

Segmentation of Sunflower Leaf Disease using Improved YOLO Network with IDMO Model

¹T. Thilagavathi, ²Dr. L. Arockiam

Submitted: 24/11/2023

Revised: 30/12/2023

Accepted: 09/01/2024

Abstract: Predicting and recognizing plant diseases at an early stage is a critical requirement for expanding agriculture, which contributes significantly to the economy and food security of our country. Early detection can help to save crops and prevent further damage. Deep learning methods are commonly used for image-based disease classification and prediction. This paper studies sunflower diseases such as *Alternaria* leaf spot and *Verticillium* wilt, and presents a deep learning segmentation model to classify them. The study enhances the YOLOv5 architecture, the scales of the fusion layer, and the multiscale detection layer to make the first-stage prediction more effective at detecting small, subtle defects with high similarity on the leaf surface. In addition, a modified version of the Improved Dwarf Mongoose Optimization Algorithm (IDMO) is used for hyperparameter tuning. This optimization method makes three minor but significant changes to the original algorithm (DMO). First, IDMO's alpha selection is different from DMO's, where calculating the probability value of each fitness member is an unnecessary computational burden that adds nothing to the alpha's or the group's overall quality. The Kaggle dataset images are pre-processed to improve classification accuracy. The experimental results validate that the proposed model outperforms state-of-the-art deep learning approaches by a margin of approximately 95%. This research has the potential to revolutionize the way plant diseases are detected and classified. By using deep learning, farmers can quickly and accurately identify diseases, which will help them to save crops and prevent further damage. This could lead to increased crop yields and improved food security.

Keywords: Improved Dwarf Mongoose Optimization; Sunflower Leaf Disease; Agriculture; Deep learning techniques; YOLO Network; Downy mildew.

1. Introduction

The *Helianthus* sunflower, popularly known as the common sunflower, has its origins in Mexico, North America, around 2100 BCE. Sunflowers, which may reach heights of 3 to 4 meters, are a common crop in tropical regions. The Journal of Environmental Management reports that nitrogen-based fertilizer has varying effects on sunflowers depending on the plant's genetic makeup and background [1]. Because sunflowers and their seeds are useful in a variety of ways, including as sources of nutrients and medications, a drop in output due to pests and illnesses might have far-reaching consequences. For example, the plant's seed and leaves can be eaten because of the nutrients they contain, and the plant's roots can be used to absorb radioactive chemicals or to create natural colours. Vitamin-rich sunflower is also used to cure a wide range of ailments, including tract infections. It's effective against bug, snake, and spider bites, too [2]. Since sunflower leaves are diuretics, they have also been used to treat bladder diseases. Given its widespread practical application, protecting sunflowers through the use of computer vision for early disease detection has become an urgent priority.

Sunflower, a crop with a long history of scientific attention due to its potential as a major industrial crop, is native to North America. Rapid progress in the field of computer vision has led to its widespread use in areas like agricultural engineering and production, giving smart agriculture a growing impact on people's everyday lives [3]. Due to the many uses of sunflower oil, growing sunflowers (*Helianthus annuus*) is an important agricultural activity. The seeds are very nutritious, with a moisture content of 5.50%, a protein gratified of 18.72%, a fat gratified of 37.47%, a fibre gratified of 28.30%, an ash gratified of 3.49%, a carbohydrate content of 6.11% [4]. Researchers have identified 90–100 new sunflower diseases over the world. Most fungal diseases, including rusts, *Phoma* black stem blight, anthracnose, and leaf spot, are rather common. Fungi are responsible for the vast majority of sunflower diseases [5]. Leaf blight, produced by the fungus *Alternaria* species, has been documented in all of the world's sunflower-growing areas [6]. However, it is more common in the tropics and subtropics.

Optical monitoring of plant leaves has long been used by low-resource farmers as a proxy for disease diagnosis [7]. Fertilizers might be used, stressing plants and leading to nutritional deficits in agricultural settings if an inaccurate diagnosis of crop loss is made [8]. The identification for maximizing the farmer's return on investment in plant development and production. Manually checking for plant diseases will not reliably yield correct findings. It's also

¹Research Scholar, Department of Computer Science, St. Joseph's College (Autonomous), Affiliated to Bharathidasan University Tiruchirappalli, Tamil Nadu, India. 620002. thilagamca96@gmail.com

²Associate Professor, Department of Computer Science, St. Joseph's College (Autonomous) Affiliated to Bharathidasan University Tiruchirappalli, Tamil Nadu, India-620002

difficult and expensive for farmers to obtain subject matter expertise in plant disease tracking [9]. Deep learning is now feasible because of developments in computer vision; it has been the subject of many studies into its potential for early disease detection in plants [10]. Plants are a serious issue in farming because of the negative impact they have on crop quality and yield. In underdeveloped countries that rely on the sum of crops, plant diseases can affect the agricultural economy, causing everything from modest symptoms to severe damage to entire planted crop regions, incurring considerable expenditures and having a big negative impact. Several DL models have been utilized to classify plant diseases, with the help of well-known DL architectures. Furthermore, several researchers have tweaked DL algorithms to progress the efficiency of disease categorization across a range of plant species [12]. K-means clustering is only two examples of the many ML methods that have been used for the problem of disease classification and identification in plants. However, due to the difficulty of picture pre-processing and feature extraction, these methods perform less effectively and move more slowly in real-time illness identification. Diseases affecting sunflower plants may be accurately identified and managed by the use of automatic identification of symptoms occurring on leaf tissue [13].

The study suggested an enhanced YOLO model for disease segmentation from sunflower leaf images. In addition, the IDMO model optimizes the hyper-parameter tuning, which will be briefly covered in the next sections. The effectiveness of the suggested model is revealed by the validation study. The secondary sources are obtainable in Section 2, and a concise summary of the study is provided in Section 3. In Section 4, we contemporary the findings of the experimental analysis. Section 5 delivers a summary and looks ahead to potential follow-up.

2. Related Works

In this research, Ghosh, P. et al. [14] combine transfer learning (TL) with a conventional (CNN) to develop a hybrid model for sunflower illness detection. On a dataset with four classes (grey mold, leaf scars, and renewed leaf), CNN hybrid perfect surpasses the other eight models. The experimental findings demonstrate that the recommended model outperforms the competition on the benchmark dataset in terms of precision, and accuracy.

