

Received 19 June 2024, accepted 3 July 2024, date of publication 10 July 2024, date of current version 22 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3426041

## RESEARCH ARTICLE

# Classification of Freshwater Fish Diseases in Bangladesh Using a Novel Ensemble Deep Learning Model: Enhancing Accuracy and Interpretability

ABDULLAH AL MARUF<sup>1</sup>, SINHAD HOSSAIN FAHIM<sup>2</sup>, RUMAISHA BASHAR<sup>3</sup>,  
ROWNUK ARA RUMY<sup>2</sup>, SHAHARIOR ISLAM CHOWDHURY<sup>1</sup>,  
AND ZEYAR AUNG<sup>4,5</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Computer Science and Engineering, Bangladesh University of Business and Technology (BUBT), Dhaka 1216, Bangladesh

<sup>2</sup>Department of Computer Science and Engineering, Stamford University Bangladesh, Dhaka 1217, Bangladesh

<sup>3</sup>Department of Computer Science and Engineering, International Islamic University Chittagong, Chittagong 4318, Bangladesh

<sup>4</sup>Center for Secure Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, United Arab Emirates

<sup>5</sup>Department of Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding author: Zeyar Aung (zeyar.aung@ku.ac.ae)

This work was supported in part by Khalifa University, Abu Dhabi, United Arab Emirates.

**ABSTRACT** Effective disease management and mitigation strategies for fish diseases depend on timely and accurate diagnosis. In recent years, artificial intelligence methods—classification algorithms in particular—have become effective instruments for automating fish disease diagnosis. This paper presents two types of ensemble models: i) the baseline averaged ensemble (AE) model and ii) the novel Performance Metric-Infused Weighted Ensemble (PMIWE) model. By leveraging pre-trained models and novel ensemble techniques, we achieve a testing accuracy of 97.53%, corresponding precision, recall, and F1-score of 97%. We also bring about enhanced interpretability and trustworthiness using the Grad-CAM (Gradient-weighted Class Activation Mapping) explainable artificial intelligence (XAI) technique.

**INDEX TERMS** Deep learning, transfer learning, ensemble model, fish diseases, aquaculture, Grad-CAM.

## I. INTRODUCTION

Aquaculture, or fish farming, has evolved from a mostly small-scale, non-commercial, and family-oriented activity into the large-scale commercial and industrial production of high-value species that are exported at national, regional, and international levels [1]. It is developing, expanding, and intensifying in almost all regions of the world [2]. Sustainable food production is essential as the global population grows to 10 billion by 2050 and animal protein demand rises 52%.

By efficiently expanding fish sources, aquaculture helps to meet the expanding demand for fish. Its performance depends on its ability to organize environmental sustainability, social responsibility, food safety, and animal care. The observed trend of growth shows the increasing importance

of aquaculture for meeting the worldwide demand for fish. In 2020, the worldwide harvest of fish amounted to 179 million tones, with aquaculture producing 88 million tones, which is equivalent to 47% of the overall production. The significant production resulted in a revenue of US\$265 billion, reflecting its global economic significance [1].

Fish farming is undoubtedly a substantial sector in Bangladesh, where rural freshwater aquaculture constitutes the majority of fish production [3]. In the field of aquaculture and fisheries, Bangladesh has achieved considerable recognition. The FAO's State of World Fisheries and Aquaculture 2022 Report [4] ranks Bangladesh third for inland fish capture production and fifth for aquaculture production globally, and 3.50 percent of the country's GDP comes from aquaculture. The profitability of the aquaculture industry depends greatly on the use of strong management techniques based on an in-depth knowledge of

The associate editor coordinating the review of this manuscript and approving it for publication was Muammar Muhammad Kabir<sup>id</sup>.

fish physiology and environmental dynamics in aquaculture environments.

However, the sector faces challenges arising from the emergence of diseases caused by infections, germs, parasites, and insufficient nutritional products, leading to reduced productivity. To minimize the impact of diseases in aquaculture sectors, it is essential to identify symptoms of disease quickly. The conventional approaches of detection, such as removing fish flesh and testing the water for contaminants (pH, BOD, dissolved oxygen levels, nitrate level), can be quite laborious, costly, and even harmful at times, and those are referred to as indirect methods [2].

However, new possibilities have arisen because of developments in computer technology and image processing. Capturing and processing a wide variety of images has become possible, including those taken with cameras, microscopes, spectral analyzers, ultrasounds, and fluorescence detectors. Segmentation, feature extraction, and further processes will easily help to identify features like shape, texture, and color, which will ultimately lead to the detection of fish diseases. More efficient and accurate disease diagnosis in fish populations is made possible by the current technology, which is less expensive and takes less time [5].

The latest developments in computer vision and artificial intelligence greatly facilitate the development of intelligent aquaculture systems for the early detection of fish diseases. Different methods of identifying fish diseases have already been used. These methods include detecting abnormal behaviour [6], predicting the quality of the water [6], and measuring dissolved oxygen [7], [8], [9], [10], [11], [12].

The physical characteristics of fish, such as body texture, eye color, head appearance, fins, scales, gills, and tail, are used by computer vision-based direct approaches to fish disease detection to identify the illness. Unfortunately, these studies are still relatively few. Furthermore, most of the existing studies only cover certain classes of diseases.

Machine learning and deep learning models have been applied to identify fish diseases, but oftentimes, it is not easy to interpret the model's decisions, especially for deep learning. With the high accuracy of deep learning models, interpretability techniques have started to apply in this field. Fish disease recognition faces numerous challenges due to the complex underwater environment, which includes brightness imbalances, abrupt changes in fish position, movement of aquatic plants, fish texture and shape, seabed structure, poor image quality, and a lack of diversity in the dataset [2].

There is very limited prior research on Bangladeshi aquaculture using artificial intelligence techniques. In Bangladesh, where fish farming is a substantial sector, only one notable study has been conducted. Sikder et al. [4] used 350 images to achieve an accuracy of 97.90%. However, they did not address the need for dataset expansion or augmentation, limiting its effectiveness in practical applications. Furthermore, they faced biased results with an overfitted model, and there was a lack of explanation of the results.

Studies in other geographical regions have also used relatively small datasets, which can lead to overfitting and limited generalization. For example, Huang and Khabusi [2] used 649 images to classify six classes of fish diseases with an accuracy of 94.28% without employing extensive data augmentation, resulting in models that may not perform well on unseen data. Mia et al. [13] used 485 photos to classify three types of fish infections and achieved an accuracy of 88.87% using a Random Forest Classifier. Additionally, some studies focused on specific regions or disease types, limiting their applicability to broader contexts.

To address the above limitations, we aim to build a computer vision-based ensemble deep learning model to detect diseases in freshwater fish in Bangladesh using their still images as inputs. Our design objective is to build a model that is well-generalized and reliable by employing a dataset-diversifying strategy. In order to be useful in practical applications, the model also needs to have a low computational cost and be trustworthy to its end users.

In this paper, the novelty is our proposed Performance Metric-Infused Weighted Ensemble (PMIWE) model. While deep learning-based ensemble models have been the subject of numerous studies in various fields, our suggested model is unique because it uses a different weight calculation mechanism. In the PMIWE scheme, we used the hyperbolic tangent (tanh) function to dynamically determine the base learner weights rather than calculating them simplistically. This novel approach improves the ensemble model's overall efficacy and robustness by ensuring that the weights accurately reflect each base learner's performance based on the multiple metrics. The tanh function is a great option for dynamically modifying weights based on multiple performance metrics because of its smooth gradient and capacity to handle a broad range of input values. This leads to improved accuracy and robustness in classifying diseases affecting Bengali freshwater fish and a more responsive and adaptive ensemble model that can better integrate the strengths of various architectures. Furthermore, the research applied Explainable artificial intelligence (XAI) techniques to increase trustworthiness.

## A. RESEARCH OBJECTIVES

Our research objectives are elaborated in an itemized form below.

- **Dataset expansion and diversity:** Our research aims to work on the diverse dataset of various fish species, diseases, environmental conditions, and image attributes. With the need for more diversity in the dataset, we have found that maximum work has been done on limited samples, so increasing the dataset is one of the major aims of our research.
- **Model generalization and reliability:** Handle cases where models work well on training data but have trouble generalizing to new data. This is known as model generalization. Reducing overfitting and underfitting

with high testing accuracy can make the model more robust and generalized.

- **Efficiency Optimization:** To reduce the time and computational resources needed for model implementation and training, we look into lightweight architectures and optimization techniques. Since most deep learning architectures are prone to high computational complexity, we have attempted to reduce it while maintaining high accuracy so that deployment can be simple and less expensive.
- **Trustworthiness:** Investigating strategies to understand classification choices and to enhance the model's credibility is also a part of our research. Implementing the XAI techniques is our research aim to make the model trustworthy and analyze the biases.

## B. CONTRIBUTIONS

In pursuing the research objectives mentioned above, our significant research contributions are listed below.

- We used offline data augmentation techniques to increase the small dataset, making it larger and more diverse, with a particular focus on fish diseases in Bangladesh. Since fish diseases are highly region-specific, such as environmental conditions, our research significantly impacts Bangladeshi aquaculture.
- As our base learners, we have employed the transfer learning models. Instead of using them in their vanilla forms, we have meticulously customized them to lower the models' sizes and computational expenses.
- We have proposed a novel ensemble model, namely, the Performance Metric-Infused Weighted Ensemble (PMIWE). Several research works used CNN as base learners to create the deep learning ensemble. But virtually all of them are based on the conventional ensemble strategies of bagging, boosting, and stacking. Here, we employed a novel strategy of forming a weighted ensemble based on multiple performance evaluation metrics of the base learners using the tanh function. Ultimately, the suggested model decreases computational cost while increasing accuracy. After evaluating our model according to a number of criteria, we discovered that it is not biased or overfitted.
- We used Grad-CAM (Gradient-weighted Class Activation Mapping) to make sure our models were dependable and trustworthy. This method highlights regions of interest in input images that are most important to the model's decision-making process, thereby providing visual explanations of model predictions. We improve our models' interpretability by adding Grad-CAM.

The overview of the research is presented graphically in Figure 1.

## II. RELATED WORK

### A. MACHINE/DEEP LEARNING METHODS FOR FISH DISEASE DETECTION

Recent research has shown the importance of unsupervised learning methods, particularly clustering, in dealing

with spotted diseases that cause fish mortality. Clustering objects in images helps fish disease detection, according to Lyubchenko et al. [14], who use public datasets to automate disease diagnosis.

Barik et al. [15] conducted another study to identify and classify aquatic animal spot disease using unsupervised methods. Their study on image processing methods, including unsupervised clustering like K-means, shows how segmentation may separate areas affected by white spot disease. These results highlight the importance of segmentation in disease diagnosis and help reduce spot disease-related fish mortality in aquaculture.

Similarly, Sikder et al. [4] used unsupervised learning and binary image classification to identify freshwater fish diseases in ecologically vital Bangladeshi areas. The combination of K- and C-means fuzzy logic clustering with Multi-Support Vector Machines (M-SVMs) shows that clustering can extract features and improve classification accuracy, ensuring a method for precise and automated fish disease detection. These results show that clustering approaches may improve fish disease diagnostics and emphasize the need for further studies to improve detection and classification accuracy.

Ahmed et al. [16] have developed a dataset of infected and disinfected salmon fish focusing on early detection of fish diseases, particularly in salmon. They performed several image preprocessing techniques like histogram equalization, RGB conversion, segmentation, etc. They used the SVM algorithm for classification after performing image preprocessing. To make input images more adaptable, they used k-means segmentation, cubic spline interpolation, and adaptive histogram equalization. The SVM model classified fish diseases with 91.42% and 94.12% accuracy on the new dataset, with and without performing the augmentation.

Instead of detecting the fish diseases from images or classifying the diseases between several classes for identification, Nayan et al. [17] analyzed the water quality to find the diseases among fishes. They used the Gradient Boosting Model (GBM) in their research. On the other hand, instead of one, Mia et al. [13] explored the implementation of several classification algorithms, including Logistic Regression (LR), GB, SVM, Random Forest (RF), and KNN, for analyzing two features sets, a co-occurrence matrix and statistical characteristics. The results of the evaluations demonstrated that RF has a significant accuracy rate of 88.87% when comparing the performance of classifiers across different matrix configurations. Although the accuracy achieved is impressive, there are opportunities for improvement in future studies that include the expansion of datasets and the inclusion of a wider range of fish illnesses.

Malik et al. [18] have proposed an innovative approach to detect Epizootic Ulcerative Syndrome (EUS), caused by *Aphanomyces invadans* that look like ulcers. Their approach involved image segmentation followed by edge detection and morphological techniques for enhancement. Their experiment showed that FAST-PCA-NN outperformed

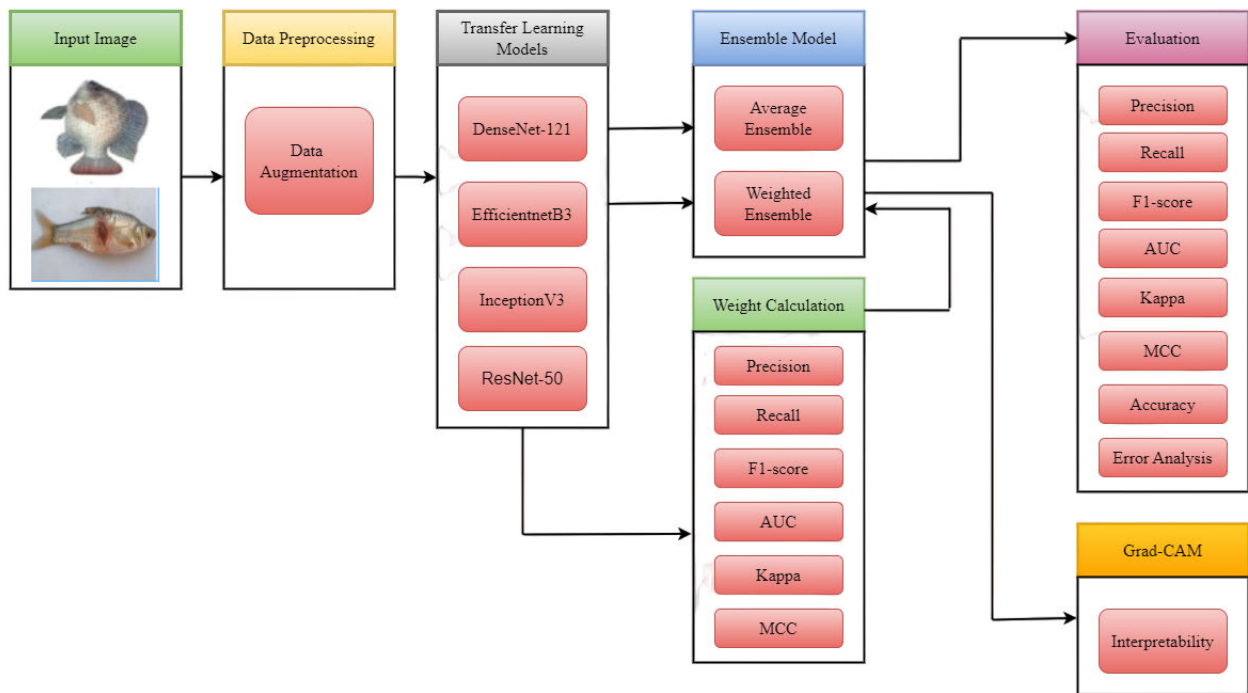


FIGURE 1. Overview of fish diseases classification using ensemble deep learning.

other combinations on real images from an EUS-infected fish. Using the ML algorithm with multiple feature descriptors and Principal Component Analysis (PCA) improves the accuracy of fish disease diagnosis.

