Computers in Biology and Medicine 192 (2025) 110273

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Deep learning meets marine biology: Optimized fused features and LIME-driven insights for automated plankton classification

Muhammad Hassan [a], Giovanna Salbitani [a,*], Simona Carfagna [a], Javed Ali Khan [b]

[a] Department of Biology, University of Naples Federico II, Naples, Italy
[b] Department of Computer Science, University of Hertfordshire, Hatfield, UK

ARTICLE INFO

Keywords:
Plankton classification
Transfer learning
Deep learning
Feature fusion
Whale optimization algorithm
LIME

ABSTRACT

Plankton are microorganisms that play an important role in marine food webs as primary producers in the trophic web. Traditional plankton identification methods using manual microscopy and sampling are time-consuming, labor-intensive, and prone to errors. Deep learning has improved the automation of plankton identification, but it remains challenging to achieve high accuracy and efficiency in computation with limited labeled data. In this paper, we proposed an improved plankton classification model that is more accurate and interpretable. We train two models, InceptionResNetV2 (transfer learning) and DeepPlanktonNet (from scratch), on the WHOI dataset. We utilize feature fusion to supplement feature representation, merging the outputs of both models. Feature selection is achieved through the Whale Optimization Algorithm (WOA), eliminating redundancy and making it more computationally efficient. Additionally, we also employ Local Interpretable Model-agnostic Explanations (LIME) to make the model more interpretable and gain insights into how the model makes decisions. Additionally, feature selection using WOA reduces feature space and has less inference and computational cost. Our method achieves a classification accuracy of 98.79 %, which is better than previous state-of-the-art methods. For robustness testing, we train nine machine learning classifiers on the optimized features. By significantly improving classification accuracy and speed, our method enables large-scale ecological surveys, water quality monitoring, and biodiversity studies. These advances allow researchers to and environmental scientists to automate plankton classification more reliably, supporting marine conservation and resource management.

1. Introduction

Plankton, a diverse group of microscopic organisms (both plants and animals) living in aquatic habitats, play a crucial role as primary producers in the trophic marine chain and contributors to the global carbon cycle [1]. Plankton is an essential factor for marine life due to its great sensitivity to environmental changes and its crucial role in maintaining the biological balance of the seas [2]. Being bioindicators of environmental status, the qualitative and quantitative study of plankton assumes considerable importance for protecting marine ecosystems. The identification and classification of plankton species is challenging due to its immense diversity and the time-consuming nature of traditional taxonomic methods.

Plankton has traditionally been investigated using tools such as Niskin bottles [3], pumps [4], and towed nets [5], by which samples are collected from large areas of the ocean. Niskin bottles are used to sample single water columns at specific depths, where accurate sampling of the plankton and water characteristics is performed. Towed nets provide sampling in the form of a continuum along an assigned transect, providing total information regarding the distribution and density of plankton. Pumps are used for sampling high rates of water for bulk filtering the plankton sample and detecting the fine-scale pattern of changes in plankton assemblages. These operations require significant field activities in very poor marine environments with mostly vague weather conditions and navigating the complexities in covering distant sea areas. Following that, such a time-consuming process is the job of only marine biology experts [6]. Samples obtained were examined piece by piece in the microscopes by professionals [7], who sorted them out and identified the type of plankton into different classes. These time-consuming procedures had much manpower and were prone to human mistakes [8], further reducing the efficiency of plankton study.

Conventional methods of plankton population analysis presented

* Corresponding author.
E-mail address: giovanna.salbitani@unina.it (G. Salbitani).

https://doi.org/10.1016/j.compbiomed.2025.110273
Received 27 February 2025; Received in revised form 22 April 2025; Accepted 24 April 2025
Available online 1 May 2025

considerable problems and considerably delayed the speed at which knowledge about plankton and their functions within marine ecosystems could be gained [9]. Since then, the plankton research discipline has continued to look for ways to improve efficiency and precision in data collection and analyses. Efforts like this involve highly sophisticated technologies: computer vision coupled with deep transfer learning approaches to auto-classify plankton images.

Although computer vision and deep learning have improved considerably, automated plankton classification remains challenging due to three critical issues: (1) the limited availability of labeled datasets, (2) the lack of interpretability in deep learning models, and (3) the high computational cost associated with feature extraction and classification. Traditional CNN-based models such as VGG16, Inception, and ResNet have been widely used for feature extraction, leveraging pre-trained architectures to accelerate convergence during training. However, these models rely on generic feature extraction techniques, which may not be well-suited for plankton images due to their highly intricate and variable morphological structures. Furthermore, CNN-based models typically function as black boxes, making it difficult to understand and interpret their classification decisions an essential requirement for ecological and environmental monitoring applications. Additionally, feature extraction often results in high-dimensional feature representations, leading to increased computational demands and potential overfitting.

Feature selection plays a crucial role in addressing these challenges, yet existing plankton classification studies rarely optimize feature extraction in combination with feature fusion and selection strategies. In this context, the Whale Optimization Algorithm (WOA) offers a compelling solution. WOA is a metaheuristic optimization algorithm that mimics the bubble-net hunting strategy of humpback whales. It is particularly effective in handling redundant and high-dimensional feature sets, allowing for the selection of the most relevant features while reducing computational costs. By incorporating WOA-based feature selection, our approach enhances computational efficiency and model generalizability without compromising classification accuracy.