The purpose of the research conducted by Barrio-Conde, M. et al. [15] is to determine how effectively deep learning (DL) systems can be seeds. An image capture system was set up with a Nikon camera and controlled illumination to photograph 6,000 seeds from six distinct types of sunflower seeds. Image datasets were created for training, validating, and testing the system. Classification of varieties (between two and six) was accomplished using a

CNN AlexNet model. The classification model has a 100% accuracy rate for two classes and an 89.5% accuracy rate for the other six classes. These values are reasonable since the categorized kinds are so similar that they are difficult to tell apart. This result illustrates the potential utility of DL algorithms for seeds rich in oleic acid.

In this work, Dawod, R.G., and Dobre, C. [16] discuss the findings of categorizing foliar diseases in sunflowers using a technique that comprises a perfect that mechanically splits the leaf lesions shadowed by the scheme. The lesions were segmented using Faster R-CNN in addition to Mask R-CNN. Diseases were classified according to lesions using the networks ResNet152. Results demonstrate that lesions from illnesses with well-defined lesions, such as *Alternaria* and rust, can be automatically segmented. At least one segmented region is corrupted in more than 90% of the photos. Diseases that cause the entire leaf to become white, such as powdery mildew, make segmentation more difficult to perform. Diseased regions could not be successfully separated from 30% of pictures. Results showed that a system classifies illnesses, allowed for the identification of pictures for which a precise classification could not be performed, and for a higher degree of accuracy in disease classification overall.

Sunflower leaves are susceptible to a variety of diseases, and Malik, A., et al. [17] looked into four of them: The paper also suggests a hybrid perfect for the detection and categorization of various disorders using MobileNet, both transfer learning models used for classification, may be combined using the technique, or a hybrid model can be constructed using the two individual models. The author used Google Images to compile data collection of 329 pictures of sunflowers, which she then sorted into five groups. Using the same data set as the initial contrast, the suggested model is compared to various pre-existing deep learning replicas in terms of accuracy.

The dataset of sunflower flower and leaf images supplied by Sara, U. et al. [18] will help researchers develop accurate disease detection algorithms. Sunflowers with and without damage, such as leaf scars, downy mildew, and grey mold, are included in the dataset. Images were captured by hand at the protest farm of the Bangladesh Institute (BARI) in Gazipur between November 25 and November 29, 2021, in conjunction with one of BARI's subject matter experts, when most diseases were present.

Using a collection of field photos depicting four foliar diseases of sunflowers (*Helianthus*), Dawod RG and Dobre C categorized the illnesses using a ResNet neural network [19]. Next, CNN visualization techniques were used to illustrate the forecast for incorrectly identified photos. In certain cases, the interpretation methods have

highlighted, incorrect categorization of background items. We also saw examples of photos with several leaves where the prediction was based on randomly picked sections of leaves. The late stage of the illness, when the lesions combine, also appears to be a possible misclassification risk. Incorrect classifications can also result from a lack of symptom representation or from images in which the visual indications are the same for different diseases due to a lack of a collection of images diverse enough to represent how a disease manifests itself. By removing many of the confounding variables that might contribute to inaccurate classification, segmented lesions were shown to increase classification accuracy in this study's visualizations.

In this work, Song, Z et al. [20] use data from the study region acquired in 2018 to construct their model, and then use data from 2019 to assess the model's efficacy. Through testing against other methods, we find that PSPNet strikes a fair mix between precision and efficiency. To further enhance model performance, this work proposes a more appropriate loss function to weigh different kinds of misclassification, one that takes into consideration the time-series connection between the categorization of neighbouring growth phases. According to the results, the improved PSPNet using the proposed weighted loss function yields the highest identification accuracy (89.01%), providing a solution for the recognition of the sunflower growth period using single-phase data.

3. Proposed Methodology for Segmentation

3.1. The YOLOv5 Model

YOLO is a regression-based technique for locating targets. By feeding photos or video into a deep network and calculating the loss function to anticipate the objects' categorization and placement, YOLO converts the target detection issue [21].

Based on the original YOLO detection construction, YOLOv5 [22] makes use of the best algorithm optimization method to emerge in recent years, with auto backbone, neck, and output as its four pillars. Input terminals often house pre-processing operations like mosaic data enhancement and adaptive picture filling. YOLOv5 is adaptable to datasets including photos of sunflower leaves because of its smart initial anchor frame sizing. The primary structure of the backbone network is a (CSP), which is used to extract feature maps of varied sizes from images. [23] and spatial pyramid pooling (SPP) [24]. SPP structure realizes the feature extraction from

several scales can build three-scale feature maps, which helps enhance the detection accuracy; bottleneck CSP is used to decrease the amount of calculation and raise the speed of inference. The neck network makes use of the FPN and PAN feature pyramid architectures. Strong is transmitted from the top feature maps to the bottom feature maps via the FPN [25] construction. While doing so, the PAN [26] architecture passes on robust localization features from lower to maps. Together, the features gathered from various levels of the network are strengthened in Backbone fusion, enhancing the detection capacity even more. The data from the head is mostly utilized to make size predictions for targets on feature maps during the final stage of detection. Four distinct designs make up YOLOv5; they are the YOLOv5s and YOLOv5x. In particular, the amount of feature extraction components kernels at various nodes of the network distinguishes one from the other.

Defects on the sunflower photographs tend to be numerous, tiny, and of varying sizes, and they tend to be scattered randomly around the picture. The original YOLOv5 model is inadequate under these conditions, and problems like low finding accuracy and misused detection occur. Therefore, the purpose of this article is to enhance the current YOLOv5 network model. To improve the extraction capacity of the perfect and the precision of illness diagnosis, we first tweaked the backbone network, and incorporated an attention-finding layer based on the unique characteristics of small disease targets in leaves, allowing the detection model to more effectively adapt to the small disease.

3.2. Improved Backbone

Focus slice, Bottleneck CSP, and SPP modules collect feature maps of varying sizes from the input picture to form YOLOv5's backbone network [27]. The model's overall detection performance is proportional to the strength of the backbone network's feature extraction.

The overall shape of a deep convolutional neural network resembles an inverted pyramid, and the size of the feature map is proportional to the depth of the network. Therefore, due to the pooling and down sampling procedure, the feature information of tiny objects in the picture may be absorbed by the feature information around the region, leading to their disappearance. Image flaws are typically negligible when compared to the complete surface. Large-scale down sampling can result in the loss of the data's semantic content, which could result in undetected events.