For the identification of EUS, Kumar and Sahoo [19] came up with a new way to get the Region of Interest (ROI) from images of diseased fish by using feature descriptions like FAST and HOG. Applying PCA to reduce the number of dimensions leads to significant changes in accuracy, with FAST-PCA-NN doing 20.2% better than HOG-PCA-NN. It shows how feature descriptions and neural networks can help improve the accuracy and speed of identifying EUS-infected fish.

Fish disease identification was improved using hybrid classification models consisting of artificial neural networks (ANN) and classification trees (CT), which are essential for effective fish tracking and meeting global production needs [20]. This research presents a hybrid model to address underwater image difficulties during fish identification. It integrates CT for feature selection and ANN for classification, outperforming individual techniques, and the hybrid model achieved an excellent training accuracy of 93.6%. However, overfitting was observed in their model from their testing accuracy of 78.0%.

In general, the methods based on conventional image feature/descriptor extraction (such as [4], [13], [14], [15], [16], [17], [18], [19], [20]) are not able to offer a high level of accuracy as compared to deep learning methods.

Recently, Huang and Khabusi [2] have developed an innovative approach using multi-layer attention fusion for

classifying images by combining Convolutional Neural Networks (CNN) with the O-SLEM (Optimized Sparse Localized Energy Model) classifier, which has significantly improved the accuracy. This model performs precise preprocessing, segmentation, and feature extraction using a well-chosen dataset of 5,165 images, enhanced from 649 originals from the internet, natural habitats, and aquariums. Its 94.3% accuracy comes from 3:1:1 training, validation, and test sets. This shows its significant architecture and capacity to automatically identify fish diseases from underwater images, boosting computer vision and recognition of objects in an affected underwater environment. One of the major advancements of the research is working with many diseases. However, being based on CNN and attention mechanism, the downside of the method is that it consumes a lot of computational power and requires much care in choosing the parameters of the model.

Fish diseases have raised death rates in aquaculture, and the need for early identification is necessary to take precautionary steps. Addressing these challenges, Waleed et al. [21] outlined a novel strategy for automatically diagnosing three fish diseases: EUS, *Ichthyophthirius* (Ich), and *Columnaris*. Their novel strategy consists of pre-processing and Gaussian distribution segmentation in the YCbCr colorspace to evaluate how various color spaces affect CNN performance. According to their research, fish diseases can be detected in aquaculture with 95.86% to 99.04% accuracy using pre-trained CNN architectures, which include AlexNet, ResNet18, ResNet-50, and ResNet101. This profound study tackles the inadequacies of existing diagnostic approaches



and shows how artificial intelligence and computer vision might revolutionize aquaculture.

Another study [22] demonstrated that the application of pre-trained CNN models improved the detection of fish diseases, specifically. By implementing Object Segmentation Inference with the MobileNetV2 transfer learning model, the research achieves an impressive 84% precision rate when differentiating EUS, a critical factor in the early identification of such microorganisms in fish aquaculture.

In general, CNN and the CNN-based transfer learning models (such as AlexNet or ResNet) can improve the model accuracy with relatively small datasets by using pre-trained models, although they tend to outperform in tasks that are very similar but need further optimization when it comes to training time and resources. CNN and its variants guarantee a high level of accuracy if used with extensive datasets and can function optimally when exposed to distinct data structures. However, they are more reliable in cases of balanced and appropriately annotated data sources. Each deep learning method presented above ([2], [21], [22]) has some strengths and weaknesses depending on the type and size of the dataset. Several factors should be considered to enhance the performance of the model.

At any rate, there is an evident shortage of research using artificial intelligence (AI) techniques in the field of fish disease detection, which emphasizes the urgent need for more investigation and creativity. The shortcomings and gaps that have been found are listed here:

- Numerous studies made use of publicly available datasets or datasets that were country- or disease-specific. To overcome this constraint, more extensive and varied datasets covering a greater variety of fish species, illnesses, environmental factors, and image characteristics must be gathered. By doing this, it would be possible to guarantee that models are trained on a more representative set of data, improving their performance and generalization in real-world situations. We have only discovered one study in Bangladeshi freshwater since fish diseases are dependent on geographic location and water quality. However, they did not take any steps to expand the dataset, such as data augmentation, in place of the limited dataset.
- We discovered a few problems where the model worked well on training data but had trouble generalizing to new data. High testing accuracy without overfitting or underfitting problems was noted in some studies. Still, even though they frequently achieve high accuracy, Deep learning models can be very complex and difficult to interpret. Developing techniques to decipher model predictions and comprehend the underlying characteristics influencing classification decisions is necessary to overcome this constraint. Model interpretability could be improved by employing strategies like attention mechanisms, feature visualization, or model explanation techniques, increasing the models' reliability and suitability for practical applications.

- Certain models might require a significant investment in computing power and time to train and implement, particularly when working with sizable datasets or intricate architectures. This constraint must be overcome by investigating lightweight architectures, optimizing models for efficiency, or utilizing methods like model compression and transfer learning.

## B. ENSEMBLE DEEP LEARNING METHODS

To our knowledge, there is no prior work on the use of ensemble deep learning in the field of fish disease detection. However, a number of research works have been conducted to investigate the efficacy of deep learning ensemble models in various fields such as medical imaging [23], [24], [25], [26], [27], [28]. While some studies used average ensemble models [24], [25], others looked into the option of stacking ensemble methods [23]. The decision between these approaches is frequently based on the study's unique aims as well as the complexities of the dataset under consideration.

In this investigation, we employed a novel weighted ensemble model based on multiple performance metrics. Compared with the average and stacking ensemble models, the weighted ensemble models have the superiority of assigning different weights for base models according to their performance, which makes them more flexible and has the potential to improve the model's performance. Usually, if implemented through cross-validation or other procedures, more precise models can make a substantial contribution to the final prediction since these weights can be adjusted to achieve greater efficiency compared to average ensembles where all the models have equal weight. Weighted ensembles avoid the pitfalls of stacking ensemble models, such as increased complexity and susceptibility to overfitting due to training a meta-model on predictions from base models, which makes them a more straightforward yet effective choice for ensemble learning in many scenarios.

On the other hand, some prior research contributed to weighted ensemble models [26], [27], [28] where the determination of model weights deviates from simplistic calculation methods observed in prior research. In our study, we introduce a novel approach to weight calculation named the hyperbolic tangent (tanh) function based on multiple performance metrics. This innovative mechanism offers a distinct advantage over traditional simplistic weighting methods by enriching the ensemble model's predictive capabilities. Our research extends the frontier of ensemble learning in image classification by presenting a promising avenue for further exploration and advancement.

## III. DATASET AND PREPROCESSING

### A. DATASET DESCRIPTION

The Fish Dataset was collected from the Kaggle website on "Fresh Water Fish Disease Dataset," which was uploaded in 2022 (<https://www.kaggle.com/datasets/utpolkantidas/fresh-water-fish-disease-dataset>).

This image data was collected from Tangor Haor and Kaptai Lake, comprising a total of 133 images, categorized into seven distinct classes, namely, (i) Argulus (23 images), (ii) Broken Antennae and Rostrum (7 images), (iii) EUS (Epizootic Ulcerative Syndrome) (23 images), (iv) Healthy Fish (31 images), (v) Red Spot (31 images), (vi) Tail and Fin Rot (9 images), and (vii) Bacterial Gill Rot (6 images).

The purpose of the dataset is to facilitate research and analysis related to fish health and diseases. The description of the diseases and diseases included in the dataset are shown in Table 1.

## B. DATA AUGMENTATION

Models that need more data to reflect the population will be more likely to overfit accurately. This means that other techniques—aside from merely gathering more data—are needed to stop overfitting and promote generalization toward the practical use of the model. It can take a lot of effort and time to obtain high-quality annotated training data. “Augmentation” is an image data preprocessing technique that changes annotated data already in existence in order to give the neural network new information [34]. An example of this would be to flip an image and its ground truth from left to right.

Offline augmentation is the application of such manipulations prior to the commencement of the training [35]. In this way, the operator keeps authority over the dataset. The operator can examine the masks and augmented images beforehand to make sure the augmentations make sense. This method’s much greater storage space requirement is one drawback. It is not possible to store every possible combination of various augmentations for a single image. An alternative, less haphazard approach is used in the offline augmentation process to minimize the number of images saved while preserving a high degree of variety [36].

The pipeline is composed of two augmenters: a geometric augments that includes various operations such as elastic distortion, flip (height, width), grid distortion, grid shuffle, optical distortion, random crop, resize, rotation, squeeze (height, width), tilt (left, right, backward), and a pixel augments that includes multiple radiometric operations such as blur, brightness, channel shuffle, color to HSV (Hue, Saturation, and Value), contrast manipulation, fog, histogram normalization, noise, rain, random erasing, RGB shift, shadow, sharpen, snow, to grey, to sepia.

The following is how the algorithm is created: The geometric augments is programmed to produce  $n_{geo} = \lceil \frac{1}{2} \cdot n_{total} \rceil$  geometrically transformed versions of the input image after a maximum number of augmentations per image ( $n_{total}$ ) is specified. Then, for each generated geometric image, the pixel augments generates  $n_{pix} = n_{total} - n_{geo}$  versions in addition to  $n_{pix}$  versions of the input image. Since there are significantly more images produced as a result than in the set  $n_{total}$ , only  $n_{total}$  randomly chosen images are kept. The ultimate distribution, with respect to the set  $n_{total}$ , between only geometrically altered images and pixel-transformed

images for a single input sample. It is computed using:

$$P_{geo} = \frac{n_{geo}}{n_{geo} \times n_{pix} + n_{pix}} \quad (1)$$

$$P_{pix} = 1 - P_{geo} \quad (2)$$

In summary, performing the augmentation approach greatly increased our dataset while maintaining the image quality.

## IV. PROPOSED METHOD

### A. TRANSFER LEARNING MODELS

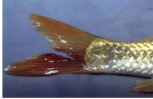






In the fish disease detection research, we first used transfer learning (a.k.a. pre-trained) models, namely ResNet-50 [37], DenseNet-121 [38], InceptionV3 [39], and EfficientNetB3 [40]. Pre-trained models are useful for various tasks because they have learned to identify common features in images, like edges, lines, and textures. The pre-trained models can be improved to identify more precise features and reach high accuracy on the new task using fewer training images by fine-tuning them on a fresh dataset of images.

The pre-trained models, DenseNet-121, ResNet-50, InceptionV3, and EfficientNetB3, were chosen for our study due to their strong architectures and excellent results in various computer vision tasks [41], [42], [43]. DenseNet-121 and ResNet-50 are renowned for their high accuracy and effective feature learning capabilities, making them ideal for capturing intricate details in fish disease images. InceptionV3 offers a balanced approach with its efficient architecture, adept at handling varying image scales and resolutions. EfficientNetB3, with its compound scaling method, provides state-of-the-art accuracy with minimal computational overhead. Our main intention was to increase accuracy while maintaining low computational cost. To achieve this, we applied a variety of architectures in our ensemble model, selecting models known for their efficiency and computational affordability.

#### 1) RESNET-50

ResNet-50 [37] has 50 layers, making it comparatively deep. It presented the idea of residual learning, which avoids the vanishing gradient problem by using shortcut connections. The first layer employs  $7 \times 7$  convolutions, while the residual blocks use  $3 \times 3$  convolutions. The residual block, which enables the network to skip layers and enable the training of very deep networks, is the main innovation of ResNet. The primary goal of Inception v3 is to reduce processing power consumption by altering the earlier Inception architectures. Factorized convolutions: by lowering the number of network parameters, they help to decrease computational efficiency. It also monitors the effectiveness of the network. Training proceeds faster when smaller convolutions replace larger convolutions with smaller convolutions. If two  $3 \times 3$  filters are used in place of a  $5 \times 5$  convolution, the number of parameters is reduced to 18 ( $3 \times 3 + 3 \times 3$ ) from the 25 in the case of a  $5 \times 5$  filter. One possible substitution for a  $3 \times 3$  convolution would be a  $1 \times 3$  convolution, followed by

TABLE 1. Overview of the diseases included in the dataset.

Preview	Name	Description
	Tail and Fin Rot	Tail and fin rot in fish is caused by bacterial or fungal infections, and it is considered a common and harmful disease. The fish's fins and tail are subject to deterioration and decay due to this condition. Fraying or disintegration of the fins may be observed in infected fish, and the affected areas may appear red or discolored. Poor water quality, injuries, or stress can be responsible for the occurrence of tail and fin rot [29].
	Bacterial Gill Rot	Bacterial gill rot in fish is caused by bacterial infections affecting the gills, and it is considered a serious and potentially fatal disease. The gill tissues deteriorate due to the disease, resulting in respiratory difficulties for the fish. Signs of infection may be observed in infected fish, such as gasping at the water's surface and reduced activity. The gills may be discolored, swollen, and may have a foul smell [30].
	Red Spot	Red spot in fish, also known as red spot disease or hemorrhagic septicemia, is a common fish disease caused by bacterial infections. It manifests as red or dark spots on the fish's body, particularly around the fins and scales [31].
	Healthy Fish	Optimal physical and behavioral characteristics are exhibited by healthy fish. They possess clear eyes, intact scales, and vibrant colors. Their fins are undamaged, and they actively and smoothly swim.
	Broken Antennae and Rostrum	The occurrence of broken antennae and rostrum in aquatic fish can be attributed to several factors, such as collisions, entanglement in fishing gear, and human activities. These anatomical structures play a crucial role in sensory perception and foraging [3].
	EUS	The oomycete known as <i>Aphanomyces invadans</i> or <i>A. piscicida</i> is thought to be the cause of epizootic ulcerative syndrome (EUS), which is histologically characterized by penetrating hyphae encircled by granulomatous inflammation. Fish in freshwater and estuaries, both wild and farmed, are affected by this epizootic condition [32].
	Argulus	Even without visible lice, fish can display nonspecific indications of infestation by <i>Argulus</i> (fish lice). Some of these symptoms are spot or pinpoint bleeding, anemia, loss of fin and scale, increased mucus production, sluggishness, erratic swimming, and poor physical condition [33].

a  $3 \times 1$  convolution. A  $2 \times 2$  convolution would require a few more parameters than the suggested asymmetric convolution in place of a  $3 \times 3$  convolution. A tiny CNN called an auxiliary classifier is placed in between layers during training, and the loss it experiences is added to the loss of the main network.

2) DENSENET-121

DenseNet is a special type of CNN architecture introduced by Huang et al. [44], consisting of dense blocks with multiple

bottleneck layers connected in a feed-forward manner to minimize the issue of vanishing gradients, which makes the training process easier. In the DenseNet architecture, each block is made up of several convolution layers that are then combined through a concatenation process, integrating the input from all previous layers within the block. This network consists of multiple layers, each consisting of a composite function denoted by  $H_i(\cdot)$ , which is a combination of operations like leaky rectified linear units named ReLU, Batch Normalisation, pooling, and convolution layers.

Here, the number of layers in this network is denoted by  $L$ . Then, the number of connections for this architecture is calculated using  $L(L + 1)/2$ , whereas in other CNNs, the number of connections is the number of layers. Implementing a nonlinear transformation  $H$  to the previous layer's output  $x_{l-1}$ , the obtained output for  $x_l$  is  $H_l(x_{l-1})$ .