To address the above challenges and improve the plankton classification system, this study integrates transfer learning with DeepPlanktonNet model training from scratch, feature fusion, and optimization techniques. We employ two models: InceptionResNetV2, utilizing transfer learning for efficient feature extraction, and a novel DeepPlanktonNet model, trained from scratch to capture dataset-specific features that may not be learned by pre-trained architectures. Feature fusion combines the best of these models for improved accuracy and robustness. WOA is then applied to select the most relevant features, reducing computational overhead while retaining classification performance. Finally, to enhance model transparency, we employ the Local Interpretable Model-Agnostic Explanations (LIME) technique, which provides visual and quantitative insights into the model's decision-making process. This is particularly vital for ecological studies, where interpretability is crucial for trusting and validating AI-driven classification results. Beyond its technical contributions, this study aligns with recent advancements in AI-driven environmental research and marine ecosystem monitoring. Automated plankton classification plays a critical role in applications such as ecological monitoring, climate change research, water quality assessment, and marine biodiversity conservation. By developing an efficient, accurate, and interpretable classification framework, this research contributes to the broader goal of leveraging AI and machine learning to enhance environmental sustainability and marine ecosystem protection. This is vital in transparency and gives the framework high accuracy and explainability. The main contributions of this paper are summarized as follows:

2. This study integrates InceptionResNetV2 and a novel DeepPlanktonNet model, combining their strengths to improve classification accuracy;
3. Feature fusion is introduced to leverage complementary features from the models, and the Whale Optimization Algorithm was applied for feature selection, ensuring computational efficiency;
4. The LIME technique was employed to interpret and explain the model's decisions, addressing the critical challenge of transparency in deep learning;
5. Experiments were conducted using the WHOI publicly available dataset, demonstrating the proposed framework's superior accuracy and efficiency compared to conventional methods.

2. State of the art

The acquisition and curation of high-quality plankton image datasets are inherently coupled with the chronological progress of specialized-purpose imaging hardware. During the past three decades, the development of such equipment has advanced extensively in favor of increasingly detailed analysis and production of benchmark standard datasets. All of this has served an instrumental role in rigorous methodology evaluation for new techniques and progress in the field of plankton studies.

The Davis et al. [10] Video Plankton Recorder (VPR) was a new device that was specifically designed to overcome the existing limitations in applying conventional plankton sampling by enabling in situ visualization of plankton communities at different scales using multiple cameras with varying magnifications. Before the VPR, imaging systems were primarily directed toward object-counting population estimation [11–13], with other approaches such as Silhouette Photography [14] and later photographic camera suspension on sampling nets [15,16]. The VPR popularity spurred the development of several state-of-the-art imaging systems, including UVP [17], SIPPER [18], among others, being widely utilized today in plankton image benchmark datasets [19]. Still more recently, Dark Field Acquisition Systems like Scripps Plankton Camera (SPC) [20] and Dual Scripps Plankton Camera (DSPC) [21] have been conceived, providing high-contrast color images with background-free illumination, yet again transforming the field of plankton imaging for enhanced precision and efficacy of analysis. In more contemporary times, Dark Field Acquisition Systems like the Scripps Plankton Camera (SPC) [20] and Dual Scripps Plankton Camera (DSPC) [21] have been made to create high-contrast color images with defined backgrounds, hence further enhancing plankton imaging capacity for better accurate and efficient analysis. Following these advances in imaging technology, Luo et al. [22] proposed using deep learning techniques, such as convolutional neural networks (CNNs), to further improve the automatic plankton identification from in situ imaging systems and enable better examination of the data collected.

A whole image processing pipeline consisting of preprocessing, segmentation, classification, and postprocessing was designed to identify classes of 108 organisms with high accuracy. This excludes the rarest taxa, which are difficult to handle and result in an increase in classification accuracy greater than 90 %. Conversely, Matuszewski et al. [23] proposed a novel plankton-monitoring system optimized for large volume, high-throughput analysis that integrates data acquisition, detection, and identification by using a technique known as Visual Rhythm with CUDA, which makes it reach a very high performance of 720 MB/s. By the method, the features of organisms have been extracted and their classification by Support Vector Machine with an accuracy of up to 92 % by reduced amount of storage. This allows automated ecological monitoring, which is useful even for environmental research. Similarly, Rawat et al. [24] proposed a CNN-based approach to classify plankton. The framework's overall accuracy on the test set was 95.3 %. It was trained on a dataset comprising 235 images of plankton from five distinct categories (insert the 5 categories here). The framework fared better than SVMs and random forests, two common machine-learning methods. A ResNet-based model is proposed for plankton classification by Li et al. [25]. The system, trained with a dataset consisting of 30,336 images belonging to 121 different plankton classes, yields a top 5 accuracy of 95.8 % for the test set.

