020)

Ge, L., Wang, G., Zhang, T., Zhuang, Y., Chen, H., Dong, H. and Chen, L.: Adaptive dynamic label assignment for tiny object detection in aerial images. IEEE J. Select. Top. Appl. Earth Obs. Remote Sens. (2024)

Han, J., Ding, J., Li, J., Xia, G.S.: Align deep features for oriented object detection. IEEE Trans. Geosci. Remote Sens. **60**, 1–11 (2021)

Hu, Z., Gao, K., Zhang, X., Wang, J., Wang, H., Han, J.: Probability differential-based class label noise purification for object detection in aerial images. IEEE Geosci. Remote Sens. Lett. **19**, 1–5 (2022)

Koyun, O.C., Keser, R.K., Akkaya, I.B., Töreyin, B.U.: Focus-and-detect: a small object detection framework for aerial images. Signal Process.: Image Commun. **104**, 116675 (2022)

Long, H., Chung, Y., Liu, Z., Bu, S.: Object detection in aerial images using feature fusion deep networks. IEEE Access **7**, 30980–30990 (2019)

Ma, Y., Chai, L., Jin, L.: Scale decoupled pyramid for object detection in aerial images. IEEE Trans. Geosci. Remote Sens. **61**, 1–4 (2023)

Nie, G., Huang, H.: Multi-oriented object detection in aerial images with double horizontal rectangles. IEEE Trans. Pattern Anal. Mach. Intell. **45**(4), 4932–4944 (2022)

Pi, Y., Nath, N.D., Behzadan, A.H.: Convolutional neural networks for object detection in aerial imagery for disaster response and recovery. Adv. Eng. Inform. **43**, 101009 (2020)

Ševo, I., Avramović, A.: Convolutional neural network based automatic object detection on aerial images. IEEE Geosci. Remote Sens. Lett. **13**(5), 740–744 (2016)

Sharma, M., Dhanaraj, M., Karnam, S., Chachlakis, D.G., Ptucha, R., Markopoulos, P.P., Saber, E.: YOLOrs: object detection in multimodal remote sensing imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **14**, 1497–1508 (2020)

Sun, P., Chen, G., Shang, Y.: Adaptive saliency biased loss for object detection in aerial images. IEEE Trans. Geosci. Remote Sens. **58**(10), 7154–7165 (2020)

Wang, J., Yang, W., Li, H.C., Zhang, H., Xia, G.S.: Learning center probability map for detecting objects in aerial images. IEEE Trans. Geosci. Remote Sens. **59**(5), 4307–4323 (2020a)

Wang, Y., Wang, L., Lu, H., He, Y.: Segmentation based rotated bounding boxes prediction and image synthesizing for object detection of high resolution aerial images. Neurocomputing **388**, 202–211 (2020b)

Wang, J., Yu, J., He, Z.: ARFP: a novel adaptive recursive feature pyramid for object detection in aerial images. Appl. Intell. **52**(11), 12844–12859 (2022)

Weng, W., Wei, M., Ren, J. and Shen, F.: Enhancing aerial object detection with selective frequency interaction network. IEEE Trans. Artif. Intell. (2024)

Xu, W., Zhong, S., Yan, L., Wu, F., Zhang, W.: Moving object detection in aerial infrared images with registration accuracy prediction and feature points selection. Infrared Phys. Technol. **92**, 318–326 (2018)

Xu, T., Sun, X., Diao, W., Zhao, L., Fu, K., Wang, H.: ASSD: feature aligned single-shot detection for multiscale objects in aerial imagery. IEEE Trans. Geosci. Remote Sens. **60**, 1–17 (2021)

Xu, C., Wang, J., Yang, W., Yu, H., Yu, L., Xia, G.S.: Detecting tiny objects in aerial images: a normalized Wasserstein distance and a new benchmark. ISPRS J. Photogramm. Remote Sens. **190**, 79–93 (2022)

Zeggada, A., Melgani, F., Bazi, Y.: A deep learning approach to UAV image multilabeling. IEEE Geosci. Remote Sens. Lett. **14**(5), 694–698 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.