For the connection with previous layers through concatenation, the output obtained for  $l^{\text{th}}$  layer that denotes the concatenation of the extracted features from the input of the previous layer is:

$$x_l = H_l([x_0, x_1, x_2, x_3, x_4, \dots, x_{l-1}])$$

where 0, 1, 2, 3, 4 and  $l - 1$  denotes the number of the previous layers and  $x_0, x_1, x_2, x_3, x_4$  are the individual extracted features from each of the previous input layer.

Again, each function  $H_l$  generates  $k$  feature maps. Consequently, the  $l^{\text{th}}$  layer has  $k_0 + k \times (l - 1)$  input feature maps, where  $k_0$  represents the number of channels in the input layer,  $k$  is denoted for growth rate of the network. Each layer provides  $k$  feature mappings to the global state. The growth rate determines the amount of new information that each layer adds to the global state from where every information can be accessed.

DenseNet-121 is a variant of DenseNet with 4 dense blocks and 3 transition layers. DenseNet-121 [38] has 121 layers, but because each layer is densely connected to every other layer, it has fewer parameters than ResNet.  $3 \times 3$  convolutional filters are used. Because each layer has direct access to the gradients from successive layers, DenseNet's architecture promotes feature reuse and lessens the issue of disappearing gradients.

### 3) INCEPTIONV3

An auxiliary classifier serves as a regularizer in InceptionV3 [39]. InceptionV3 aims to scale up networks efficiently by applying factorized convolutions, aggressive regularisation, and enhancements in the Inception architecture that deviated from traditional methods to improve network accuracy through its unique principles of avoiding representational bottlenecks and processing high-dimensional representations with ease [45], [46]. It employs low-dimensional embeddings for spatial aggregation without sacrificing representative capacity, as well as simultaneously increasing the network's width and depth to achieve optimal performance within a given computational budget. The InceptionV3 network's overall layout consists of a series of layers beginning with a  $3 \times 3.2$  conv2d layer with ReLU activation and batch normalization for an input size of  $229 \times 229 \times 3$ , followed by similar conv2d levels with patch size, stride, and input size adjustments. It includes three different Inception Modules being applied in succession (3 times Inception Module 1, 5 times Inception Module 2, and 2 times Inception Module 3), each with a max pooling layer, a linear logits layer, and a softmax classifier to handle inputs and output for classifications, encapsulating

the architecture's comprehensive design tailored for robust performance enhancements.

### 4) EFFICIENTNETB3

One member of the EfficientNet family that is particularly useful for transfer learning tasks—such as fish disease detection is EfficientNetB3 [40]. With its scalable architecture, EfficientNetB3 provides an attractive solution for a range of computer vision applications by skillfully balancing model size, accuracy, and computational efficiency. EfficientNetB3 maintains competitive performance by efficiently using computational resources through the use of techniques like depth, width, and resolution scaling. Its reliance on effective building blocks, such as MBConv, mitigates issues such as vanishing gradients by encouraging feature reuse and gradient flow, much like DenseNet. EfficientNetB3 reduces computational overhead while capturing spatial features effectively through the use of  $3 \times 3$  convolutional filters. EfficientNetB3, which takes its cues from models such as InceptionV3, incorporates auxiliary classifiers into its training to enhance regularization and generalization. It is useful in our research because it excels in achieving high accuracy with minimal training data and focuses on reducing computational costs through factorized convolutions.

### 5) CUSTOMIZING TRANSFER LEARNING MODELS

The architecture makes use of transfer learning with ImageNet-pre-trained models, a popular deep learning technique in which models that have been pre-trained on sizable datasets such as ImageNet are used as a basis for additional tasks [47]. By doing this, the network can apply the pre-trained model's knowledge of a wide range of features, improving generalization to new tasks with smaller datasets. When loading the pre-trained model, setting "include\_top=False" means that the fully connected layers—or "top" layers—are not included in the pre-trained model. This enables us to add and modify layers that are appropriate for the classification of fish diseases. We started with the base model and added the Global Average Pooling Layer (GAP) to the modified transfer learning model. The architecture then includes a GAP layer, which calculates the mean value of each feature map over all of its spatial dimensions. By combining all of the feature maps into a single one, global pooling allows for the easy comprehension of all pertinent data by a single dense classification layer as opposed to several layers. It can be used for both 1D and 3D input and is commonly implemented as either max pooling (GlobalMaxPooling2D) or average pooling (GAP2D). GAP is a pooling operation that can be used for transfer learning and is intended to replace the fully connected and flattened layers. GAP can be expressed as:

$$\text{GAP}(x_{ab}) = \frac{1}{h \times y} \sum_{a=1}^h \sum_{b=1}^y x_{ab} \quad (3)$$



In this case, the activation of a neuron in the  $a$ -th row and  $b$ -th column of the feature map is represented by  $x_{ab}$ , and the feature map's height and width are indicated by  $h$  and  $y$ , respectively. The GAP layer improves computational efficiency and lessens overfitting by shrinking the spatial dimensions of the feature maps while keeping relevant features.

$$\text{Dropout}(x) = \begin{cases} x & \text{with probability } 1-p \\ 0 & \text{with probability } p \end{cases} \quad (4)$$

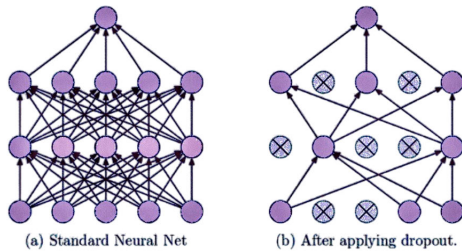


FIGURE 2. Before and after applying the dropout.

Then, a dropout layer is added, having a dropout rate of 0.2. During training, the Dropout layer arbitrarily sets a portion of the input units to zero, preventing the network from becoming overly dependent on any one set of features and thereby reducing overfitting. As shown in Figure 2, “dropout” describes the process of removing a neural network’s input and hidden layer nodes. A new network architecture is created out of the parent network by temporarily removing all forward and backward connections with a dropped node. There is a  $p$  dropout probability for the nodes.

The model picks up statistical noise in the overfitting issue. More specifically, given all the units (neurons), the primary goal of training is to reduce the loss of function. As a result, overfitting can cause a unit to adjust to correct the errors made by the other units. Because of the inability of these complex co-adaptations to generalize to the unknown dataset, they result in complex co-adaptations, which in turn cause the overfitting issue. Because a unit’s presence is highly unpredictable in every iteration, dropout prevents these units from correcting the mistakes made by other units, preventing co-adaptation. Thus, it forces the layers to adopt a probabilistic approach and assume some degree of responsibility for the input by arbitrarily dropping a few units (nodes). When the dropout layer was first implemented, a unit (node or neuron) in a layer was chosen with a keep probability (1-drop probability) during training. As a result, the training batch is given a thinner architecture, which varies with each training run. The following equations apply to the standard neural network during forward propagation:

$$z_i^{(l+1)} = w_i^{(l+1)}y^l + b_i^{(l+1)} \quad (5)$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \quad (6)$$

Here,  $\mathbf{z}$  represents the output vector from layer  $(l + 1)$  prior to activation,  $\mathbf{y}$  represents the output vector from layer  $l$ ,

$\mathbf{w}$  represents the layer’s weight, and  $\mathbf{b}$  represents the layer’s bias. Any activation function can be represented by  $f$ , as in  $f(x) = \frac{1}{1+\exp(-x)}$ .

$$r_j^{(l)} \sim \text{Bernoulli}(p), \quad (7)$$

$$y_i^{(l)} = r_j^{(l)} \cdot y_i^{(l)}, \quad (8)$$

$$z_i^{(l+1)} = w_i^{(l+1)}y_i^{(l)} + b_i^{(l+1)}, \quad (9)$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \quad (10)$$

The dense (fully-connected) layer is added at the end. After computing a weighted sum of its inputs, the dense layer activates the softmax, which turns the raw scores into probabilities. It is calculated as follows:

$$\text{Dense}(x) = \text{softmax}(Wx + b) \quad (11)$$

In this case, the input vector is denoted by  $x$ , the weight matrix by  $W$ , the bias vector by  $b$ , and the softmax activation function by softmax. Thanks to the final classification layer, which generates probabilities for the seven classes, the model can classify input images into the appropriate categories. The architecture makes the best use of transfer learning, tailoring it with GAP and Dropout layers to enhance generalization and prevent overfitting before classifying using the dense layer.

Using the “include\_top=False” option, which removes the fully connected top layers of the pre-trained models, we altered the transfer learning models in our study to minimize their size and computational costs. Drastically lowering the number of trainable parameters improved the models’ efficiency while preserving their potent feature extraction capabilities. We further minimize computational demands while maintaining high performance by swapping out these dense layers with lighter, custom classification layers for our particular task. This optimized method guarantees effective use of the available computing power, leading to shorter training times and lower inference latency. As a result, our computationally efficient models remain effective in classifying diseases affecting Bengali freshwater fish and are more suitable for implementation in resource-constrained environments.

This optimized computational efficiency for DenseNet-121 by reducing the trainable parameters to 3,058,759 out of a total of 7,044,679 parameters. With 7,286,905 trainable parameters out of 10,794,294 in its configuration, EfficientNetB3 was designed to be highly efficient. With 1,949,703 trainable parameters reduced from 21,817,127 in InceptionV3, the model became lighter and required less computing power. With the customization, ResNet-50 produced just 14,343 trainable parameters out of 23,602,055, indicating that the model is rapid and efficient. By substituting lighter, task-specific classification heads for the excluded top layers, we ensured that each model remained highly practical and computationally feasible for distinguishing diseases in Bengali freshwater fish.

Table 2 shows the computational details of pre-trained models.

**TABLE 2. Customized pre-trained models architecture and details. (Note: the training time is measured on the Google Colab Pro+ platform using Tensorflow/Keras as described in Section V-A.)**

Model	Training Time (Sec.)	Model Size (MB)	Total params	Trainable Parameters	Non-trainable parameters
DenseNet-121	700	26.87	7044679	3058759	3985920
EfficientNetB3	900	41.18	10794294	7286905	3507389
InceptionV3	500	83.23	21817127	1949703	19867424
ResNet-50	700	90.3	23602055	14343	23587712

## B. PROPOSED ENSEMBLE TECHNIQUE

We moved toward the transfer learning base ensemble after deploying the pre-trained models one at a time. The pre-trained model ensemble has shown to be more effective recently than the individual models [48], [49]. Ensemble models in deep learning are developed by combining predictions from multiple individual models, using different predicting algorithms or training datasets in order to obtain ideal final predictions and to reduce generalization errors [50].

Ensemble learning combined with pre-trained models has shown remarkable effectiveness, especially in tasks such as image classification, natural language processing [51], and time series forecasting [52]. It reduces variance, makes predictions more stable, and takes advantage of the valuable features extracted by pre-trained models trained on large datasets. The combined effort not only outperforms the latest models but also tackles the issue of overfitting by utilizing a diverse ensemble and the inherent adaptability of pre-trained models to smaller datasets.

One significant advantage is the capacity to use the information stored within pre-trained models, which is especially useful in scenarios with limited training data. These models provide advanced features that perform well at capturing complex relationships and are customized to enhance overall performance. Ensemble models are known for their ability to resist noise and variations, making them more robust. As we mentioned earlier, we have used 4 pre-trained models. The ensemble model was a combination of these four models.

In our research to classify fish diseases into seven distinct classes, we applied two types of ensemble models, namely, (i) the baseline averaged ensemble (AE) and (ii) the novel Performance Metric-Infused Weighted Ensemble (PMIWE). Each model is described here briefly.

### 1) BASELINE ENSEMBLE MODEL: AVERAGED ENSEMBLE (AE)

The baseline proposed ensemble model of our work is the averaged ensemble (AE) model. This model is a non-weighted ensemble model that combines our pre-trained models independently and calculates the average of the probability scores and the performance of the base learners in a relatively straightforward manner. When compared to a weighted ensemble, this one does not frequently require the initialization of weights, and predictions are not dependent on weights. This is because each model makes an equal

amount of contribution to the detection process. Individual pre-trained models have significant pros and cons, like low bias and high variance, which may lead to overfitting and accuracy-related issues. As this proposed approach combines various models having advantages and limitations, it reduces bias and variances, which ultimately helps in generalization across different datasets. Due to the integration of all models, it is less prone to overfitting, which ultimately improves accuracy.

Let us assume that  $n$  is the number of models, and the input samples are  $P_1, P_2, P_3, \dots, P_n$ . The general equation for the averaged ensemble model is:

$$P_{\text{ensemble}} = \frac{1}{n} \sum_{i=1}^n p_i \quad (12)$$

where  $p_i$  is the prediction of the  $i$ -th individual model, and  $P_{\text{ensemble}}$  is the ensembled prediction.

We used softmax scores obtained by these individual models and combined and averaged them to predict the highest outcome classes. The general equation for softmax( $z_i$ ) is:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (13)$$

where  $e^{z_i}$  is the exponential function for the individual input sample,  $k$  is the number of classes, and  $S(z_i)$  is the function for the output vector.

So, the calculated ensemble output is:

$$P_{mn} = \frac{e^{\text{softmax}(O_{mn})}}{\sum_{k=1}^k e^{O_{mk}}} \quad (14)$$

Here,  $P_{mn}$  is the probability outcome of the  $n$ -th unit on the  $m$ -th base learner,  $O_{mn}$  is the output of the  $n$ -th unit of the  $m$ -th base learner, and  $k$  is the number of classes ( $k = 7$  in our case).

### 2) NOVEL ENSEMBLE MODEL: PERFORMANCE METRIC-INFUSED WEIGHTED ENSEMBLE (PMIWE)

Initially, the primary results of the base models can affect the ensemble outcomes. Then, assigning different weights on the base models has a particular effect on the ensembled outcome. During training, base learners generate probability scores that can be used to calculate assigned weights. A model trained on the test dataset uses these weights throughout the ensemble. This accurate method assures the independence and reliability of predictions. The maximum weight given to the base learners has the most priority in a prediction system

of the ensemble model. Weighted ensembling is an effective method for classifier fusion. To ensure the ensemble's success, however, the weights assigned to the various base learners must be carefully considered. The majority of methods in the literature determine the weights through experimentation or by considering only the classifier's accuracy. When there is a class imbalance in the dataset, this may not be the best metric to use. To ascertain the base learners' priority, additional assessment metrics like F1-score, AUC, precision, recall, kappa and MCC may offer reasonably reliable data. In order to achieve this, we developed a novel weight allocation strategy in this study.

Initially, the weights assigned to each base learner using the suggested strategy are determined using the probability scores that the base learners obtained during the training phase. Using these generated weights, an ensemble trained on the test set is created. This tactic ensures that the test set is totally independent and that no information from it leaks.

During the evaluation, predictions ( $y_{\text{pred}}$ ) from the  $i$ -th model are generated. Compared with the true labels ( $y$ ) to compute the influential performance metrics, including precision score ( $prc^i$ ), recall scores ( $rec^i$ ), F1-score ( $F1^i$ ), MCC score ( $MCC^i$ ), and Cohen's Kappa ( $kappa^i$ ), which are described in detail in Section V-D. The weight  $W^i$  is assigned to each classifier and then calculated using the hyperbolic tangent function shown in Eq. 15. The hyperbolic tangent function,  $\tanh$ , is a mathematical operation that reduces input values into a specific range. In the context of our experiment, it appears to be in the range  $[0, 0.865]$ , although the usual range is typically  $[-1, 1]$ . In this scenario, the variable  $z$  represents an evaluation metric, with values limited to the range of 0 to 1.