A transferred parallel CNN framework is proposed for plankton classification on the large imbalanced dataset by Wang et al. [26]. This approach achieves an F1 score of 0.5444 on a test set using a CNN pre-trained model with small classes for feature extraction that will be subsequently tweaked on all the data. Deep transfer learning-based methods get very high accuracy in the plankton classification features in the data sets that are available to the public. The other deep learning methods are outperformed by fine-tuning the pre-trained CNN models, and combining several CNN models can result in even better performance [27]. Deep learning methods using CNN outperformed other machine learning methods, which accurately identified phytoplankton from microscope images. With possible applications to environmental monitoring, aquaculture, fisheries, and research, deep learning could be a viable method for automated plankton identification [28]. Adding context metadata to plankton image classifiers on a dataset of 350,000 images from 26 different plankton classes resulted in an accuracy increase of up to 5 %, showing the value of metadata as a tool for better image classification.

Kerr et al. [29] proposed an automatic plankton classification, which includes deep learning algorithms such as convolutional neural networks and multilayer perceptron's to overcome class imbalance. This imbalance occurs when certain plankton typologies are over-represented in datasets, which can skew the accuracy of models. Using a collaborative framework formed by multiple deep-learning models, they achieved a notable increase in the detection accuracy of poorly represented plankton classes. Lumini et al. [30,31] employed a deep learning technique for monitoring and classification of marine habitats. The work by Lumini et al. explores several pre-trained CNNs; these CNNs are optimized across several datasets to leverage their unique advantages. Notably, the study investigates the testing of different CNN models combined into a heterogeneous ensemble to increase the robustness and accuracy of underwater imagery analysis. The insights into this challenge are that many datasets involved yield a much better ensemble technique than single models, which provides the most feasible solution for properly classifying plankton and coral species, which is so important to ocean health. This indicates a potential path for future research in automated marine ecosystem monitoring and takes a major step toward applying sophisticated machine-learning approaches to environmental protection. Py et al. [32] proposed some constraints in designing a deep convolutional neural network structure to guarantee performance gain when going deep into the architecture and performing plankton image classification. The study proposed by Py et al. also focuses on optimizing CNNs for better accuracy in identifying different plankton species from images, exploring various CNN architectures and training techniques.

Current studies have applied machine learning and optimization algorithms across a multitude of domains, stressing their universal applicability. Nagwan et al. [33] proposed the hybrid metaheuristic approach for image-based classification and optimization methods for better classification accuracy. Massimiliano et al. [34] proposed an anomaly detection system that recognizes important morphological variations in phytoplankton populations as a sign of environmental disturbances. The method uses Vision Transformer-based feature extraction with dimensionality reduction through PCA and class-wise anomaly detection models. While this method is effective at identifying population-level phytoplankton changes over time, it differs from our approach in final goal anomaly detection rather than species-level classification. Similarly, Loris et al. [35] explored CNN and transformer model fusion together with novel Adam-based optimization variants for improved general image classification on varied datasets. While the study effectively demonstrates the benefit of ensemble learning and transformer models (DeiT, ViT, Swin, CoAt), the primary focus is on the optimization of CNN-Transformer fusion rather than feature-level fusion. Unlike the proposed study, which employs deep feature fusion and WOA-based feature selection for high-precision species classification, both methods [34,35] have different targets. The outlier detection system in Ref. [34] is designed to detect outliers and not classify plankton into specific species. In contrast, the proposed work specifically integrates feature representations of InceptionResNetV2 and DeepPlanktonNet, where WOA is applied for optimal feature selection. Not only does this enhance classification, but it reduces computational complexity as well. Again, whereas [35] is engaged in CNN-Transformer fusion optimization, our work is dealing with model interpretability using LIME, enhancing transparency in decision-making for the classification of plankton.

Previous works on plankton classification using transfer learning, CNN, and deep learning models have achieved a high accuracy of 95.8 %. However, these methods still face certain critical limitations: explainability, optimization, computation costs, and issues related to imbalance. Though models like ResNet50 and VGG16 provide automated procedures, they are computationally inefficient and challenging to generalize. Few-shot learning approaches, dealing with limited data, have also been underperforming due to class imbalances. This paper addresses these challenges by proposing a hybrid model incorporating transfer learning into handcrafted feature extraction and utilizing the Whale Optimization Algorithm to optimize feature selection. In general, incorporating LIME into the proposed model for its interpretability increases the accuracy of the model's classification, therefore adding value to this field of plankton detection with an automatic, scalable, efficient, and interpretable model for environmental monitoring. Table 1 outlines the existing Plankton classification approaches.

3. Methodology

This study used a hybrid approach combining a pretrained deep learning model and a *DeepPlanktonNet* built neural network to classify plankton species. This hybrid approach enabled efficient feature extraction and fine-tuned classification, ensuring high accuracy even in complex and diverse plankton morphologies. The proposed research methodology is shown in Fig. 1, which presents an abstract representation of the proposed hybrid approach for classifying plankton. Below, we elaborated in detail on each methodological step.

3.1. Dataset

The dataset used here for this research has been provided by WHOI (https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.4319/lom.2007.5.204) and proves to be an important dataset used in the analysis and research in the field of study. This dataset used for the research study was gathered with the help of an Imaging *FlowCytobot* (IFCB), from the water samples drawn from Woods Hole Harbor (41°31′36″N 70°39′47″W). Sampling was carried out during the late autumn and early spring seasons of 2004 and 2005. The collected samples are available at the provided source [38]. The dataset consists of 6598 images, classified into 22 different Classes, as shown in Fig. 2.