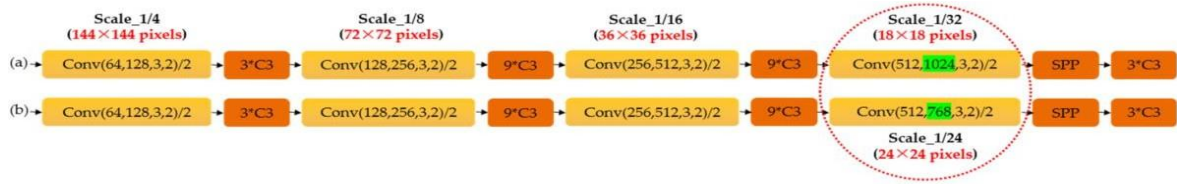


Fig 1. The backbone system before (a) and after (b) modification.

By replacing the original YOLOv5's Conv and C3 layer, which extracted features at a 1/32 scale (refers to the size of feature maps generated is 18 18 pixels), Scale 1 (green background, red circle) was formed after the network, improving the feature extraction capability of the disease defect backbone network. Since already know that YOLOv5 produced large, medium, and small-scale feature maps, the simply scaled down the size of the large-scale feature map level, which enhanced detection accuracy by decreasing the meddling of large-scale worthless information. Figure 1 displays the original and updated backbone networks.

3.3. The Attention Mechanism

The morphological characteristics recovered by the backbone network from multi-scale feature maps comprised both foreground and background information. The foreground info in the feature map is less dense for tiny targets. We included an efficient channel and gave it the monitor ECA-C3 to aid YOLOv5 in filtering out irrelevant information and concentrating on relevant target items. Then, swapped out the C3 module in the baseline YOLOv5 infrastructure with the ECA-C3 module. The efficient dimensionality-preserving method of cross-channel interaction at the local level. Simultaneously, it may adaptively pick the size. Information regarding interactions between channels is captured efficiently by the ECA module, leading to a notable speed boost.

ECA-Net takes into account each channel and its K neighbours to collect local cross-channel interaction information, allowing the neural network to channel in an adaptable manner. How many of the channel's neighbours are factored into the attention computation is represented by the convolutional kernel size K, which in turn indicates the coverage of local cross-channel interactions. Since K is proportional to C, the kernel size K may be computed in an adaptable fashion using the same equation:

$$K = \Psi(C) = \left\lfloor \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma_{odd}} \right\rfloor \quad (1)$$

where C is the channel dimension, $\lfloor t \rfloor_{odd}$ signifies the nearest odd sum of t; γ is set to 2 and b to 1.

The generated convolution kernel undergoes a convolution operation, and the weights for each channel are computed using a function. Here is the equation:

$$\omega_C = \sigma(C1D_k(y)) \quad (2)$$

where $C1D_k$ represents a convolution process with a kernel of K, s is the function, whose calculation formula is $\sigma = \frac{1}{1+e^{-z}}$, y signifies the diverse channels, ω_C is the weight of each channel created, and its measurement is $1 \times 1 \times C$.

Finally, the extracted features are made more directed by weighting and summing the created attention weights and the input feature maps. Below is the weighing formula:

$$X_c = X_c \otimes \omega_C \quad (3)$$

where $\otimes$ signifies increasing element by component, and X_c is the output result after transitory the ECA unit.

3.4. Increasing Scale of Model Prediction

When it comes to diagnosing diseases, it can be tricky since the problem region is so little in comparison to the whole leaf surface that its characteristics are obscured unless the photos are blown up. The following enhancements to the architecture of the detection layer were implemented to address this issue.

As was discussed before, shallow features are more suited for detecting small-scale objects, whereas deep features are better suited for detecting large-scale targets. The feature network (FPN) and pixel aggregation network (PAN) designs are utilized in the feature fusion layer and detection layer network to improve the transferability of positioning. Finally, the three output detection layers are scaled at 1/8, 1/16, and 1/32 for the detection of small, medium, and large targets, respectively. These scales correspond to the three sizes of new feature maps produced by the three feature fusion layers: 72 72 255, 36 36 25, and 18 18 255.

To better capture shallow feature info on the leaf surface, this research employed a 1/4-scale detection layer, denoted by a green backdrop, to create 144 144 feature maps. The yellow backdrop indicates a Scale 4 detecting head. The new fusion layer begins reinforcing the second layer of the backbone network's features, fixing a problem with the previous YOLOv5 model: it was unable to fully use the shallow features. This enhancement allows the deeper network to take advantage of the properties of the shallower network. To enhance tiny item recognition, a prediction head was added to the network's tail. This head

can now provide low-level and high-resolution feature maps.

3.5. Hyper-parameter tuning using the IDMO Model

The natural phenomenon of the dwarf mongoose inspired the development of the DMO [29] algorithm. The alpha (male and female), scout, and babysitter subgroups are all accounted for in the author's model of the dwarf mongoose population. In addition, the animal's ability to respond to changes in its environment, such as predation and food availability, was modeled and included in the design. Then, the assigned babysitters can stay in the mound while the scouting party goes out to find food. It's commonly considered that the process of foraging or searching for food illustrates the optimization process's exploration phase [30]. In addition, the DMO considers the time when a group of dwarf mongooses establishes themselves in a mound after discovering a new food supply to be the exploitation or intensification stage. Additionally, the system modelled the emergence of additional mounds as a consequence of the search. The diagram also demonstrated the group's preparedness for trading babysitters.

Dwarf mongooses as a whole may be represented using Eq. (4), which describes the behaviour of a group of these animals. Dwarf mongoose populations typically include an alpha group, a juvenile group, and a scout, so we isolate the alpha group and then divide the rest of the population up accordingly. Meanwhile, Eq. (5) was used to determine this subset of alpha females (a) in light of their significance to the population. To determine who fits the alpha female's ideal profile, it is necessary to first assess the group's and individuals' fitness levels (a).