The  $\tanh$  function is selected for its monotonic increase within the specified range, indicating that as metric values rise, the output of the  $\tanh$  function also increases. In practical terms, this translates to an approach where higher metric values are given more importance while lower values are dismissed. This approach is in line with the idea that higher metric values typically suggest superior performance in evaluation. When using a truncated range of  $[0, 0.865]$ , the  $\tanh$  function gives more importance to instances with metric values closer to 1. This means that it focuses on the cases that perform better within the overall metric range of  $[0, 1]$ .

$$w(i) = \sum_{z \in A(i)} \tanh(z) = \sum_{z \in A(i)} \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (15)$$

Eq. 15 assigns a weight  $w(i)$  to the  $i$ -th base learner in an ensemble model based on its performance metrics. Here,  $A(i)$  represents the set of evaluation metrics for the  $i$ -th base learner, such as precision, recall, F1-score, MCC, or Cohen's Kappa. Each metric  $z$  in this set is transformed using the hyperbolic tangent function,  $\tanh(z)$ , which scales  $z$  to a value in the range  $[0, 0.865]$ . The equation generates a weight  $w(i)$  that reflects the base learner's overall effectiveness, giving more importance to models with better performance metrics.

After that, we calculated the final prediction for ensemble models by using the process described below.

Firstly, we computed the probability scores of each base classifier for a specific class and then the prediction made by the ensemble model. For multi-class probability array:

$$p_j^{(i)} = [p_{j,1}^{(i)}, p_{j,2}^{(i)}, \dots, p_{j,C}^{(i)}]$$

The probability array for the  $j$ -th sample by the  $i$ -th base classifier ( $p_j^{(i)}$ ) is a vector of length  $C$  representing the probability distribution over the  $C$  classes for that sample.

When calculating the ensemble probability for multi-class scenarios, we used the probabilities of the base classifiers for each class. This is done by calculating a weighted average of these probabilities:

$$en_{p_j} = \left[ \frac{\sum_i w^{(i)}}{\sum_i w^{(i)} \cdot p_{j,1}^{(i)}}, \frac{\sum_i w^{(i)}}{\sum_i w^{(i)} \cdot p_{j,2}^{(i)}}, \dots, \frac{\sum_i w^{(i)}}{\sum_i w^{(i)} \cdot p_{j,C}^{(i)}} \right] \quad (16)$$

In Eq. 16,  $en_{p_j}$  represents the ensemble prediction for the  $j$ -th instance across  $C$  classes. Each element  $en_{p_j}[c]$  in  $en_{p_j}$  is computed as the inverse-weighted average of predicted probabilities  $p_{j,c}^{(i)}$  from  $i$ -th base learners. Here,  $\sum_i w^{(i)}$  sums the weights assigned to each base learner  $i$ , reflecting their importance based on performance metrics. The denominator  $\sum_i w^{(i)} \cdot p_{j,c}^{(i)}$  adjusts each  $p_{j,c}^{(i)}$  by its corresponding weight  $w^{(i)}$ , ensuring that better-performing base learners contribute more to the final prediction. This ensemble approach optimizes prediction accuracy by combining diverse model outputs while giving greater weight to more reliable predictions, as determined during training.

Ultimately, the ensemble model predicts the class using Eq. 17.

$$Prediction_j = \text{argmax}(\text{ensemble}_{p_j}) \quad (17)$$

We applied the  $\text{argmax}$  function to identify the class with the highest probability among the classes where  $prediction_j$  is the predicted class of the samples. The architecture of the proposed weighted ensemble model is shown in Figure 3.

The algorithm of the proposed Performance Metric-Infused Weighted Ensemble (PMIWE) model is described below.

## V. EXPERIMENTAL SETUP

### A. IMPLEMENTATION

Python is the main programming language used for fish disease detection using Google Colab Pro+. A number of libraries and frameworks were used in our research. Among the foundational frameworks, TensorFlow stands out for its strong capabilities in neural network model building and training. Keras is a high-level API that works in tandem with TensorFlow to enable quick experimentation with deep learning architectures on top of TensorFlow's backend. Scikit-learn's abundance of preprocessing, model evaluation, and selection tools enhances the ecosystem. NumPy's array-based operations improve computational efficiency, and

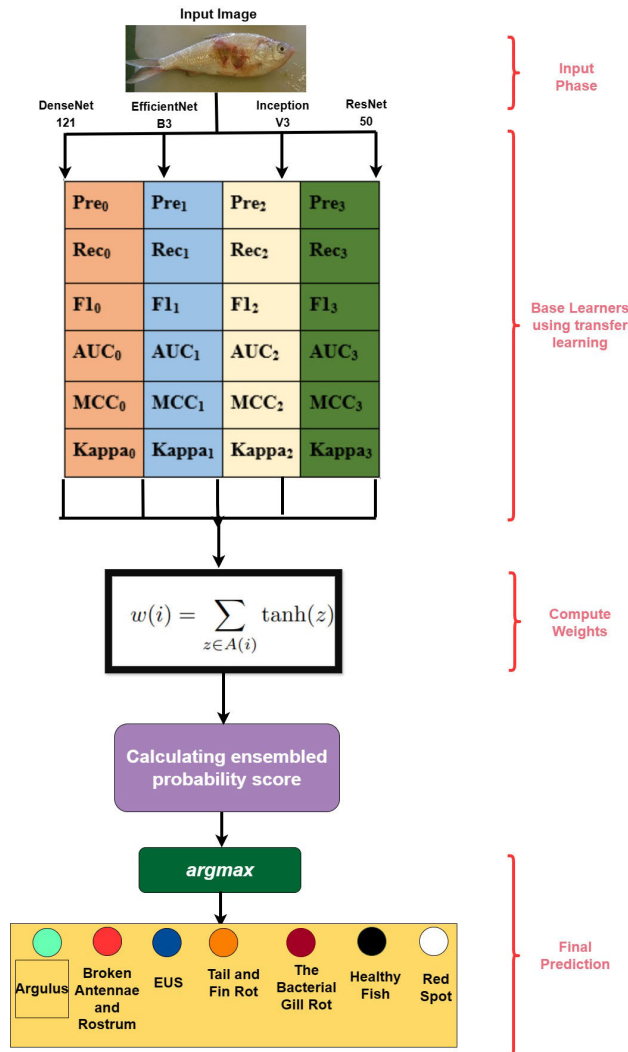


FIGURE 3. Proposed novel Performance Metric-Infused Weighted Ensemble (PMIWE) model.

pandas make data manipulation and analysis easy. These are important tasks for preprocessing and organizing datasets. When combined, these tools provide a complete toolkit that makes it possible to create advanced models for the detection of fish diseases in the Google Colab Pro+ environment.

**B. DATASET SPLIT**

By using the augmentation process on the original dataset, we significantly enhanced the dataset. By using this method, we can make sure that our model is exposed to a greater variety of data variations. After that, we organized the resultant enhanced dataset, containing both the original and the augmented images, into three distinct folders: training, testing, and validation. By doing this, we can make sure that our model is trained, tested, and validated on a variety of data samples, which will increase its resilience and capacity for generalization. We have 1,616 images for training, 178 for validation, and 203 for testing. Table 3 shows the number of training, validation, and testing samples of each class.

**Algorithm 1** Performance Metric-Infused Weighted Ensemble (PMIWE) Model

**Input:** Training dataset with features and labels, Test dataset with features

**Output:** Predicted classes for the samples in the test dataset

**Base Model Creation:**

**for** each base learner *i* **do**

Train base learner *i* on the training dataset.

Generate probability scores  $p(i)_j$  for each sample *j* in the training dataset.

Evaluate base learner *i* on the test dataset.

Compute predictions  $y^i_{pred}$  from base learner *i*.

Compute performance metrics using  $y^i_{pred}$  and true labels.

Calculate the weight  $W^i$  for each classifier using the hyperbolic tangent function:

$$W^i = \sum_{z \in A(i)} \tanh(z) \text{ based on performance metrics } prc^i, rec^i, F1^i, MCC^i, \text{ and } kappa^i.$$

**end for**

**Ensemble Model Creation:**

Assign weights to each base learner based on  $W(i)$

Create the ensemble model using the weighted base learners.

Train the ensemble model on the training dataset.

**Prediction Using Ensemble Model:**

**for** each sample *j* in the test dataset **do**

Compute the probability scores for each class using the base learners' probability arrays:

$$p(i)_j = [p(i)_{j,1}, p(i)_{j,2}, \dots, p(i)_{j,C}]$$

Calculate the weighted ensemble probability for each class:

$$enp_j = [\frac{\sum_{(i)} W(i)*p(i)_{j,1}}{\sum_{(i)} W(i)}, \dots, \frac{\sum_{(i)} W(i)*p(i)_{j,C}}{\sum_{(i)} W(i)}]$$

Predict the class with the highest probability:

$$Prediction = \text{argmax}(ensemble_p)$$

**end for**

TABLE 3. Class distribution in training, validation, and testing datasets.

Class	Number of Images		
	Training	Validation	Testing
Tail and Fin Rot	235	27	25
Bacterial Gill Rot	235	27	31
Red Spot	206	16	22
Healthy fish	235	27	34
EUS	235	27	33
Broken antennae and rostrum	235	27	29
Argulus	235	27	29

**C. HYPERPARAMETERS**

- **Learning Rate:** One hyperparameter that regulates the step size or the speed at which the model weights are updated during training is the learning rate. It is an important parameter since it has an impact on



the optimization and convergence processes in finite time [53], [54]. A low learning rate could result in an excessively slow training process, whereas a high learning rate could cause the model to exceed the minimum. Sometimes, a model's accuracy is dependent on its learning rate settings.

In symbolic terms, the gradient descent updating rule for a parameter  $w$  with learning rate  $\eta$  is as follows:

$$w = w - \eta \cdot \nabla L(w)$$

Here, the model parameter is denoted by  $w$ . The rate of learning is  $\eta$ . With regard to the parameter  $w$ , the gradient of the loss function  $L(w)$  is represented as  $\nabla L(w)$ .

Stable updates to the model parameters are ensured by a lower learning rate, such as 0.0001, which lowers the possibility of overshooting the loss function minimum, an issue that can arise with higher learning rates. In order to achieve a smooth and consistent convergence, this stability is essential. Furthermore, fine-tuning parameters at a slower learning rate is crucial for accurately identifying the complex patterns linked to fish illnesses. Extensive empirical testing was used to validate this decision, and the results showed that 0.0001 offered the best trade-off between training time and model performance. While lower learning rates needlessly prolonged training without producing appreciable performance gains, higher learning rates caused instability. This decision is further supported by the Adam optimizer [55], which is well-known for working well at lower learning rates [56] and for enhancing training dynamics and model accuracy. In order to ensure efficient and accurate model training, stability, precision, and empirical validation were therefore factors in the decision to choose a 0.0001 learning rate. In fact, in the experiments, we also tried out a higher learning rate of 0.001. But, 0.0001 indeed offers better results.

- **Loss Function:** During training, the loss function calculates the degree to which the model's predictions agree with the target values. It measures the accuracy or accuracy of the model's predictions by quantifying the error between the predicted and actual values.

The loss function can be represented symbolically as  $L(y, \hat{y})$ , where  $y$  stands for the true target values and  $\hat{y}$  for the forecast values. Depending on the problem being solved, the loss function's specific form (for example, mean squared error in regression or cross-entropy in classification) will vary. The objective of the training is to reduce the loss function.

The categorical cross-entropy loss function, which is commonly used for multi-class classification, is defined as:

$$L(y, \hat{y}) = - \sum_{i=1}^k y_i \cdot \log(\hat{y}_i)$$

where  $k$  is the number of possible classes ( $k = 7$  in our case),  $y_i \in \{0, 1\}$  is the actual class label for the class  $i$ , and  $\hat{y}_i \in \{\mathbb{R} | 0 \leq \hat{y}_i \leq 1\}$  is the predicted probability (a.k.a. confidence) for the class  $i$ . (Notes: among all the  $k$  classes, only one  $y_i$  is 1, and the remaining are all 0. The sum of the  $\hat{y}_i$  values for all the  $k$  classes is 1.)

- **Optimizer:** To reduce the loss function during model training, optimizers are crucial algorithms used in machine learning. To improve model performance, they iteratively modify the model's parameters by the gradients of the loss function. Optimizers are essential because of their effectiveness in reducing loss, promoting quicker convergence, and adjusting to different optimization issues like sparse data or non-convex loss surfaces. Moreover, they allow practitioners to fine-tune optimization algorithms for particular workloads and datasets by providing robustness and customization choices through hyperparameter adjustment. Achieving dependable and consistent performance depends heavily on how robust the optimizer is that we employ in our model. We selected the Adam optimizer [55] because of its adaptive learning rate and track record of success with noisy and sparse gradients [57], [58]. Based on the first and second moments of gradients, Adam modifies the learning rates for each parameter, resulting in faster convergence and a lower chance of becoming trapped in local minima. Its capacity to manage sparse gradients is beneficial for our model, which processes a variety of features taken from photos of fish illnesses. Adam also uses bias correction to improve stability during training by mitigating initialization bias in moment estimates. The optimizer is appropriate for larger models and datasets due to its computational efficiency and lower memory requirements when compared to other optimizers, guaranteeing real-world application in the aquaculture sector. The ideal configuration was further guaranteed by extensive hyperparameter tuning, which struck a balance between model stability and convergence speed. Overall, Adam's resilience in learning dynamics greatly enhances the model's efficacy and dependability in practical applications.

Table 4 contains a list of the suggested model's important hyperparameters, which represent the estimated computing dimension of a particular neural network. It should be noted that the suggested design uses Softmax activation functions for the output layer and Rectified Linear Unit (ReLU) for the hidden layer activation.

#### D. PERFORMANCE EVALUATION METRICS

All of the widely used performance evaluation metrics [59], namely, accuracy, precision, recall, F1-score, and AUC-ROC, are used in this evaluation to support the model's high performance. These metrics are defined below.

Confusion matrices are produced in order to analyze the error. A confusion matrix is a table that shows the counts of TP, TN, FP, and FN predictions for each class, thereby

**TABLE 4.** Experiment settings.

Parameter	Value
Learning Rate	0.001, 0.0001
Optimizer	Adam
Loss Function	CategoricalCrossentropy
Activation (Hidden)	Relu
Activation (Output)	Softmax
Epochs	50
Verbose	2
Multiprocessing	True
Base Model	Pre-trained
Classes	7
IMG_SIZE	299*299
Batch Size	128

summarizing the performance of a classification algorithm. Here, FN stands for false negative, FP for false positive, TP for true positive, and TN for true negative. When an instance (fish image in our case) is predicted by the classifier as belonging to the class of interest, it is defined as a “positive”. When an instance is predicted as belonging to the other classes, it is regarded as a “negative”. We take each of the seven classes as the class of interest at a time and calculate the seven sets of TP, TN, FP, and FN values. From each set of values, we can calculate the following performance metrics.

The percentage of correctly classified instances relative to all instances is known as accuracy. It is used to assess the overall classification performance. Accuracy is computed as shown in Eq. 18.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (18)$$

Precision is the percentage of correctly predicted positive instances among all positive predictions. It is computed as shown in Eq. 19.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (19)$$

Recall (a.k.a. TP rate) is a metric that quantifies the percentage of accurately predicted positive instances among all actual positive instances. It is also referred to as sensitivity or true positive rate. It is computed as shown in Eq. 20.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (20)$$

The harmonic mean of recall and precision yields the F1-score, also called the F-score or F-measure, which is a balanced metric between the two, and it is calculated as shown in Eq. 21.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (21)$$

Plotting the TP rate (TP/(TP+FN)) against the FP rate (FP/(FP+TN)) at different threshold settings, the ROC-AUC calculates the area under the curve (AUC) of the receiver operating characteristic (ROC) curve [60]. The AUC-ROC is computed by integrating the area under the ROC curve.