The majority of these categories represent phytoplankton taxa at the genus level. Specifically, 16 categories are attributed to diatoms, including *Asterionellopsis* spp., *Chaetoceros* spp., *Cylindrotheca* spp., *Cerataulina* spp., as well as morphologically similar species of *Dactyliosolen* such as *D. fragilissimus* and other species morphologically resembling *D. blavyanus*. Other diatom categories include *Dinobryon* spp., *Ditylum* spp., *Euglena* spp., and other euglenoids, *Guinardia* spp., *Licmophora* spp., *Phaeocystis* spp., *Pleurosigma* spp., *Pseudonitzschia* spp., *Rhizosolenia* spp., and rare instances of *Proboscia* spp., *Skeletonema* spp., *Thalassiosira* spp., and similar centric diatoms. The remaining categories encompass mixtures of particles and cell types that share morphological similarities, including ciliates, detritus, dinoflagellates larger than 20 μm, nanoflagellates, other cells smaller than 20 μm, and other single-celled pennate diatoms. The dataset was partitioned into two sets, namely the training and testing sets, with an equal number of images in each. Each set comprises 22 distinct categories, with 300 unique images

Table 1
Overview of current methods for plankton classification.

Reference	Methods	Advantages	Limitations	Dataset	Publication Year	Accuracy
[24]	Transfer learning	Reduced computational cost	Lack of explainability	235 images gathered from internet sources	2016	95.3
[25]	Residual Neural network + VGG 19	Process automation	No efficient	Kaggle	2016	95.8
[26]	transferred parallel neural network	Overcoming the catastrophic forgetting problem	Small number of images used for training	WHOI	2018	94.98
[27]	Fine tuning of deep learning approaches and transfer learning	Diverse dataset	No optimization technique for hyperparameter	WHOI, ZooPlankton, Kaggle	2019	95.27
[28]	Pre-trained VGG16	Process automation	Imbalance dataset used for training	RCO-BRIN	2022	88.75
[29]	MLP with collaborative Deep learning models	Contextual data integration	Detection problems with certain minority classes	Custom Dataset of Planktons	2020	97.40
[30]	Transfer learning	Requiring a few parameters tuning	Lack of explainability	**Plankton Dataset:** WHOI, ZooPlankton, Kaggle **coral dataset:** EILAT, RSMAS	2020	95.80
[32]	Classification using CNN	Solve constrained optimization problem	Not so efficient	Dataset is collected from F.G. Walton Smith in Straits of Florida	2016	62.50
[36]	TL Resnet50	Diverse dataset used	Higher computational cost	WHOI, ZooPlankton, Kaggle	2023	96.6
[37]	Few-shot learning + Transfer learning	Minimize the intra-class distance of the depth features	Highly imbalanced dataset	WHOI, ZooPlankton, Kaggle, miniP-Plankton	2021	72.87