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d-1} & x_{1,d} \\ x_{2,1} & x_{2,2} & \ddots & \ddots & x_{2,d-1} & x_{2,d} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d-1} & x_{n,d} \end{bmatrix} \quad (4)$$

This section's application technique for mathematical models is reflected in the table below.:

- | |
|---|
| <p>a. Initialize all control parameters</p> <p>b. The population of the dwarf mongoose is first generated</p> <p>c. Subgrouping the entire population into alpha (male and female), scouts, and babysitters</p> <p>d. Determine the available number of search agents by subtracting the babysitters from the entire population</p> <p>e. Set the rate of exchange for babysitting tasks as L</p> <p>f. While the termination condition is not satisfied, do the following:</p> <p>i. Compute the fitness of the mongoose population/ group</p> <p>ii. Activate and set the time counter</p> <p>iii. Apply Eq. (5) to deduce the size of the alpha female</p> |
|---|

$$a = \frac{fit_i}{\sum_{i=1}^n fit_i} \quad (5)$$

Dwarf mongoose populations are analysed to determine and isolate scout groups, which stand in for the population's labour force. To accomplish this, they derived the corresponding equation (Eq. (6)), which accounts for both foraging and the search for new sleeping mounds.

$$X_{i+1} = \begin{cases} X_i - CF * phi * rand * [X_i - M^{\rightarrow}] & \text{if } \varphi_{i+1} > \varphi_i \\ X_i + CF * phi * rand * [X_i - M^{\rightarrow}] & \text{else} \end{cases} \quad (6)$$

where rand is a random sum among [0, 1] and $CF = \frac{(2 - \frac{iter}{Maxiter})}{(2 - \frac{iter}{Maxiter})}$ is the collective-volatile drive control limit and $M^{\rightarrow} = \sum_{i=1}^{X_i \times sm_i} \frac{X_i \times sm_i}{X_i}$ controls the drive of the mongoose to the novel sleeping mound, and $\varphi = \frac{\sum_{i=1}^n sm_i}{n}$

Dwarf mongooses on a mound must constantly relocate to engage in predatory and foraging behaviors. In many cases, this will need the use of some kind of positioning update system. To determine a person's present location, they use a combination of a randomly generated variable (phi) in the range (1,1) and the peep female (a). The position update model is defined by the following equation:

$$X_{i+1} = X_i + phi * peep \quad (7)$$

The finding of a new sleeping mound is a significant event that is characterized by predatory and foraging activity. During the iterative optimization process, this new hill is expected to evolve. The sleeping mound (sm) is calculated using Eq. (7) as a model for its finding, and its values may be averaged using Eq. (8).

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}, fit_i|\}} \quad (8)$$

v. Update the position of a potential food source using Eq. (7)
v. Iterate over each individual and compute the fitness of X_i
. Derive the sleeping mound for the population using Eq. (8)
. Determine the movement vector $M \rightarrow$
i. Exchange the babysitters
x. Compute the scout group using Eq. (6)
x. Update the solution so far
g. Return the best solution

While the DMO does show some degree of originality, it is possible to improve the algorithm by taking advantage of insights into optimization gleaned from the dwarf mongoose's naturally occurring and cooperative social structure. Improving efficiency and striking a good balance between the discovery and exploitation phases need the consideration of such occurrences. The next part provides a comprehensive breakdown of the enhanced dwarf mongoose optimization algorithm's design and analysis.

3.5.1. The Improved Dwarf Mongoose Optimization Algorithm (IDMO) Perfect

It is proposed that the IDMO will improve DMO exploitation and exploration. DMO's inability to deliver the ideal solution for problems F9, F15, and F17 is evidenced by the solutions it did return. This optimization method makes three very minor but significant adjustments to the original algorithm (DMO). To begin, IDMO's alpha selection is not like DMO's, where calculating the probability value of each fitness member is only an unnecessary computing burden that adds nothing to the alpha's or the group's overall quality. To improve the IDMO's capacity for exploration and exploitation, we pick the healthiest dwarf mongoose to serve as the alpha and incorporate a novel operator to regulate the alpha's behavior. Second, to shake up the search and check out new places, the scout groups' routes are changed randomly. Finally, the babysitter exchange criterion is altered so that, instead of initializing the babysitters anew as is done in DMO.

In contrast to DMO, where scouting and foraging are the same behaviour, this model demonstrates how the two are kept distinct. The search agents, represented by the and matrix in Eq. 9, are the individual dwarf mongooses. Through the phases depicted in Eq. 12, the modified alpha (Eq. 11) guides the team into unfamiliar territory during the exploration phase. The IDMO's capacity for exploration and exploitation is improved by modelling a new operator, in Equation 10. This operator regulates the alpha motion. As demonstrated in Eq. 14, randomization is used to alter the scout group's movements, bringing variety to the search process and allowing them to

investigate previously uncharted territory. Once the criterion for exchanging babysitters has been satisfied, exploitation can occur, as indicated in Eq. 15. At this stage, the solution is further optimized.

Population Initialization

As indicated in Eq. (9), the IDMO population is first seeded with a random matrix of potential dwarf mongooses. The population vector lies among the optimization problem's bounds.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & \dots & x_{1,d-1} & x_{1,d} \\ x_{2,1} & x_{2,2} & \ddots & \ddots & x_{2,d-1} & x_{2,d} \\ \vdots & \vdots & \ddots & x_{i,j} & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \dots & \dots & x_{n,d-1} & x_{n,d} \end{bmatrix} \quad (9)$$

where n is the sum of dwarf mongoose in a mound, $x_{i,j}$ denotes the site of the j th dimension of the i th populace and each $x_{i,j}$ is distinct in Eq. (10).

$$x_{i,j} = rand \times (U - L) + L \quad (10)$$

Alpha Group

To estimate the size of this community, we take the mongooses and divide it by the estimated number of babysitters. The most physically capable female dwarf mongoose is chosen to be the group's alpha (a) leader, as shown in Eq. 11. IDMO's alpha selection is distinct from DMO's, in which calculating the probability fitness is only an unnecessary computing burden that adds nothing to the overall quality of the alpha or any other members.

$$a = \min(fit_1, fit_2, \dots, fit_n) \quad (11)$$

The group is held together by the alpha female's peep-like vocalizations.

As specified by Eq. 12, the IDMO does a random walk over the issue space. Once the healthiest dwarf mongoose in the family, it now leads its relatives to a possible meal. This scenario differs significantly from the DMO, in which the alpha uses just vocalization to manipulate the other dwarf mongoose's posture. In IDMO, the alpha's position determines the additional mongoose's location, and a new worker, defined in Eq. 13, governs the alpha's movement, improving the IDMO's capacity for exploration and exploitation.

$$X_{i+1} = a + phi * rand * (X_i - X_k) \quad (12)$$

$$\omega = e^{-4*(C_{iter}/Max_{iter})^2} \quad (13)$$

where $phi = (\frac{peep}{2}) * rand * \omega$, X_i is the preceding dwarf location, and the rand is a consistently distributed random sum $[-1,1]$. X_k is an arbitrarily designated dwarf mongoose.