Since our dataset is not balanced, we also used Kappa and MCC to calculate the agreement. The Kappa coefficient [61] is an appropriate measure of the reliability of a classifier. It is regarded as a robust measure, as the possibility of the correct classification by chance is taken into account. It quantifies the degree of agreement between observed and predicted classifications by taking into account true and false positives as well as negatives. It is defined in Eq. 22.

$$\text{kappa} = \frac{2 \times (\text{TP} \times \text{TN} - \text{FN} \times \text{FP})}{(\text{TP} + \text{FP}) \times (\text{FP} + \text{TN}) + (\text{TP} + \text{FN}) \times (\text{FN} + \text{TN})} \quad (22)$$

The Matthews correlation coefficient (MCC) [62] evaluates the accuracy of binary classifications for unbalanced data. It also takes all of TP, TN, FP, and FN into account and is regarded as a balanced measure that can be used even if the classes are of different sizes. MCC runs from 1 (perfect agreement) to  $-1$  (complete disagreement). It is calculated using Eq. 23.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{FP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (23)$$

## VI. RESULT AND ANALYSIS

We calculate the average values from the seven sets of performance metrics described above in Section V-D (one set for each class).

### A. MODEL GENERALIZATION ANALYSIS

Table 5 displays the performance of the base pre-trained models and ensemble models across training, validation, and testing phases.

**TABLE 5.** Training, validation, and testing accuracies of the base pre-trained models and the proposed ensemble models (AE = Averaged Ensemble; PMIWE = Performance Metric-Infused Weighted Ensemble; the subscripts 0.001 and 0.0001 are the learning rates).

Algorithms	Training	Validation	Testing
DenseNet-121	76.31%	87.52%	78.89%
EfficientNetB3	99.51%	96.51%	80.90%
InceptionV3	98.57%	96.51%	83.91%
ResNet-50	98.46%	97.63%	82.91%
AE <sub>0.001</sub>	98.10%	89.61%	84.54%
AE <sub>0.0001</sub>	99.27%	93.74%	87.11%
PMIWE <sub>0.001</sub>	98.65%	92.23%	86.65%
PMIWE <sub>0.0001</sub>	97.70%	97.54%	97.53%

From Table 2, we can observe that DenseNet-121 exhibits the smallest model size of 28.57 MB and a moderate number of total parameters of 7,046,239, where 7,038,569 parameters are trainable, which made the model adaptable. Despite its faster training time of 700 sec., however, its lower training accuracy of 76.31% indicated slight underfitting where the validation and testing accuracy is much higher than that. It achieved 87.52% validation accuracy and 78.89% testing accuracy.

Despite having the highest training accuracy of 99.51%, EfficientNetB3 suffers from overfitting. It happened because the validation dropped to 96.51% with the testing accuracy of 80.90%. It has a model size of 41.8 MB and more total parameters of 10,704,294 than DenseNet-121, with a significant number of trainable parameters of 7,256,895. It has the largest number of trainable parameters.

With a model size of 83.23 MB and a high number of parameters of 21,817,127, InceptionV3 has a considerable number of non-trainable parameters of 1,967,424. Here, the training accuracy for InceptionV3 is 98.57%, the validation accuracy is 96.51%, and the testing accuracy is 83.91%.

On the other hand, ResNet-50 also showed a similar trend to InceptionV3, where it has the largest model size at 90.3 MB and the greatest number of total parameters of 23,620,255. However, only 14,343 parameters are trainable, which indicates that most of the network's parameters are frozen, which might help in leveraging pre-trained features. The validation accuracy is much lower than the training accuracy, which suggests a slightly overfitting situation. The training accuracy for ResNet-50 is 98.46%, while the validation accuracy is 97.63%, with a testing accuracy of 82.91%. Pre-trained models have shown promising results in reducing overfitting to some degree.

Through modification of these models, a large fraction of parameters were made non-trainable, thereby reducing computational complexity. However, another key goal of our research is to determine how the model can perform on unseen data. At first, using customized transfer learning models allowed for a decrease in processing costs. Afterward, using ensemble methods produced impressive increases in test accuracy, which satisfied the main objective of our study. As mentioned above, We have applied the AE and PMIWE ensemble models.

The PMIWE with 0.001 learning rate obtained 98.65% training accuracy; the validation accuracy was 92.23%, and the testing accuracy was 86.65%, indicating that the ensemble model overfitted but performed well on unseen data compared to the pre-trained models. Additionally, the training accuracy for the AE model with a learning rate of 0.001 was 98.10%, while the validation and testing accuracy was 89.61% and 84.54%, respectively. Although the testing accuracy is increased compared to pre-trained models in some situations, the overfitting issue is still not resolved.

Furthermore, when the learning rate is applied to 0.0001, a significant change was observed among the ensemble models. For a 0.0001 learning rate, PMIWE acquired 97.70% training accuracy with an almost similar validation accuracy of 97.54%. Here, the training and validation accuracy is quite the same with 97.53%, so this PMIWE model is well-fitted. On the other hand, the AE model with a learning rate of 0.0001 performed reasonably well but still with some overfitting (training 99.27%, validation 93.74%, and testing 87.11%).

In our research, the PMIWE model performed best in classifying fish diseases, and the proposed model was

well-fitted. The testing accuracy is 97.53%. First of all, by giving each base model a distinct weight for its prediction based on multiple performance metrics, the PMIWE model leverages the advantages of multiple base models.

Large datasets are used to train pre-trained models, which enable them to pick up general features and patterns. These pre-trained models can be made more adaptive and knowledgeable about the available data by developing them on particular tasks or datasets. In fact, combining these improved, pre-trained models into an ensemble enables the utilization of the various advantages of various architectures and training approaches. Because of this diversity, the ensemble is better able to predict outcomes because it can draw from the combined knowledge and experience of several models.

Fish disease classification performance was enhanced with a lower learning rate of 0.0001, which allowed for more accurate updates to the model's parameters, finer adjustments, and more effective convergence. Moreover, ensemble approaches reduce the chance of any one model memorizing noise or unimportant patterns in the data by combining predictions from several models, which naturally reduces the risk of overfitting. This ensemble effect improved the overall model's capacity for generalization, which improved our fish disease classification's performance on unseen data.

In Figure 4, we present our model's performance metrics over 50 epochs. A pattern becomes apparent during training: training and validation losses both go down gradually, indicating that the model is capable of learning. Crucially, we see a steady increase in training and validation accuracies, indicating that the model successfully generalizes to new data without displaying overfitting during the whole training period. Once more, the graphical representations demonstrated the robustness of the model.

## B. PERFORMANCE ANALYSIS

We have thoroughly examined our model's behavior regarding overfitting and generalization to previously unseen data. Our analysis produced a strong model with few overfitting indicators, indicating that it can generalize well outside of the training set. To extensively evaluate the model's performance, we also used a wide range of assessment criteria. These standards gave us detailed information about different aspects of its efficacy and guaranteed a comprehensive comprehension of its potential. Table 6 presents the performance of the models evaluated by accuracy, precision, recall, and F1-Score. The same data are also displayed in a graphical format in Figure 5.

Firstly, DenseNet-121 displays reasonable recall and precision rates of 78% and 79%, respectively. But EfficientNetB3 outperforms DenseNet-121 with a higher recall of 81% and a precision of 80%, demonstrating its better capacity to recognize pertinent instances with negligible precision loss. Both models have an F1-score of 78% and 80%. On the other hand, InceptionV3 places more emphasis on accuracy, outperforming DenseNet-121 and EfficientNetB3

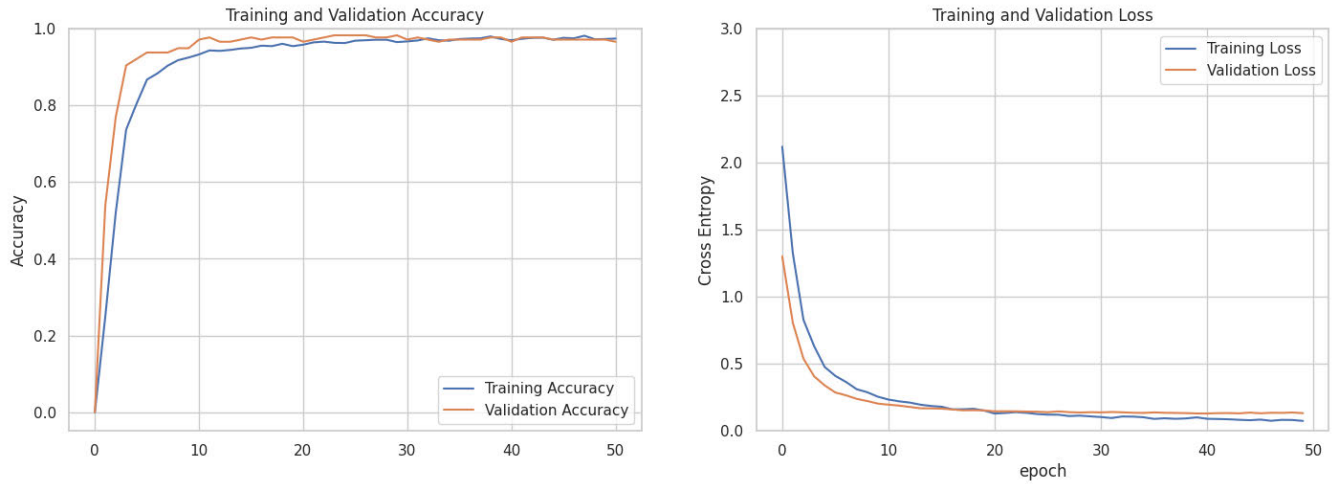


FIGURE 4. Graphical representation of training and validation accuracy, training and validation loss of every epoch (PMIWE<sub>0.001</sub>).

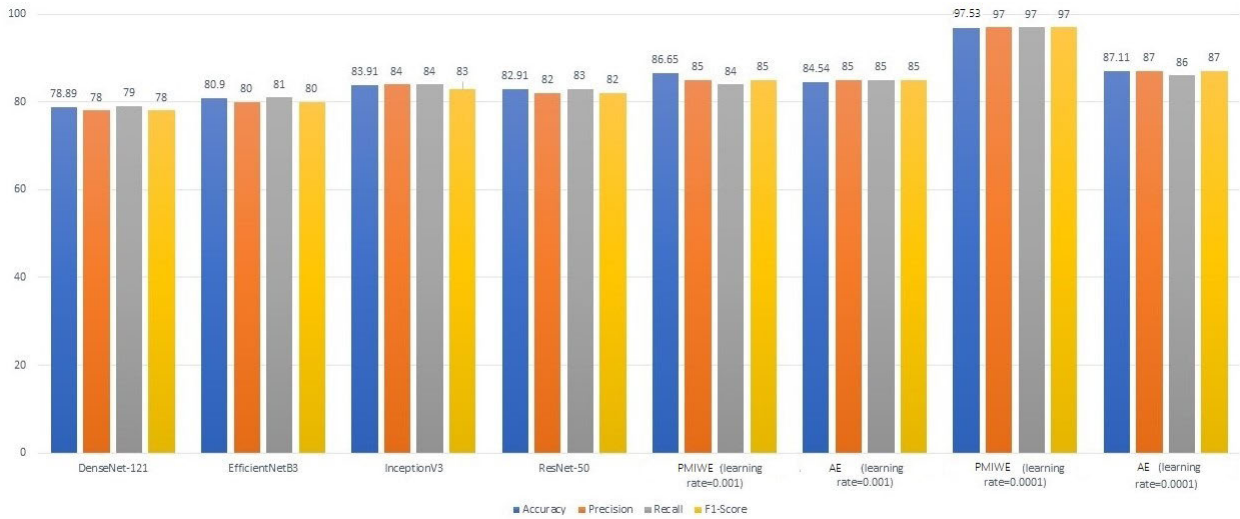


FIGURE 5. Performance of all models.

TABLE 6. Average values of accuracy, precision, recall, and F1-score of the base pre-trained models and the proposed ensemble models on the testing dataset.

Algorithms	Accuracy	Precision	Recall	F1-Score
DenseNet-121	78.89%	78%	79%	78%
EfficientNetB3	80.90%	80%	81%	80%
InceptionV3	83.91%	84%	84%	83%
ResNet-50	82.91%	82%	83%	82%
AE <sub>0.001</sub>	84.54%	85%	85%	85%
AE <sub>0.0001</sub>	87.11%	87%	86%	87%
PMIWE <sub>0.001</sub>	86.65%	85%	84%	85%
PMIWE <sub>0.0001</sub>	97.53%	97%	97%	97%

with an 84% rate. This preference for accuracy points to InceptionV3’s methodical identification process, making it an appealing option. InceptionV3 performed well on unseen data and had fewer overfitting problems than any pre-trained model. Additionally, the F1-score of 83% indicates the

identification of positive and negative instances better than any pre-trained models. Lastly, ResNet-50 outperforms DenseNet-121 and EfficientNetB3 with a recall rate of 83%. The F1-score and precision are 82%.

The ensemble model that we used in our study to improve detection performance has already proven to be the best. Additionally, we discovered that they had significantly better identification rates for false positive and negative cases and also true positive and negative cases than the pre-trained models. When various ensemble methods with different learning rates are compared, distinct patterns in precision, recall, and F1-score metrics are revealed. At a learning rate of 0.001, PMIWE consistently achieves 85%, 84%, and 85% for precision, recall, and F1-score, respectively. Comparably, the AE model with the same learning rate performs steadily, keeping recall and precision at 85% apiece, yielding an F1-score of 85%. As an illustration, PMIWE which uses a



lower learning rate of 0.0001, achieves remarkable results in terms of precision, recall, and F1-score, with values of 97%, 97%, and 97%, respectively, across all metrics. A lower learning rate in the AE model results in better recall and precision (87% and 86%, respectively), which contributes to an F1 score of 87%. The significant improvement in performance metrics, especially when the PMIWE model is used at a lower learning rate, indicates that the ensemble makes good use of the various advantages that each model has to offer. The more careful adjustments made possible by the lower learning rate lead to improved generalization and finer pattern learning, which in turn produce higher recall, F1-score values, and precision. Consequently, PMIWE, with a learning rate of 0.0001, comes out on top, demonstrating its robustness and effectiveness in the classification of fish diseases, where we have previously found that it is well-fitted and performs best on unseen data. As evidenced from the analysis, the PMIWE model with a learning rate of 0.0001 performed best among all models to classify the fish diseases; we have presented the class-wise performance of this model in Table 7.