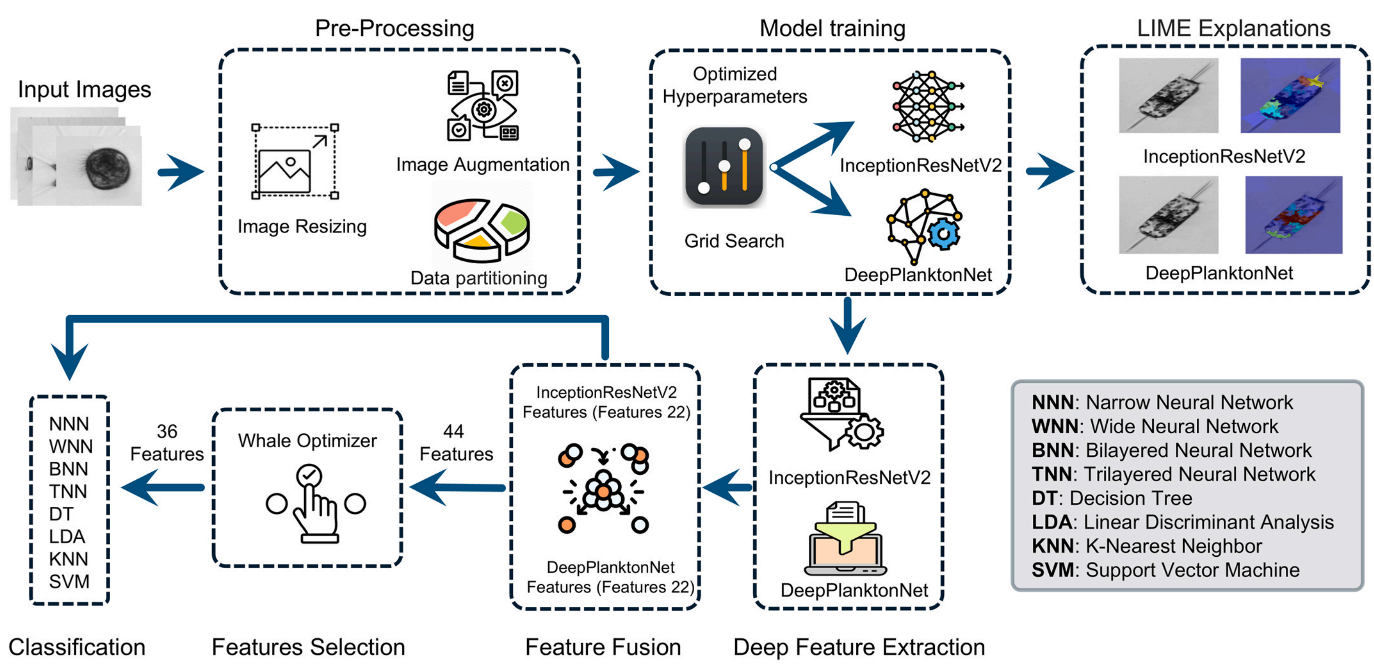


Fig. 1. Proposed methodology for plankton's classification using hybrid and LIME explainability model.

per category except Detritus class having 298 images shown in Fig. 3. Following our experimental design, the training set was used for the purpose of learning, while the testing set was employed to evaluate the performance of the classification system.

3.2. Preprocessing

Preprocessing is a crucial step in a machine learning-based experiment that involves preparing raw data to ensure it is clean, consistent, and ready for training. Raw data collected from various sources often contains noise, missing values, inconsistencies, or irrelevant information. Preprocessing aims to address these issues, transforming raw data into a format that can enhance the performance of machine learning algorithms and improve the overall quality of insights derived from the data. The three preprocessing steps applied to the dataset are data augmentation, image resizing, and data partitioning.

3.2.1. Data augmentation

Data augmentation is a widely known technique in both machine learning and deep learning for artificially inflating the size and variation of any given training dataset by various transformations of the data already available. This is especially useful in the case of a limited or

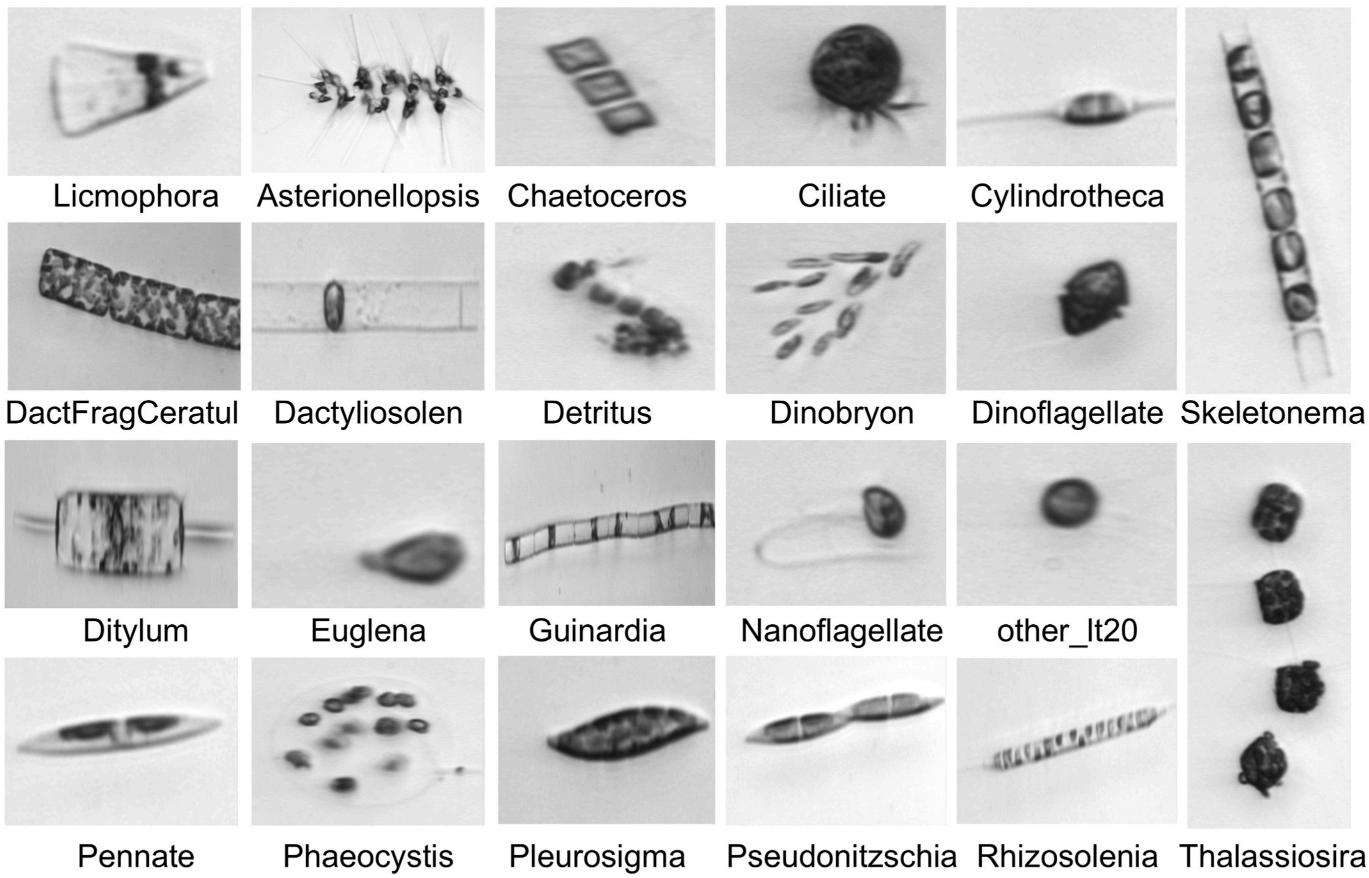


Fig. 2. An overview of the WHOI dataset's 22 classes, with a sample image from each category.