Scout Group

Since dwarf mongooses are semi-nomadic and never return to the same sleeping mound twice, it is up to the scouts to find a new one. The IDMO represents the scout group as it searches for a new sleeping mound following a day of foraging. Since dwarf mongooses like to congregate near plentiful food sources, the healthiest scout is often chosen as the site of the chosen sleeping mound. Modelling the scouts using Eq. 14.

$$X_{i+1} = a + phi * rand * (X_k - X_h)/2 \quad (14)$$

where rand is a random sum among $[0, 1]$, and X_k , X_h are arbitrarily selected dwarf mongooses.

The Babysitters

Eq. 15 provides the exchange criterion that is used by the babysitter. Instead of starting again with a new set of dwarf mongooses as is done in zero when the criterion has been satisfied, allowing the traded babysitters to learn about food sources and the location of the next sleeping mound, perhaps resulting in better-suited mongooses. Eq. 16 models this enhancement by showing how the dwarf mongooses is chosen at random and their details are communicated to the babysitters. If L goes to zero, the iteration number and CF are multiplied together to start again.

$$L = \begin{cases} Roundup(0.6 * n * dim * (\frac{1}{C_{iter}})) \\ L * C_{iter} * CF \text{ when } l < 0 \end{cases} \quad (15)$$

$$X_{i+1} = (X_j + rand * (a - \frac{X_k + X_h}{2}) * b) \quad (16)$$

$CF = (1 - \frac{iter}{Max_{iter}})^{(2 - \frac{iter}{Max_{iter}})}$ controls the shared volitive movement of the dwarf mongooses, X_j , X_k , and X_h are randomly designated to replace the babysitters, and br is the birthrate.

The IDMO greatly reduces computational complexity by minimizing the burden of alpha selection. When the dwarf mongooses, led by the dominant female, head out to find food, that's when the optimization process begins. Some people are left behind to act as babysitters.

Finding plenty of stuff to eat is like being in the IDMO's exploring phase. Since dwarf mongooses are not known to deliver food for others, the babysitters switch off about lunchtime. When this occurs, the new set of babysitters will likely head back to their old haunts in search of food before discovering any new hiding places or sleeping mounds. This hypothetical situation represents the IDMO at its exploitation stage. The end-of-day scouting for a sleeping mound is an exploration and exploitation of the search space. Similar to the DMO, the IDMO algorithm simply has a single tuning knob (the total number of nannies).

Conceptual Advantage of the IDMO

The theoretical advantage of the suggested IDMO may be traced back to the inherent randomness of the IDMO's operations. The process begins with creating a new population and moving on to upgrading the alpha and scout groups. These solutions benefit from improved exploration and exploitation of the random Mongooses interchange them. The IDMO may be adjusted in only one way, and its implementation is straightforward and versatile.

4. Results and Discussion

As sunflowers have numerous benefits the study proposed an improved YOLO model with the help of the IDMO model.

Figure 2 presents the sample images of the data that are collected from the Kaggle dataset, (<https://www.kaggle.com/code/noamaanabdulazeem/sunflower-disease-recognition/ notebook>).

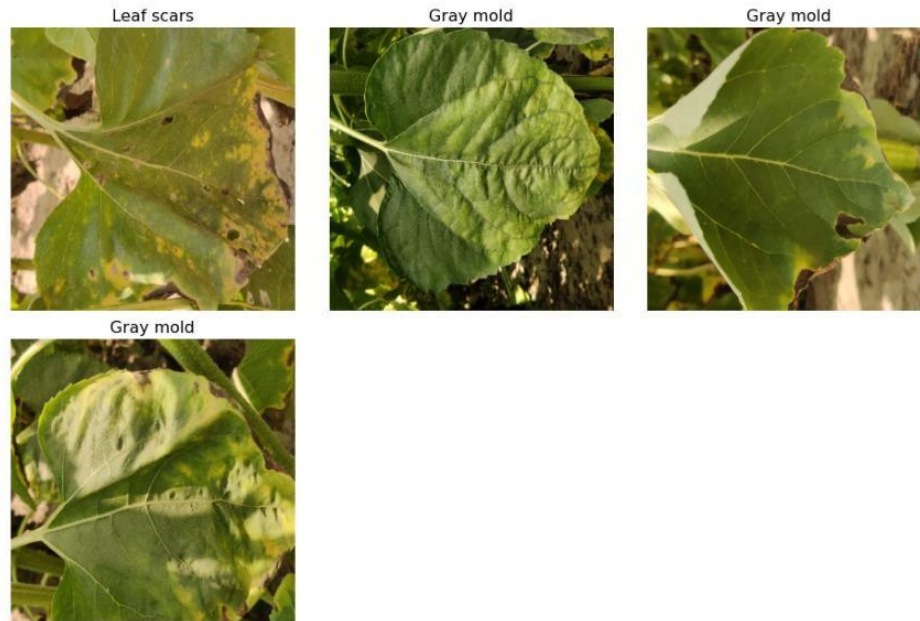


Fig 2: Sample images from the Kaggle Dataset

4.1. Pre-processing of the data

In the first stage, "Data pre-processing," the photos of size $224 \times 224 \times 3$ are converted into an array, the LabelBinarizer function is used to create a list of labels of images, and finally, the ImageDataGenerator is used to supplement the images. The dataset is split into a training set and a testing set using `train_test_split()`, with the training set comprising 80% of the data and the testing set 20%, with the `random_split()` selecting every 3 photos for the splitting. After the dataset has been split up, model training (an enhanced YOLO model) may begin using

`model.fit_generator()` and 25 epochs. Using the NumPy library and the model, compile the predicted values from both models into a single array once training is complete. The Keras predicted function. Now, using the resulting stacked dataset as a training set, create a model and then save the predicted values of the stacked perfect with the stacked dataset for later use in determining the model accuracy on a testing set.

4.2. Validation Analysis of Proposed Model

Figure 3 presents the data distribution and the confusion matrix is presented in Figure 4.