**TABLE 7. Classification report of our PMIWE<sub>0.0001</sub> model (class-wise).**

Class	Precision	Recall	F1-Score
Argulus	0.97	0.97	0.97
Broken antennae and rostrum	1.00	1.00	1.00
EUS	0.94	0.97	0.95
Healthy Fish	1.00	1.00	1.00
Red Spot	0.91	0.87	0.89
Bacterial Gill Rot	1.00	1.00	1.00
Tail And Fin Rot	1.00	1.00	1.00

### C. AGREEMENT ANALYSIS

Agreement analysis metrics go deeper than precision, recall, and F1-score in providing insightful views into the model's capacity to accurately classify instances and strike a balance between positive and negative instances. The overall agreement of the model with the ground truth is revealed by Cohen's Kappa coefficient, which takes into consideration the agreement between true and predicted classifications that go beyond chance. In contrast, the correlation between true and predicted classifications—which takes into account both true and false positives and negatives—is captured by the Matthews correlation coefficient. A more thorough understanding of the model's performance is obtained by integrating these metrics into the evaluation process. This includes not only the model's ability to achieve high precision, recall, and F1-score but also its ability to maintain consistent agreement and correlation with the true classifications across all classes. With the highest Cohen's Kappa coefficient of 0.8106 among the individual models, InceptionV3 stands out as indicating a substantial agreement that goes beyond chance. It also has the highest MCC (0.8118), indicating a high degree of agreement between the true and predicted classifications. Closely behind, EfficientNetB3 and ResNet-50 also show strong metrics for agreement, with MCC

values of 0.7740 and 0.7986 and Cohen's Kappa values of 0.7755 and 0.7989, respectively. The performance of DenseNet-121 is also impressive, if marginally lower, with Cohen's Kappa and MCC coefficients of 0.7520 and 0.7508, respectively. Turning now to ensemble techniques, Cohen's Kappa coefficients of 0.8341 and 0.8242, respectively, show improved agreement metrics for the PMIWE and AE models with a learning rate of 0.001 when compared to individual models. Similarly, stronger correlations between the true and predicted classifications are reflected in their MCC values of 0.8217 and 0.8357, respectively. All other models are significantly outperformed by the PMIWE model with a lower learning rate of 0.0001, which results in an MCC of 0.9562 and a Cohen's Kappa coefficient of 0.9523. The outcomes prove the ensemble's ability to capture intricate relationships in the multi-class dataset, leading to better predictive performance.

**TABLE 8. Kappa and MCC values of the models.**

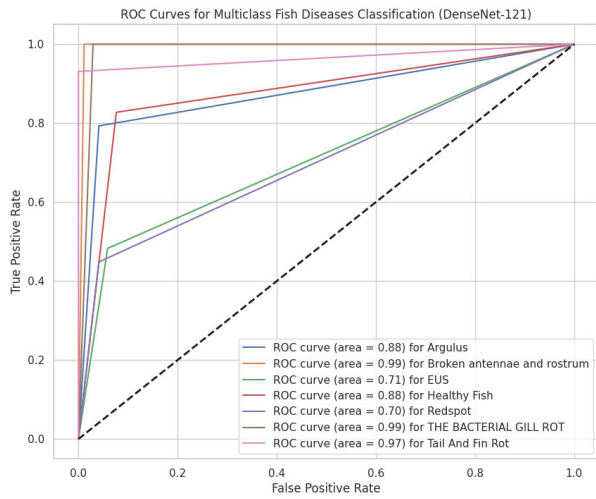
Model	Kappa	MCC
DenseNet-121	0.7520	0.7508
EfficientNetB3	0.7755	0.7740
InceptionV3	0.8106	0.8118
ResNet-50	0.7989	0.7986
AE <sub>0.001</sub>	0.8242	0.8217
AE <sub>0.0001</sub>	0.8476	0.8455
PMIWE <sub>0.001</sub>	0.8341	0.8357
PMIWE <sub>0.0001</sub>	0.9523	0.9562

### D. ERROR AND GRAPHICAL ANALYSIS

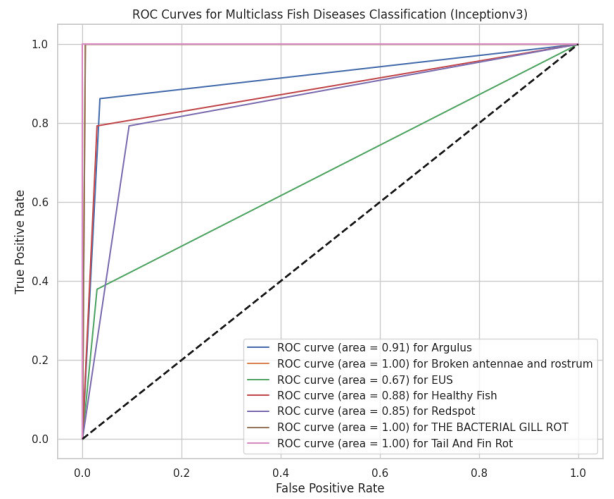
A ROC curve is a graphical plot of the true positive rate versus the false positive rate at various thresholds or a linear function of multiple other parameters being rates. For every cut-off point or method of calculating probabilities, a pair of sensitivity and specificity are obtained. ROC curves analysis for multi-class Fish disease classification with the four transfer learning models and Ensemble Models with seven different Fish Disease classes are demonstrated in Figure 6.

We calculated all class's Area Under Curve (AUC). For ResNet-50, it exhibits commendable performance across most classes. The class 'Argulus' has an AUC of 0.93, which means the model performs pretty well in identifying cases of this particular disease against other conditions. The class 'Broken Antennae and Rostrum' incredibly exhibits an AUC of 0.99, with most classes having near-perfect classification of the class. This is vice-versa for 'EUS,' which has a moderate AUC of 0.72. It also worked well for 'Healthy Fish,' whereby the AUC is 0.86, but slightly drops to 0.78 for 'Red Spot,' hence the possibility of some areas needing some model improvement. At the top of the list, both 'Bacterial Gill Rot' and 'Tail and Fin Rot' classes yield a perfect score of AUC at 1.00, therefore signifying that the model has an even higher classification accuracy for such classes.

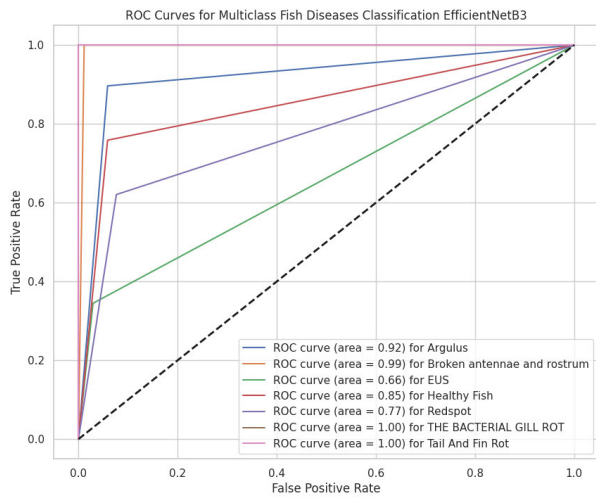
The InceptionV3 model has high discriminative abilities, scoring an AUC of 0.91 for 'Argulus'. The model does



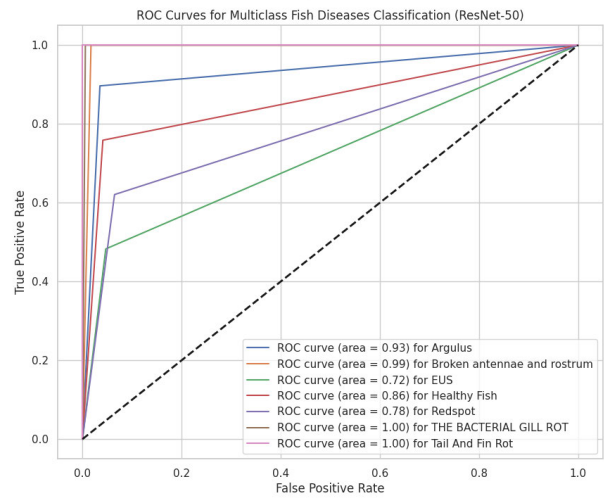
(a) DenseNet-121



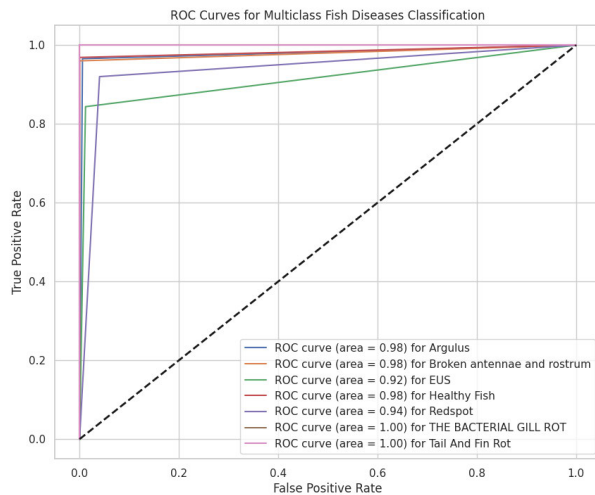
(c) InceptionV3



(b) EfficientnetB3



(d) ResNet-50



(e) PMIWE<sub>0.0001</sub>

FIGURE 6. ROC curve (AUC value) of the pre-trained models and proposed (PMIWE<sub>0.0001</sub>).

a perfect job distinguishing the ‘Broken Antennae and Rostrum’, having achieved an AUC of 1.00. On the other hand, the class ‘EUS’ showed less effective discrimination; that is, the value of AUC is 0.67. Competent discrimination was observed among the class ‘Healthy Fish’ and class ‘Red Spot,’ where the AUC was found to be 0.88 and 0.85, respectively. More specifically, the InceptionV3 model, like the ResNet-50 model, also displayed perfect discrimination for the classes ‘Bacterial Gill Rot’ and ‘Tail And Fin Rot,’ with both having an AUC of 1.00, hence rendering the strength of this model to identify these diseases as very high.

The ROC curve of the EfficientNetB3 model displayed very strong predictive performance, where the AUC for ‘Argulus’ is 0.92. The ‘Broken Antennae and Rostrum’ class demonstrates performance almost at the perfect level: AUC = 0.99. However, ‘EUS’ appeared to be a challenging condition for the model to discern accurately, with a lower AUC of 0.66. ‘Healthy Fish’ performs admirably even with an AUC of 0.85, while ‘Red Spot’ can only come up to a modest AUC of 0.77, meaning it has more space for performance. The model can score the perfect AUC of 1.00 for the class of ‘Bacterial Gill Rot’ and another perfect score of 1.00 for the class ‘Tail And Fin Rot,’ demonstrating a great ability to classify.

For the DenseNet-121 model, the ROC curve revealed the good discriminatory power of the class ‘Argulus,’ with an AUC value of 0.88. On the other hand, the ‘Broken Antennae and Rostrum’ class has an AUC value of 0.99, while the ‘EUS’ class has a reasonably good AUC value of 0.71. The model can differentiate ‘Healthy Fish’ with a high level of reliability, as evidenced by the computed AUC score of 0.88. On the other hand, the class of ‘Red Spot’ is not well discriminated, whereby the AUC equals 0.70. The ‘Bacterial Gill Rot’ and ‘Tail and Fin Rot’ classes had an AUC of 0.99 and 0.97, respectively, which points out very good predictive power at discriminating by these diseases.

Pre-trained models performed admirably in several classes, but they faced considerable difficulties, especially in the ‘EUS’ class, where all models had very low AUC values. However, the PMIWE model with a learning rate of 0.0001 performed much better as compared to the pre-trained models with the highest AUC. The result for the ‘EUS’ class was 0.92 AUC, and this outperformed the AUC that the individual pre-trained models could attain at 0.72. Notably, the AUC values for classes like ‘Bacterial Gill Rot’ and ‘Tail and Fin Rot’ reached 1.00, indicating accurate classification without any cases of misclassification. Despite facing challenges in the ‘Red Spot’ class, the suggested model’s effectiveness was further demonstrated by its respectable AUC score of 0.94. This outstanding performance points out how ensemble models can enable accurate and dependable fish disease detection, which is essential for preserving fish health in aquaculture operations.

The proposed model is evaluated using a variety of criteria, such as analyzing the overfitting or underfitting,

discussing the classification reports as well as the agreement analysis. Based on high performance (ref. Table 5), it is observed that the PMIWE model with a learning rate of 0.0001 is the best-performing ensemble model for fish disease classification. A thorough error analysis is carried out to acquire a further understanding of the effectiveness of the suggested approach.

Figure 7 presents the confusion matrix of the best ensemble model, and based on this, the quantitative analysis is performed. Table 3 shows the number of test samples of each class. The class ‘Tail and Fin Rot’ contains 25 test samples. The confusion matrix shows that the proposed model does not incur any errors in classifying this class. Among the 25 samples, it can accurately classify all samples. The ‘Bacterial Gill Rot’ class consists of 31 test samples, and among them, all are predicted accurately. Our model made some errors in classifying the ‘Red Spot’ class. Among the 22 ‘Red Spot’ instances, the model classified 1 instance as ‘EUS’ and one instance as ‘Argulus.’ The same situation is observed in the ‘EUS’ class, where 2 samples are predicted to be ‘Red Spot.’ Among the 33 test instances, it has predicted 31 instances correctly. The model classified all the test instances correctly for ‘Healthy Fish.’ The same scenario happened for the Broken antennae and rostrum class, where all samples were predicted without making any errors. Finally, among the 29 ‘Argulus’ instances, the model predicted 1 sample to be a ‘Red Spot.’ Among the total 203 test instances our PWIME model classifier 198 samples accurately among all classes.

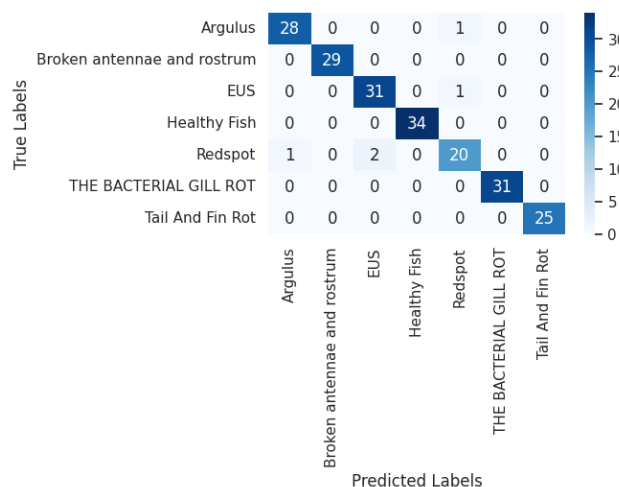
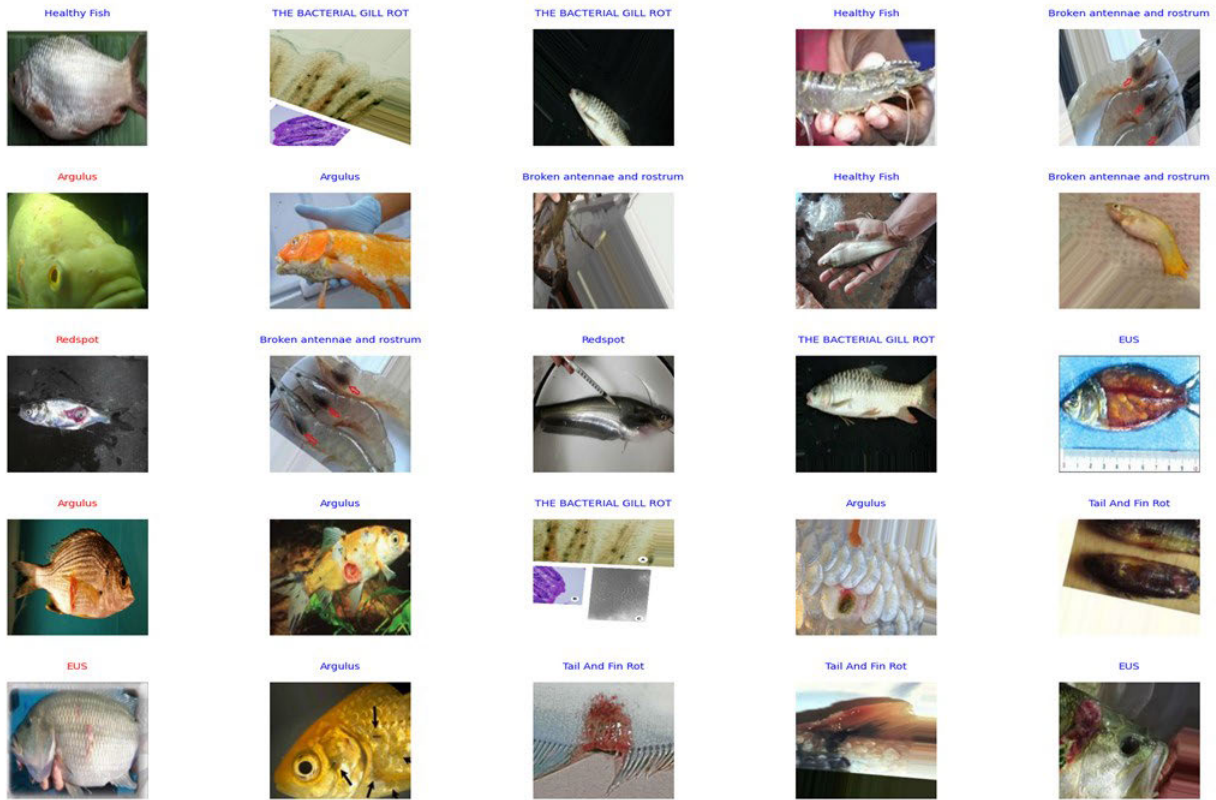
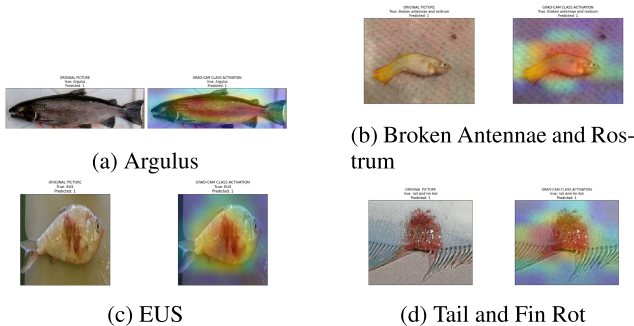


FIGURE 7. Confusion matrix of the proposed weighted ensemble model (PMIWE<sub>0.0001</sub>).