Fig. 3. Distribution of images per class and total number of classes. The y-axis represents the number of images in each class, while the x-axis shows the total number of classes in the dataset.

imbalanced dataset, which helps in improving model generalization, avoiding overfitting and hence providing good performance. First, the freely available WHOI dataset was obtained from Lumini et al. [31]. This dataset contains 22 categories and was placed in the training directory. Then, the function imageDataStore was used to extract the images from the training directory within the dataset.

Therefore, data augmentation is used here to test the capability of transfer learning and the generalized performance of DeepPlanktonNet. Data augmentation is a method of enriching the available dataset by applying numerous procedures to existing data without data acquisition, often improving image classification. The small number of images in the dataset made implementing the augmentation strategy necessary. The images inside the training set underwent random rotation within a range of -20 to 20° and arbitrary translation of up to thirty pixels in both vertical and horizontal directions. This process was carried out to generate supplementary images. The imageDataAugmenter function dynamically generates augmented image sets during each training.

The data augmentation approach substantially increased the number of images in the training set. This augmentation facilitated the training of the proposed deep learning-based approach with a much larger number of training images, enhancing its effectiveness and generalizability. Additionally, the augmented images were only used for training in the proposed framework. In contrast, the testing phase exclusively utilized original images from the dataset to evaluate the performance of the proposed approach.

3.2.2. Image resizing

As part of the preprocessing pipeline, the plankton images were resized to a uniform resolution to ensure compatibility with the input dimensions of the deep learning models. The input images from the dataset were scaled to meet the input image specifications of the pre-trained CNN model. The dataset included images of diverse dimensions, but the model necessitates input images with dimensions of 299-by-299. Consequently, before integrating into the deep learning network, the training and testing images underwent an automated scaling process using enhanced image datasets from transfer learning.

3.2.3. Data partitioning

Data partitioning is the last step of the preprocessing pipeline, which splits the dataset into a 5-fold cross-validation scheme so that it is possible to robustly and reliably test the model. In the process, the dataset is divided into five partitions, and in each iteration, four of them are used for training, and one is used for testing. This process is repeated for five iterations so that all data points are included in both the training and test sets. This strategy enhances the generalizability of the model by reducing variability in performance over different subsets of data.

3.3. Deep learning models details

This sub-section describes the architecture details of both deep learning models (InceptionResNetv2 and DeepPlantonNet) utilized in this study.

3.3.1. InceptionResNetv2

InceptionResNetV2 [39] is a deep convolutional neural network that effectively incorporates the inception architecture into residual connections; hence, it is powerful yet efficient. Considering the design of the inception blocks to capture multi-scale features, the model uses parallel convolutional layers with kernel sizes 1×11, 11×1, 3×33, 33×3, and 5×55, 55×5. This helps the model in learning spatial hierarchies from the input data effectively. The augmented ones with residual connections enable the easy flow of gradients at backpropagation, which are responsible for handling the vanishing gradient problem. The architecture is also supported by batch normalization in stabilizing and giving speed to training while factoring down the convolutions for less computational cost.

It is structured as several stem, reduction, and inception-residual blocks, where the stem block extracts low-level features, and reduction blocks down-sample the feature maps while maintaining critical information. The model terminates with a global average pooling layer, a dense layer, and a softmax layer for classification. Pre-trained on ImageNet, InceptionResNetV2 provides a robust feature representation, making it highly suitable for transfer learning. This model was fine-tuned from the pre-trained weights by changing its domain to this dataset, thereby utilizing the power of deep feature extraction and efficient usage of parameters, which are good at accuracy and computational efficiency.

InceptionResNetV2 was chosen because it can grasp the most complex patterns by integrating the Inception architecture with residual connections, offering a trade-off between accuracy and computational efficiency. Parallel convolutional layers at multiple scales help capture a wide range of features, while residual connections reduce the vanishing gradient problem and thus allow the training of deeper networks. Its efficient design and batch normalization make its training even faster while minimizing computational overhead. Pretraining on ImageNet also gives a very good starting point for transfer learning so that it can utilize strong pre-extracted features highly transferable to other tasks. InceptionResNetV2 will be a good choice for Plankton classification because it has achieved good results in various domains. At the same time, high performance can be attained with good computational efficiency by making a trade-off.