Fig 3: Distribution of the data

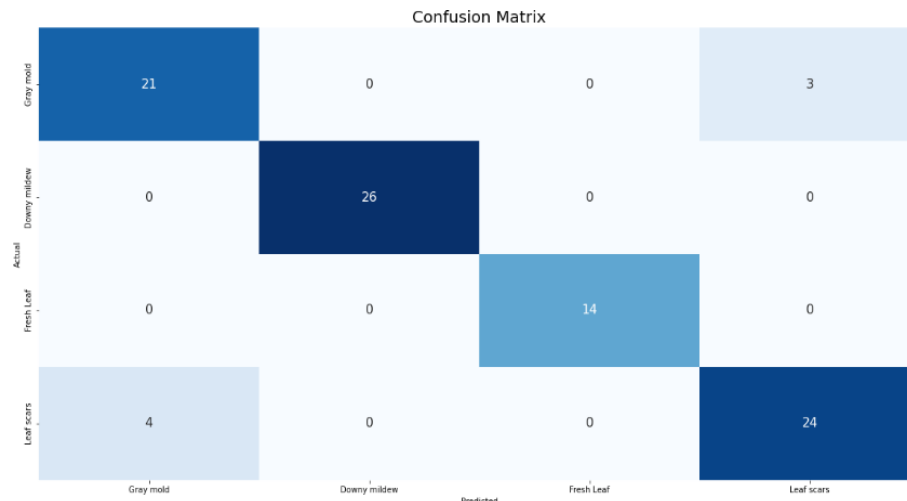


Fig 4: Confusion Matrix of the Projected model

Figure 5 provides the AUC curve of the dataset using a projected model.

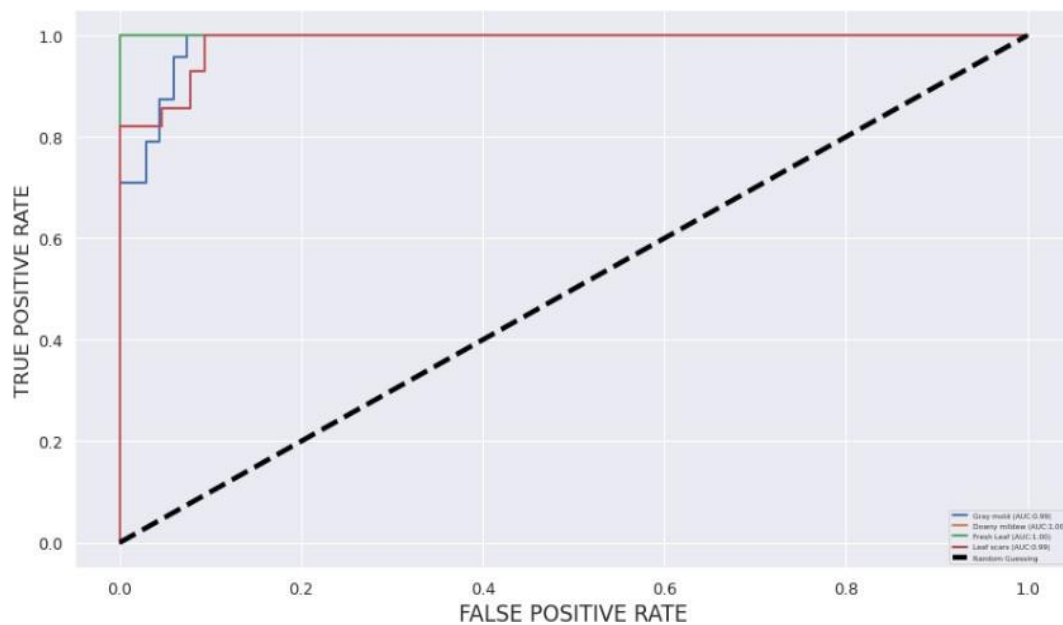


Figure 5: ROC AUC Curve of the data

Table 1: Comparative analysis of the Projected segmentation model with existing procedures

Classifiers	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)
CNN	84.27	72.45	76.34	72.45
RNN	88.67	87.91	78.75	78.35
LSTM	73.55	74.18	79.79	79.25
YOLO	70.78	88.27	82.51	89.20
Proposed Model	95.87	92.26	93.98	94.64

In the above Table, 1 represents the Comparative investigation of Projected segmentation perfectly with

existing techniques. In this analysis, the using different classifier models such as CNN, RNN, LSTM, and YOLO

with the Proposed Model. In this analysis, the initial classifier model as CNN reached an accuracy of 84.27 and also a sensitivity value of 72.45 and the specificity value of 76.34, and finally an F1-score value of 72.45 respectively. The RNN model reached an accuracy rate of 88.67 and also a sensitivity value of 87.91 and the specificity value of 78.75 and finally an F1-score value of 78.35 respectively. LSTM reached an accuracy of 73.55 and also a sensitivity value of 74.18 and the specificity value of 79.79 and finally an F1-score value of 79.25

respectively. The YOLO model reached an accuracy of 70.78 and also a sensitivity value of 88.27 and the specificity value of 82.51 and finally an F1-score value of 89.20, respectively. The proposed Model reached an accuracy of 95.87 and also a sensitivity value of 92.26 and the specificity value of 93.98 and finally an F1-score value of 94.64 respectively. In this comparisons analysis, the proposed model reaches a better result than other compared classifier models.