Figure 8 presented some sample output produced by our model PMIWE<sub>0.0001</sub>. A total of 25 samples are presented here. Among the model accurately classified the 21 samples and 4 samples were misclassified. The correct classification is blue, whereas red is marked as the wrong classification.



**FIGURE 8.** Sample output of the proposed model. Here, the correct classification is marked in blue, and the wrong classification is marked in red (PMIWE<sub>0.0001</sub>).



**FIGURE 9.** Gradient-weighted class activation map (Grad-CAM) decision visualization of fish disease detection. The original image is on the left side, and the Grad-CAM-generated image is on the right side.

**E. GRAD-CAM VISUALIZATION ACROSS FOUR BASE LEARNERS FOR ENHANCED INTERPRETABILITY IN FISH DISEASE**

We used gradient-weighted class activation mapping (Grad-CAM) in this study to interpret the decision-making process of four pre-trained advanced models, namely, DenseNet-121, EfficientNetB3, InceptionV3, and ResNet-50, applied for the classification of images into seven distinct classes of fish diseases. GradCAM, a technique that generates visual explanations for decisions made by convolutional neural networks, offers a window into understanding which features

within an image contribute most significantly to a model’s classification output. This method is of particular interest in areas requiring a high level of interpretation, such as the detection and diagnosis of fish diseases in aquaculture.

To determine how much each highlight outline feature map (FM) contributes to the prediction, CAM determines how many weights each FM has depending on the final convolutional layer. The final feature map ( $C_{ij}^k$ ) and the projected value  $\hat{y}$  are connected via a linear association. The linear layers consist of fully connected layers (FCLs) and global average pooling (GAP). While FCLs, with weights  $w_{gk}$ , generate an output specified by Eq. 24, where  $C^k$  displays the visualization of the  $k$ th feature map (FM), GAP produces the output  $A^k = C_{ij}^k$ .

$$y_g = \sum_k w_{gk} A^k = \sum_k w_{gk} C_{ij}^k = \sum_{k,i,j} w_{gk} C_{ij}^k \quad (24)$$

Consequently, we employ GradCAM as global average FM gradients as weights rather than grouping them. Rather than being categorized in the usual way, special class weights are gathered from the last convolutional layer using the Global Average FM Gradients (GAG). Equation 24, in which  $P$  denotes the number of pixels in an FM,  $g$  denotes the class



gradient, and  $C_{ij}^k$  represents the  $k$ -th FM's value.

$$a_{gk} = \frac{1}{P} \sum_{i,j} \frac{\partial y_g}{\partial C_{ij}^k} \quad (25)$$

GradCAM's main feature is its capacity to use gradients of the target idea flowing into CNN's final convolutional layer to emphasize the portions of an image that are crucial for class label prediction. For each class  $c$ , the GradCAM localization map  $L_{\text{Grad-CAM}}^c$  can be expressed as follows:

$$L_{\text{Grad-CAM}}^c = \text{softmax} \left( \sum_k \alpha_k^c A^k \right) \quad (26)$$

where  $\alpha_k^c$  indicates the significance of these activations towards the class  $c$ , computed as the gradient of the class score concerning the feature map activations, averaged over all spatial locations.  $A^k$  represents the feature activations of the  $k^{\text{th}}$  channel of the final convolutional layer. By using the softmax function, it is made sure that only the features that positively impact the class prediction are displayed.

Utilizing this technique, we analyzed the activation patterns induced by different disease manifestations in fish, with each of the four models assigned to a specific disease condition for visualization purposes. The DenseNet-121 model's GradCAM output for detecting *Argulus* showcased a concentrated focus on the body surface areas, pinpointing the exact locations where *Argulus* parasites typically manifest. This precision in localization reflects the model's capability to discern subtle textural and morphological cues indicative of the disease.

Similarly, the EfficientNetB3 model generated heatmaps highlighting the areas of the head and antennae most damaged by such physical damage when applied to photos of broken rostrums and antennae. This focus on specificity in the model's attention highlights how good it is at spotting structural irregularities, which is an important part of evaluating physical injury in fish.

The GradCAM representations emphasized ulcerated regions and lesions, which exactly aligned with the clinical signs of EUS, for the identification of EUS using the InceptionV3 model. The model's ability to identify disease-specific textural alterations is validated by these visual explanations, supporting the model's diagnostic accuracy.

Several previously trained base learners are combined in the proposed ensemble model. Every base learner possesses a unique set of abilities and can recognize significant regions in the input pictures. Our algorithm can identify and highlight significant regions or characteristics in the photos by using Grad-CAM representations, which makes the decision-making process easier. The ensemble model successfully synthesizes a variety of insights by combining the predictions of these base learners, which improves its predictive ability. As shown in Figure 9, the ensemble model makes predictions by carefully selecting regions in

images that have higher red density, as identified by Grad-CAM analyses. The analysis shows that the model highlights the place where the disease symptoms are presented. This thorough approach ultimately advances the model's applicability across multiple domains by fostering robustness and reliability in the predictions in addition to improving interpretability.

## F. COMPARATIVE ANALYSIS

The results' analysis showed that the ensemble method, weighted, was the most successful model for classifying fish diseases. We also compared the performance of the suggested model to that of other methods, with many perspectives, to assess our research contribution and model efficacy. We have presented some comparisons with prior research in Table 9. (Note: The comparisons are merely indicative because different datasets and/or different experimental settings were used.)

While many studies have concentrated on binary classification tasks, our study tackled a more intricate multi-class scenario and achieved an impressive accuracy of 97.53%. As an example, using CNN with attention mechanisms produced an accuracy of 94.28% in a noteworthy study by Huang et al. [2], but at a significantly higher computational cost, estimated at roughly 51,639,400 trainable parameters. In addition to surpassing this benchmark, our model had a total of 12,309,710 trainable parameters and significantly less computational complexity. We also compared our findings with those of other researchers who used baseline classifiers like SVM and RF. While Ahmed et al. [16] and Manik et al. [18] obtained accuracies of 94.12% and 91.42%, respectively, with SVM classifiers, Mia et al. [13] reported an accuracy of 88.87% with the RF classifier. Our results showed a significant improvement in accuracy over these approaches, confirming the superiority of our suggested ensemble model.

It can be observed that our comparative study highlights the novelty and effectiveness of our suggested ensemble model in the field of fish disease classification. It is not only more accurate than previous approaches, but it also provides a computationally efficient solution. These results represent a significant contribution to the field and open new avenues for research on disease classification. In another work [22], the single transfer learning model produced 59.33% accuracy. They have applied ResNet-50, where we applied this pre-trained model with other models to make an ensemble. However, the ensemble model showed that instead of applying a single model, the ensemble can perform well. A similar scenario is observed [21], where the single CNN model performed worse than ours.

## G. DISCUSSIONS

It is essential to recognize our suggested model's limitations and potential biases, particularly regarding its reliance on specific environmental conditions within the dataset. Since the training data predominantly originates from two particular

**TABLE 9.** Indicative comparison of the proposed work with existing research.

Ref.	Image samples	Classes	Model	Accuracy (%)
[2]	649	6	CNN with OSLEM (multilayer fusion attention)	94.28%
[13]	485	3	Random Forest Classifier	88.87%
[21]	43	3	CNN-HOG	90.00%
[22]	24	2	ResNet-50	59.33%
[16]	231	2	SVM (with augmentation)	94.12%
[18]	-	2	SVM	91.42%
[20]	350	2	CT-ANN	78.00%
[19]	-	2	FAST-PCA-NN	86.00%
Proposed	133; After augmentation: 1,997	7	Deep Ensemble Model (Performance Metric-Infused Weighting)	97.53%

locations in Bangladesh, there's a risk of bias introduced by the limited representation of different fish populations and environmental contexts. Future research could focus on gathering information from a broader range of sites, encompassing various freshwater fish habitats throughout Bangladesh, to address this limitation. By enlarging the dataset to include a more comprehensive representation of environmental factors and fish species from different geographical areas, we can improve the model's generalizability and efficacy in practical applications. Continued efforts to update and maintain this larger dataset will ensure the model's accuracy and dependability in real-world scenarios. In conclusion, while our investigation did not reveal evidence of biases or overfitting in the current dataset, it is imperative to acknowledge the potential impact of environmental factors on fish diseases and take proactive measures to address them.

Our research offers game-changing advantages in productivity, sustainability, and disease control for Bangladesh's aquaculture sector, with important practical ramifications. First of all, aquaculture farmers now have a useful tool for early disease detection and diagnosis thanks to the application of our sophisticated deep learning model for classifying diseases affecting freshwater fish. This ability is essential for prompt interventions, allowing farmers to deal with health problems before they become widespread epidemics. As a result, there may be significant decreases in the financial losses brought on by fish mortality and lower productivity. Second, by correctly classifying diseases according to two environmental factors, the model aids farmers in understanding the precise factors that contribute to the occurrence of disease. With this understanding, they can modify their risk-reduction strategies. Farmers can improve the overall productivity of their farms by creating healthier environments for their fish, such as by implementing targeted treatment protocols, optimizing feeding strategies, or adjusting water quality parameters. Furthermore, our model can be used as a basis for creating all-inclusive disease management systems that incorporate data analytics and real-time monitoring. This can further improve farmers' operational efficiency and decision-making processes by providing them with predictive capabilities and actionable insights.

In summary, our study provides useful instruments and approaches to enhance disease control in Bangladesh's aquaculture sector. Our model contributes to the expansion, resilience, and long-term sustainability of aquaculture operations in the area by facilitating early detection, customized interventions, and sustainability.

## VII. CONCLUSION

Our study introduces a novel method combining transfer learning with pre-trained CNN models and two ensemble techniques with adjustable learning rates to detect seven different types of freshwater fish diseases using augmented and non-augmented images. This approach improves neural network training using information from an initial task to enhance convergence, resulting in improved model performance, accuracy, and robustness on unidentified data. Our evaluation metrics show that pre-trained models with weighted ensemble learning (PMIWE), having a learning rate of 0.0001, outperform individual pre-trained models and ensemble techniques with different learning rates, achieving 97% accuracy, precision, recall, and F1-score. To ensure the interpretation of results and the decision-making process of neural network algorithms, we have used Grad-CAM for more concise visualization by highlighting the region of infected images, leading to effective fish disease detection. Along with concise visualization, this work contributes to reducing computational loss with huge training parameters during accuracy improvement, resulting in a computationally efficient solution compared to previous research in fish disease detection. Furthermore, our accuracy surpasses that of previous works using traditional machine learning methods.

As aquaculture is a growing and sustainable sector worldwide, further research is needed. Subsequent efforts could improve the interpretability of the ensemble model's predictions by utilizing novel techniques such as advanced segmentation and preprocessing techniques, attention mechanisms, and model distillation. These improvements would provide a more in-depth understanding of the decision-making mechanisms that underlie the classifications in the model. Furthermore, investigating how well the method works with different datasets and fish species shows potential for improving its robustness and real-world applicability.

It is crucial to acknowledge the presence of intrinsic limitations, such as possible biases in the training data and computational resource constraints, as these could affect the scalability and generalizability of our suggested ensemble model. Furthermore, incorporating real-time data collection and monitoring systems could improve our model's responsiveness and accuracy in disease detection and classification tasks, given the dynamic nature of fish diseases and the impact of environmental factors. We hope to progress the field of fish disease classification and promote more effective management techniques in aquaculture operations by carefully considering the inherent limitations and carefully navigating these future directions.