3.3.2. DeepPlanktonNet

The proposed convolutional neural network starts with an input layer that takes an image of size $224 \times 224 \times 3$. This input is preprocessed by normalizing the data using z-score normalization, which helps in standardizing the pixel intensity values for efficient training. The first convolutional layer (Conv1) applies 64 filters of size 7×7 with a stride of 2, followed by a max pooling layer (Pool1) with a pool size of 3×3 and a stride of 2. These initial layers extract coarse features while down-sampling spatial dimensions to reduce computational complexity. Moreover, the residual connections begin with the Res2a block, which operates at a spatial resolution of $56 \times 56 \times 64$. Each residual block contains two convolutional layers and uses Leaky ReLU activations instead of ReLU to avoid the problem of dead neurons. Then, the architecture goes deeper: each subsequent residual block, namely Res3a, Res4a, and Res5a, further reduces the spatial dimensions and increases the number of feature maps. These blocks operate at resolutions of $28 \times 28 \times 128$, $14 \times 14 \times 256$ and $7 \times 7 \times 512$, respectively, using strides of 2 for down sampling.

This consists of a global average pooling layer that reduces the feature maps into one vector, drastically reducing the number of trainable parameters. The output is fed into a fully connected layer of 1096 neurons, followed by a dropout layer with a rate of 0.5 to avoid overfitting. The model finally ends with a dense output layer with 22 neurons, corresponding to the number of classes, and applies a softmax activation function to produce class probabilities. This architecture balances depth and efficiency and is well-suited for plankton classification tasks. Table 2 contains details on the architecture of the proposed model.

3.4. Feature extraction

3.4.1. Pre-trained model feature extraction

Subsequently, we used the pre-existing deep neural network model, namely Inceptionresnetv2, to evaluate its efficacy in identifying and classifying various types of plankton. The proposed transfer learning

Table 2
DeepPlanktonNet: A deep CNN architecture with convolutional layers, residual blocks, and pooling for efficient plankton classification, outputting 22 categories.

Layer Name	Layer Type	Output Size	Key Parameters
Input	Image Input Layer	$224 \times 224 \times 3$	Normalization: zscore
Conv1	Convolutional Layer	$112 \times 112 \times 64$	Filter: 7×7, Stride: 2
Pool1	Max Pooling Layer	$56 \times 56 \times 64$	Pool: 3×3, Stride: 2
Res2a	Residual Block	$56 \times 56 \times 64$	2 Conv Layers, Leaky ReLU
Res3a	Residual Block	$28 \times 28 \times 128$	Stride: 2
Res4a	Residual Block	$14 \times 14 \times 256$	Stride: 2
Res5a	Residual Block	$7 \times 7 \times 512$	Stride: 2
Pool5	Global Avg. Pooling	$1 \times 1 \times 512$	–
Fully Connected	Dense Layer	1096	Dropout: 0.5
Output	Dense + Softmax Layer	22	–

(TL) model comprised layers from a pre-trained network, with the final three layers being adjusted to accommodate the additional image classifications, namely 22 classes. The transfer learning model used a softmax layer to do image classification and classify images into a total of 22 distinct classes. In the context of InceptionResNetV2, the components known as "predictions," "predictions_softmax," and "ClassificationLayer_predictions" were substituted with a "fully connected layer," a "softmax layer," and a "classification output" layer, respectively. The extra layers were integrated with the network's final transferred layer, the "average pooling" layer. For the features extraction, we use the last fully connected layer, "new_fc" and the feature vector is (N × 22) represented with V1.

3.4.2. DeepPlanktonNet feature extraction

This DeepPlanktonNet is a ResNet-inspired architecture designed for Plankton 22-class image classification. It features an input layer for 224 × 224 RGB images, followed by convolutional layers, batch normalization, and leaky ReLU activations. Residual blocks with skip connections enhance gradient flow, while filters progressively increase from 64 to 512, reducing spatial dimensions via strided convolutions and pooling layers. The model ends with global average pooling, fully connected layers, dropout for regularization, and a softmax layer for classification, combining depth and efficiency for robust performance. Features are extracted from the last fully connected layer, "fc8" and the feature vector is (N × 22) represented with V2.

3.5. Feature fusion

Feature fusion is employed to combine InceptionResNetV2 and DeepPlanktonNet's various feature sets into a more discriminative and informative representation. The pre-trained InceptionResNetV2 can extract generalized, high-level features, while the DeepPlanktonNet trained from scratch extracts dataset-specific, fine-grained features critical for plankton classification. By combining these complementary feature representations, the proposed model simultaneously benefits from both transfer learning and domain-specific adaptability, hence achieving higher classification accuracy. While feature fusion automatically increases the dimensionality of the feature set, computational efficiency is ensured with the Whale Optimization Algorithm (WOA), where discriminative features and redundancy are pruned. In this manner, there is a balance between computational efficiency and enhanced performance. Besides, the proposed fusion approach maximizes generalizability, particularly in sparse-label scenarios. Since DeepPlanktonNet is trained from scratch, it adapts the unique distribution of small datasets and fills the robust feature extraction capability of InceptionResNetV2. Thus, the fusion approach remains effective even when there are few labeled training samples.

The Correlation Extended Serial Approach is employed to integrate extracted features while maintaining correlation strength. The mathematical formulation for this process remains as follows:

$$Correlation = \frac{\sum (R_i - \overline{R})(X_j - \overline{X})}{\sqrt{\sum (R_i - \overline{R})^2 \sum (X_j - \overline{X})^2}} \tag{1}$$

Where R_i and X_j represent features extracted from InceptionResNetV2 and DeepPlanktonNet, respectively, while $\overline{R}$ and $\overline{X}$ are their mean values.