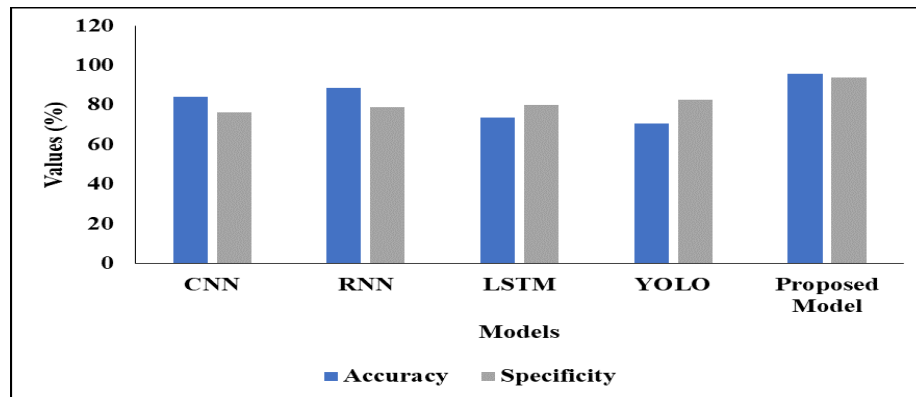


Fig 6: Graphical Comparison of different models

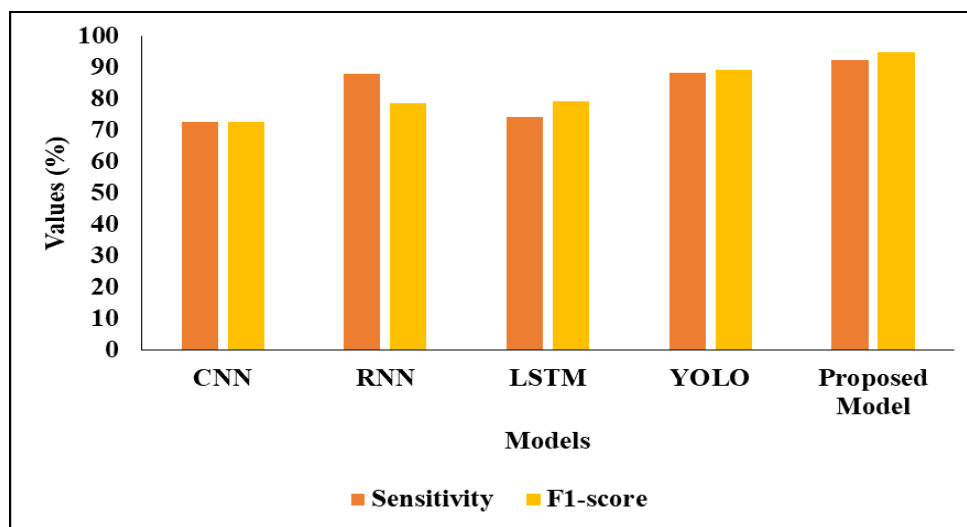


Fig 7: Analysis of various techniques in terms of different metrics

Table 2: Validation Analysis of Proposed Model on various disease classes

Classifiers	Healthy	Leaf Spot	Downy Mildew	Phoma Blight	Verticillium wilt
Accuracy	76.25	75.6	72.9	72.5	73
Sensitivity	78.75	78.4	76.3	77.5	75
Specificity	79.44	80.13	79.95	80.87	79.44
F1-score	83.05	82.07	82.95	83.34	81.92

In the above Table, 2 represents the Validation Analysis of the Proposed Model on various disease classes. The different diseases such as Healthy, Leaf Spot, Phoma Blight, and Verticillium wilt. In the healthy leaf condition, the accuracy value of 76.25 and the sensitivity value was 78.75 and the Specificity value was 79.44, and finally, the F1-score was 83.05 respectively. And the Leaf Spot condition has an accuracy value of 75.6 and the sensitivity value of 78.4 and the Specificity value of 80.13 and finally, an F1-score of 82.07 respectively. And the Downy

Mildew condition has an accuracy value of 72.9 and the sensitivity value of 76.3 and the Specificity value of 79.95 and finally, an F1-score of 82.95 respectively. And the Phoma Blight condition has an accuracy value of 72.5 and the sensitivity value of 77.5 and the Specificity value of 80.87 and finally, an F1-score of 83.34 respectively. And finally, the Verticillium wilt conditions an accuracy value of 73 and the sensitivity value of 75 and the Specificity value of 79.44, and finally, an F1-score of 81.92 respectively.

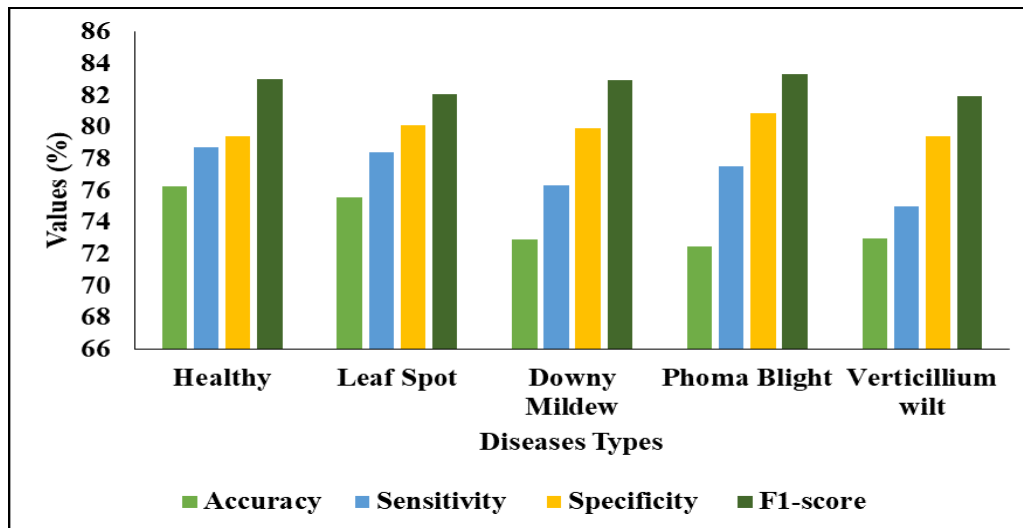


Fig 8: Analysis of the Proposed model on various disease types

5. Conclusion

In this study, The IDMO to assist us propose a YOLO model that is best for segmenting sunflower leaf diseases. It utilized this model to divide sunflower leaves into five groups: healthy, infected with one of four diseases, and infected with all four. The processed photos were analysed with the Improved-YOLOv5 model to locate the flaws. The modified the YOLOv5 in several ways to boost the model's recognition accuracy at each step. The step conducted extensive trials and testing to show that our model greatly outperformed the competition. The above-mentioned studies show that there are substantial variances across kinds within categories, whereas the gaps between disease types are often narrow. It's simple to confuse one for the other throughout the identifying procedure. So, to further increase the recognition accuracy, future studies may be conducted on other sorts of flaws using relevant data augmentation and other approaches.