## REFERENCES

- [1] R. Subasinghe, D. Soto, and J. Jia, "Global aquaculture and its role in sustainable development," *Rev. Aquaculture*, vol. 1, no. 1, pp. 2–9, Mar. 2009.
- [2] Y.-P. Huang and S. P. Khabusi, "A CNN-OSELM multi-layer fusion network with attention mechanism for fish disease recognition in aquaculture," *IEEE Access*, vol. 11, pp. 58729–58744, 2023.
- [3] S. Aftabuddin, M. N. Islam, M. A. B. Bhuyain, M. A. Mannan, and M. M. Alam, "Fish diseases and strategies taken by the farmers in freshwater aquaculture at southwestern Bangladesh," *Bangladesh J. Zool.*, vol. 44, no. 1, pp. 111–122, Nov. 2016.
- [4] J. Sikder, K. I. Sarek, and U. K. Das, "Fish disease detection system: A case study of freshwater fishes of Bangladesh," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 867–871, 2021.
- [5] D. Li, X. Li, Q. Wang, and Y. Hao, "Advanced techniques for the intelligent diagnosis of fish diseases: A review," *Animals*, vol. 12, no. 21, p. 2938, Oct. 2022.
- [6] J.-H. Wang, S.-K. Lee, Y.-C. Lai, C.-C. Lin, T.-Y. Wang, Y.-R. Lin, T.-H. Hsu, C.-W. Huang, and C.-P. Chiang, "Anomalous behaviors detection for underwater fish using AI techniques," *IEEE Access*, vol. 8, pp. 224372–224382, 2020.
- [7] K. P. R. A. Haq and V. P. Harigovindan, "Water quality prediction for smart aquaculture using hybrid deep learning models," *IEEE Access*, vol. 10, pp. 60078–60098, 2022.
- [8] S. P. Khabusi and Y.-P. Huang, "A deep learning approach to predict dissolved oxygen in aquaculture," in *Proc. Int. Conf. Adv. Robot. Intell. Syst. (ARIS)*, Aug. 2022, pp. 1–6.
- [9] L. Kuang, P. Shi, C. Hua, B. Chen, and H. Zhu, "An enhanced extreme learning machine for dissolved oxygen prediction in wireless sensor networks," *IEEE Access*, vol. 8, pp. 198730–198739, 2020.
- [10] D. Li, J. Sun, H. Yang, and X. Wang, "An enhanced naive Bayes model for dissolved oxygen forecasting in shellfish aquaculture," *IEEE Access*, vol. 8, pp. 217917–217927, 2020.
- [11] W.-C. Hu, L.-B. Chen, B.-K. Huang, and H.-M. Lin, "A computer vision-based intelligent fish feeding system using deep learning techniques for aquaculture," *IEEE Sensors J.*, vol. 22, no. 7, pp. 7185–7194, Apr. 2022.
- [12] H. Yang, X. Wang, J. Sun, and D. Li, "Dissolved oxygen prediction using RBF network based on improved conjugate gradient method," in *Proc. IEEE 11th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Oct. 2020, pp. 515–518.
- [13] M. J. Mia, R. B. Mahmud, M. S. Sadad, H. A. Asad, and R. Hossain, "An in-depth automated approach for fish disease recognition," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7174–7183, Oct. 2022.
- [14] V. Lyubchenko, R. Matarneh, O. Kobylin, and V. Lyashenko, "Digital image processing techniques for detection and diagnosis of fish diseases," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 6, no. 7, pp. 79–83, 2016.
- [15] R. C. Barik, L. A. Kanuga, L. Mishra, A. K. Panda, and S. N. Panda, "Spot disease identification using unsupervised machine learning based image segmentation with its remedial solution in aquatic fauna," *J. Surv. Fisheries Sci.*, vol. 10, no. 2, pp. 912–922, 2023.
- [16] M. S. Ahmed, T. T. Aurpa, and M. A. K. Azad, "Fish disease detection using image based machine learning technique in aquaculture," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5170–5182, Sep. 2022.
- [17] A.-A. Nayan, A. N. Mozumder, J. Saha, K. R. Mahmud, A. K. Al Azad, and M. G. Kibria, "A machine learning approach for early detection of fish diseases by analyzing water quality," 2021, *arXiv:2102.09390*.
- [18] S. Malik, T. Kumar, and A. K. Sahoo, "Image processing techniques for identification of fish disease," in *Proc. IEEE 2nd Int. Conf. Signal Image Process. (ICSIP)*, Aug. 2017, pp. 55–59.
- [19] T. Kumar and A. Sahoo, "Fish disease detection using HOG and FAST feature descriptor," *Int. J. Comput. Sci. Inf. Secur.*, vol. 15, no. 5, pp. 13065–13074, 2018.
- [20] V. J. D. Almero, R. S. Concepcion, E. Sybingco, and E. P. Dadios, "An image classifier for underwater fish detection using classification tree-artificial neural network hybrid," in *Proc. RIVF Int. Conf. Comput. Commun. Technol. (RIVF)*, Oct. 2020, pp. 1–6.
- [21] A. Waleed, H. Medhat, M. Esmail, K. Osama, R. Samy, and T. M. Ghanim, "Automatic recognition of fish diseases in fish farms," in *Proc. 14th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2019, pp. 201–206.
- [22] F. Rachman, M. Akbar, and E. Putera, "Fish disease detection of epizootic ulcerative syndrome using deep learning image processing technique," in *Proc. Int. Conf. Fisheries Aquaculture*, Jan. 2023, pp. 23–34.
- [23] A. K. Jakhar, A. Gupta, and M. Singh, "SELF: A stacked-based ensemble learning framework for breast cancer classification," *Evol. Intell.*, vol. 17, no. 3, pp. 1341–1356, Jun. 2024.
- [24] T. Dang, T. T. Nguyen, J. McCall, E. Elyan, and C. F. Moreno-García, "Two-layer ensemble of deep learning models for medical image segmentation," *Cognit. Comput.*, vol. 16, no. 3, pp. 1141–1160, May 2024.
- [25] H. Gunasekaran, K. Ramalakshmi, D. K. Swaminathan, and M. Mazzara, "GIT-Net: An ensemble deep learning-based GI tract classification of endoscopic images," *Bioengineering*, vol. 10, no. 7, p. 809, Jul. 2023.
- [26] V. Anand, S. Gupta, D. Gupta, Y. Gulzar, Q. Xin, S. Juneja, A. Shah, and A. Shaikh, "Weighted average ensemble deep learning model for stratification of brain tumor in MRI images," *Diagnostics*, vol. 13, no. 7, p. 1320, Apr. 2023.
- [27] F. Younas, M. Usman, and W. Q. Yan, "A deep ensemble learning method for colorectal polyp classification with optimized network parameters," *Int. J. Speech Technol.*, vol. 53, no. 2, pp. 2410–2433, Jan. 2023.
- [28] T. Iqbal and M. A. Wani, "Weighted ensemble model for image classification," *Int. J. Inf. Technol.*, vol. 15, no. 2, pp. 557–564, Feb. 2023.
- [29] P. Timalisina, B. Gurung, R. Thapa, P. Kunwor, A. Rayamajhi, N. Pradhan, M. Bhandari, S. Singh, and A. Sharma, "Tail and fin rot disease and antibiotics resistance pattern in major rainbow trout farms of Nepal," *Int. J. Fisheries Aquatic Stud.*, vol. 10, no. 1, pp. 33–37, Jan. 2022.
- [30] J. L. Fryer and J. S. Rohovec, "Bacterial diseases of fish," in *Pathobiology of Marine and Estuarine Organisms*. Boca Raton, FL, USA: CRC Press, 2021, pp. 53–83.
- [31] S. K. Bhoi, S. K. Panda, K. K. Jena, C. Mallick, and A. Khan, "A fuzzy approach to identify fish red spot disease," *Grey Syst., Theory Appl.*, vol. 10, no. 3, pp. 249–263, Apr. 2020.
- [32] P. E. Arshad and J. Arockiaraj, "Pathogenicity and pathobiology of epizootic ulcerative syndrome (EUS) causing fungus *Aphanomyces invadans* and its immunological response in fish," *Rev. Fisheries Sci. Aquaculture*, vol. 28, no. 3, pp. 358–375, Jul. 2020.
- [33] S. Kumar, T. Sathish Kumar, R. Vidya, and P. K. Pandey, "A prospective of epidemiological intervention in investigation and management of argulosis in aquaculture," *Aquaculture Int.*, vol. 25, no. 1, pp. 303–325, Feb. 2017.
- [34] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
- [35] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *J. Big Data*, vol. 8, no. 1, p. 101, Dec. 2021.
- [36] F. Wagner, A. Eltner, and H.-G. Maas, "River water segmentation in surveillance camera images: A comparative study of offline and online augmentation using 32 CNNs," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 119, May 2023, Art. no. 103305.
- [37] A. Victor Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of pneumonia from chest X-ray images," *Global Transitions Proc.*, vol. 2, no. 2, pp. 375–381, Nov. 2021.
- [38] S. A. Albelwi, "Deep architecture based on DenseNet-121 model for weather image recognition," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 10, pp. 559–565, 2022.



- [39] K. Joshi, V. Tripathi, C. Bose, and C. Bhardwaj, "Robust sports image classification using InceptionV3 and neural networks," *Proc. Comput. Sci.*, vol. 167, pp. 2374–2381, Jan. 2020.
- [40] A. Batool and Y.-C. Byun, "Lightweight EfficientNetB3 model based on depthwise separable convolutions for enhancing classification of leukemia white blood cell images," *IEEE Access*, vol. 11, pp. 37203–37215, 2023.
- [41] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for Rice diseases classification," *Artif. Intell. Agricult.*, vol. 9, pp. 22–35, Sep. 2023.
- [42] S. Jana, A. I. Middy, and S. Roy, "Participatory sensing based urban road condition classification using transfer learning," *Mobile Netw. Appl.*, pp. 1–17, Feb. 2023, doi: [10.1007/s11036-023-02118-6](https://doi.org/10.1007/s11036-023-02118-6).
- [43] M. M. Shukla, B. K. Tripathi, T. Dwivedi, A. Tripathi, and B. K. Chaurasia, "A hybrid CNN with transfer learning for skin cancer disease detection," *Med. Biol. Eng. Comput.*, pp. 1–15, May 2024, doi: [10.1007/s11517-024-03115-x](https://doi.org/10.1007/s11517-024-03115-x).
- [44] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
- [45] J. S. Murugaiyan, M. Palaniappan, T. Durairaj, and V. Muthukumar, "Fish species recognition using transfer learning techniques," *Int. J. Adv. Intell. Informat.*, vol. 7, no. 2, p. 188, Jul. 2021.
- [46] J. Huang, W. Gong, and H. Chen, "Menfish classification based on Inception\_V3 convolutional neural network," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 677, no. 5, Dec. 2019, Art. no. 052099.
- [47] M. A. Morid, A. Borjali, and G. Del Fioli, "A scoping review of transfer learning research on medical image analysis using ImageNet," *Comput. Biol. Med.*, vol. 128, Jan. 2021, Art. no. 104115.
- [48] N. S. Shaik and T. K. Cherukuri, "Transfer learning based novel ensemble classifier for COVID-19 detection from chest CT-scans," *Comput. Biol. Med.*, vol. 141, Feb. 2022, Art. no. 105127.
- [49] S. Vallabhajosyula, V. Sista, and V. K. K. Kolli, "Transfer learning-based deep ensemble neural network for plant leaf disease detection," *J. Plant Diseases Protection*, vol. 129, no. 3, pp. 545–558, Jun. 2022.
- [50] N. Gianchandani, A. Jaiswal, D. Singh, V. Kumar, and M. Kaur, "Rapid COVID-19 diagnosis using ensemble deep transfer learning models from chest radiographic images," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 5, pp. 5541–5553, May 2023.
- [51] A. Das, M. M. Hoque, O. Sharif, M. A. A. Dewan, and N. Siddique, "TEmoX: Classification of textual emotion using ensemble of transformers," *IEEE Access*, vol. 11, pp. 109803–109818, 2023.
- [52] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, "TEMPO: Prompt-based generative pre-trained transformer for time series forecasting," 2023, *arXiv:2310.04948*.
- [53] B. Zhou, B. Huang, Y. Su, and C. Zhu, "Interleaved periodic event-triggered communications-based distributed formation control for cooperative unmanned surface vessels," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 19, 2024, doi: [10.1109/TNNLS.2024.3351218](https://doi.org/10.1109/TNNLS.2024.3351218).
- [54] H. Peng, B. Huang, M. Jin, C. Zhu, and J. Zhuang, "Distributed finite-time bearing-based formation control for underactuated surface vessels with levant differentiator," *ISA Trans.*, vol. 147, pp. 239–251, Apr. 2024.
- [55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [56] J. Jepkoech, D. M. Mugo, B. K. Kenduinywo, and E. C. Too, "The effect of adaptive learning rate on the accuracy of neural networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 8, pp. 736–751, 2021.
- [57] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.
- [58] S. Chaudhury and T. Yamasaki, "Robustness of adaptive neural network optimization under training noise," *IEEE Access*, vol. 9, pp. 37039–37053, 2021.
- [59] M. Y. Kiang, "A comparative assessment of classification methods," *Decis. Support Syst.*, vol. 35, no. 4, pp. 441–454, Jul. 2003.
- [60] C. Marzban, "The ROC curve and the area under it as performance measures," *Weather Forecasting*, vol. 19, no. 6, pp. 1106–1114, Dec. 2004.
- [61] M. Banerjee, M. Capozzoli, L. McSweeney, and D. Sinha, "Beyond kappa: A review of interrater agreement measures," *Can. J. Statist.*, vol. 27, no. 1, pp. 3–23, Mar. 1999.
- [62] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews correlation coefficient metric," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0177678.



**ABDULLAH AL MARUF** received the B.Sc. degree from Bangladesh University of Business and Technology (BUBT), in 2022. This milestone set the stage for his subsequent successes. He stepped into a software engineer position after completing his academic studies with ease, where he developed his algorithmic thinking and problem-solving abilities. His quest eventually took him to the exciting field of machine learning, driven by his unrelenting curiosity. He accepted

the post of a Machine Learning Engineer, a position that used his existing experiences while opening him new opportunities for advancement, with a thirst for innovation. His research interests include bioinformatics, image processing, human psychology in AI, and natural language processing (NLP). Due to this intense interest, he has published several noteworthy research papers in esteemed journals and at important international conferences.



**SINHAD HOSSAIN FAHIM** received the B.Sc. degree in computer science and engineering from Stamford University, Bangladesh. He is working in the field of CS to maintain his academic background after the completion of his graduation. He is also continuing as an Instructor with the "Sheikh Rasel Digital Laboratory Python Programming" Project under the ICT Division to accelerate digital literacy and programming skills development for children of Bangladesh.

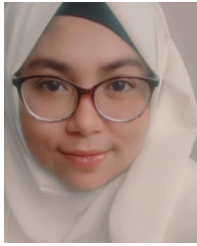
He specializes in machine learning and deep learning, with a focus on computer vision, actually forming the basis of this specialization into a career devoted to the progress of artificial intelligence (AI). His research and implementation of social AI work ensure that these changes are being effectively translated to find successful, practical applications. He is very much interested in using AI for the betterment of mankind in ways for make this world a better place and he looks forward to getting an opportunity to contribute to this space meaningfully.



**RUMAISHA BASHAR** is currently pursuing the B.Sc. degree in computer science and engineering with International Islamic University Chittagong. She has taken part in a number of programming contests. Besides her studies, she has worked on numerous software development projects that showcase her abilities in various fields. She has experience in the fields of machine learning and deep learning, focusing on the application of natural language processing and image datasets.

This academic foundation provided the starting point for a career focused on the advancement of AI. She actively participates in the advancement of novel methodologies that create the cognitive capabilities of machines, with a particular focus on natural language processing and the interpretation of image data. Such methodologies are referred to as transformer-based models and they intrigued her interest significantly. She tries to use AI exclusively for significant technological advancements. With a combination of comprehensive research and inventive application, she hopes to positively impact both technological advancement and practical consequences in the field of AI in an attempt to push the limits of artificial intelligence (AI).





**ROWNUIK ARA RUMY** received the Bachelor of Science degree from Stamford University Bangladesh. She practiced in the field of machine learning and deep learning with a particular interest in natural language processing (NLP) and image dataset applications. This scholarly base provided the building blocks for a career dictated by AI enhancements. She is part of the development of new approaches that increase the intelligence of machines, especially in NLP and image data interpretation. Such approaches are known as transformer-based models in which she is taking particular interest. She works toward using AI strictly for significant technical progressions. She intends to push the limits of AI technology by aspiring to positively influence both technological development and practical aftermaths in the field of AI via a combination of thorough research and innovative application.



**SHAHARIOR ISLAM CHOWDHURY** is currently pursuing the degree in computer science and engineering with Bangladesh University of Business and Technology (BUBT). He is deeply passionate about exploring pioneering technologies, particularly in the fields of image processing, computer vision, natural language processing (NLP), and low resource language technologies. His research interests include machine learning and telecommunication security and advancement of electronic transaction systems.



**ZEYAR AUNG** (Senior Member, IEEE) received the Ph.D. degree in computer science from the National University of Singapore, in 2006. From 2006 to 2010, he was a Research Fellow with the Institute for Infocomm Research (I2R), Agency for Science, Technology, and Research (A\*STAR), Singapore. In 2010, he joined the Masdar Institute, which later became part of Khalifa University, United Arab Emirates, as an Assistant Professor. He is currently an Associate Professor with the Department of Computer Science, Khalifa University. He is also a member of the Center for Secure Cyber-Physical Systems (C2PS). His current research interests include data analytics, machine learning, and their applications in various domains, such as cyber security, social media, financial systems, renewable energy, and environmental science.

...