References

- [1] Chen S, Huo P, "Improved detection of yolov4 sunflower leaf diseases", In 2021 2nd International Symposium on Computer Engineering and Intelligent Communications (ISCEIC) pp. 56-59, 2021.
- [2] Bantan R.A, Ali A, Naeem S, Jamal F, Elgarhy M, and Chesneau C, "Discrimination of sunflower seeds using multispectral and texture datasets in combination with region selection and supervised classification methods", Interdisciplinary Journal of Nonlinear Science, Volume 30(11), 2020.
- [3] Kurtulmuş F, "Identification of sunflower seeds with deep convolutional neural networks", Journal of Food Measurement and Characterization, Volume 15(2), pp.1024-1033,2021.
- [4] Çetin N, Karaman K, Beyzi E, Sağlam C. and Demirel B, "Comparative evaluation of some quality characteristics of sunflower oilseeds (*Helianthus annuus* L.) through machine learning classifiers", Food Analytical Methods, 14, pp.1666-1681,2021.
- [5] Shafique S, Javed A, Shafique S, Hussain A, Rafique R. and Mubarak A, "Isolation and identification of *Pithomyces saccharin* as a leaf spot pathogen of *Helianthus annuus* from Pakistan", Scientific Reports, Volume 12(1), 2022.
- [6] Lakshmi Prasad M.S, Naresh N, Sujatha K, Usha D, Sujatha M, Sarada C, Rao S.C. and Chowdappa P, "Population structure of *Alternaria* species causing leaf blight of sunflower (*Helianthus annuus* L.) in India". *Phytoparasitica*, Volume 48, pp.335-356,2020.

- [7] Min B, Kim T, Shin D. and Shin D, “Data Augmentation Method for Plant Leaf Disease Recognition”, *Applied Sciences*, olume13(3), 2023.
- [8] Ahmed A.A. and Reddy G.H, “A mobile-based system for detecting plant leaf diseases using deep learning”, *AgriEngineering*, Volume 3(3), pp.478-493,2021.
- [9] Arsenovic M, Karanovic M, Sladojevic S, Anderla A. and Stefanovic D, “Solving current limitations of deep learning-based approaches for plant disease detection”, *Symmetry*, Volume 11(7), 2019.
- [10] Saleem M.H, Khanchi S, Potgieter J. and Arif K.M, “Image-based plant disease identification by deep learning meta-architectures”, *Plants*, Volume 9(11), 2020.
- [11] Roy A.M. and Bhaduri J, “A deep learning-enabled multi-class plant disease detection model based on computer vision”, Volume 2(3), pp.413-428, 2021.
- [12] Singh V, “Sunflower leaf disease detection using image segmentation based on particle swarm optimization”, *Artificial Intelligence in Agriculture*, Volume 3, pp.62-68,2019.
- [13] Pandian J.A, Kanchanadevi K, Kumar V.D, Jasińska E, Goño R, Leonowicz Z. and Jasiński M, “A five convolutional layer deep convolutional neural network for plant leaf disease detection”, *Electronics*, Volume 11(8), 2022.
- [14] Ghosh P, Mondal A.K, Chatterjee, S, Masud M, Meshref H. and Bairagi A.K, “Recognition of Sunflower Diseases Using Hybrid Deep Learning and Its Explain ability with AI”, *Mathematics*, Volume 11(10), 2023.
- [15] Barrio-Conde M, Zanella M.A, Aguiar-Perez J.M, Ruiz-Gonzalez R, and Gomez-Gil J, “A Deep Learning Image System for Classifying High Oleic Sunflower Seed Varieties”, *Sensors*, Volume 23(5), 2023.
- [16] Dawod R.G. and Dobre C, “Automatic Segmentation and Classification System for Foliar Diseases in Sunflower”, *Sustainability*, Volume 14(18), 2022.
- [17] Malik A, Vaidya G, Jagota V, Eswaran S, Sirohi A, Batra I, Rakhra M and Asenso E, “Design and evaluation of a hybrid technique for detecting sunflower leaf disease using a deep learning approach”, *Journal of Food Quality*, pp.1-12, 2022.
- [18] Sara U, Rajbongshi A, Shakil R, Akter B, Sazzad S and Uddin M.S, “An extensive sunflower dataset representation for successful identification and classification of sunflower diseases”, *Data in Brief*, Volume 42, 2022.
- [19] Dawod R.G. and Dobre C, “ResNet interpretation methods applied to the classification of foliar diseases in sunflowers”, *Journal of Agriculture and Food Research*, Volume 9, 2022.
- [20] Song Z, Wang P, Zhang Z, Yang S and Ning J, “Recognition of sunflower growth period based on deep learning from UAV remote sensing images”, *Precision Agriculture*, pp.1-22, 2023.
- [21] Liu X, Li G, Liu L, Wang Z, “Improved YOLOV3 target recognition algorithm based on adaptive edge optimization”, *Microelectronics Computing*, Volume 36, pp: 59–64, 2019.
- [22] GitHubYOLOV5-Master, Available online: <https://github.com/ultralytics/yolov5.git/> , 2021.
- [23] Kim D, Park S, Kang D, Paik J, “Improved centre and scale prediction-based pedestrian detection using convolutional block”, In *Proceedings of the 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pp. 418–419, 2019.
- [24] He K, Zhang X, Ren S, Sun J, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, *IEEE Transactions on pattern analysis and machine Intelligence*, Volume 37, pp: 1904 - 1916, 2015.
- [25] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, Berg A, “SSD: Single Shot multibox Detector”, In *Proceedings of the European Conference on Computer Vision*, Springer, pp. 21–37, 2016.
- [26] Wang W, Xie E, Song X, Zang Y, Wang W, Lu T, Yu G, Shen C, “Efficient and accurate arbitrary-shaped text detection with pixel aggregation network”, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8440–8449, 2019.
- [27] Zeiler M D, Taylor G W, Fergus R, “Adaptive deconvolutional networks for mid and high-level feature learning”, In *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2018–2025, 2011.
- [28] Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q, “ECA-Net: Efficient channel attention for deep convolutional neural networks”, In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [29] Premkumar K, Vishnupriya M, Babu T S, Manikandan B V, Thamizhselvan T, Ali A, N. Islam, M R, Kouzani, “Black widow optimization-based optimal PI-controlled wind turbine emulator”, *Sustainability* Volume 12(24), pp: 1–19, 2020.
- [30] Hashim F A, Houssein E H, Mabrouk M S, Al-Atabany W, & Mirjalili S, “Henry Gas solubility optimization: A novel physics-based algorithm”, *Future Generation Computer Systems*, Volume 101, pp: 646–667, 2019.