This process involves assigning features with a correlation value of 0 or −1 to vector V_{c4}, while features with a positive correlation (+1) are added to a new vector, V_{c3}. Next, the mean value of V_{c4} is calculated as follows:

$$CT = \begin{cases} Vec_{upd} \; V_{c4} \geq 0 \\ Ignore_{feat} \quad V_{c4} \langle 0 \end{cases} \tag{2}$$

The fusion of both vectors is performed using the following mathematical formulation:

$$Vec_{Final} = \begin{pmatrix} Vec_{upd} \\ V_{c3} \end{pmatrix} \tag{3}$$

where Vec_{Final} represents the final fused vector, consisting of 36 optimized features. By leveraging feature correlation and optimization techniques, the proposed approach ensures high accuracy while maintaining computational efficiency.

3.6. Feature optimization using whale optimizer

Feature selection is a key step in machine learning pipelines, particularly for high-dimensional data, as it reduces computational expenses, improves model accuracy, and increases interpretability by removing redundant and non-informative features. In the current study, we apply the Whale Optimization Algorithm (WOA) [40] to optimize feature space and further boost plankton classification accuracy. WOA was used over other metaheuristic optimization algorithms such as the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) since it possesses high exploration-exploitation trade-off, low computation expense, and versatility. Crossover and mutation-based GAs are computationally costly and tend to converge prematurely towards solutions with low quality. PSO involving an update of particles based on acceleration and velocity also tends to converge prematurely towards poor-quality solutions in addressing the large-sized feature space. Contrarily, WOA dynamically adapts its search behavior according to decreasing encircling and spiral movement mechanisms to obtain a more convergent path and better feature subset selection. It is this advantage that makes WOA most efficient in plankton image classification, where precision in determining detailed visual patterns is key to accuracy. The WOA algorithm follows three key phases. In the Exploration Phase, multiple whales (candidate solutions) are randomly distributed across the feature space to ensure diverse sampling. The following equation mathematically represents this phase:

$$D = |EZ^*(j) - Z(j)| \tag{4}$$

Here, D is the distance between the current whale's position Z(j) and the best-known position Z*(j), where Z*(j) represents the whale's position relative to the best-known food source at iteration *j*.

In the Bubble-Net Hunting Phase, two mechanisms, shrinking encircling and spiral motion, are employed. The shrinking encircling mechanism gradually narrows the search space, refining the selection process. At the same time, the spiral motion ensures diverse search patterns and prevents the algorithm from getting trapped in local optima. This phase is mathematically described by:

$$Z(j+1) = Z^*(i) - CD \tag{5}$$

Here, Z(j+1) is the updated whale position at iteration j+1, and C is a coefficient that controls the update magnitude. The coefficients C and E are defined as:

$$C = 2ct_1 - c \tag{6}$$

$$E = 2t_2 \tag{7}$$

where t_1 and t_2 are auxiliary parameters, with $t_1 = 0$ and $t_2 = 1$ The convergence factor c, which governs the algorithm's transition from exploration to exploitation, is calculated as:

$$C = 2 - 2\frac{J}{J_{Max}} \tag{8}$$

Here, J is the current iteration, and J_{Max} is the total number of iterations.

The shrinking encircling behavior is further elaborated by:

$$Z(j+1) = Z^*(j) - F_q e^{cl} \cos(2\pi l) \tag{9}$$

in this equation, l is a random number between 0 and 1, and F_q represents the distance between the current whale and the food source. The whale reduces this distance by shrinking the circle around the food source using the spiral updating mechanism. The likelihood of this strategy is calculated as:

$$Z(j+1) = \begin{cases} Z^*(i) - CD \quad prob < P_b \\ Z^*(j) - F_q e^{cl}\cos(2\pi l) \quad prob \geq P_b \end{cases} \tag{10}$$

in the initial phase, whales explore a wide search area, adjusting their positions randomly to locate potential food sources. This exploration is represented by:

$$F = EZ_{rand} - Z(j) \tag{11}$$

$$Z(j+1) = Z_{rand} - CF \tag{12}$$

Where Z_{rand} is the random position of a whale, and F is the current location's fitness value. Finally, during the Exploration Phase, the algorithm seeks to exploit the most potential solutions found in the exploration phase. The algorithm decides the optimal feature subsets based on their contribution to classification accuracy while maintaining an appropriate balance between feature reduction and model performance.

Initially, the feature fusion process generated 36 features of InceptionResNetV2 and DeepPlanktonNet. When using WOA-based feature optimization, features were minimized to 22, which represents a reduction in the dimensionality of 38.9 %. This not only enhances computational efficiency but also relieves the curse of dimensionality, leading to improved generalization and accuracy in classification. By removing unnecessary and less descriptive features, the new method ensures that only the most significant traits are utilized in classification, with high accuracy maintained and computation costs reduced.

Even after feature selection, the model interpretability is not lost. The feature set is narrowed down to the extent that only the most discriminative and relevant attributes are utilized, and hence, LIME explanations remain interpretable. This enhances interpretability and reliability, and the model becomes more transparent and suitable for tracking marine ecosystems. The improved stability of LIME visualizations ensures that environmental professionals can trust and understand the model's decision-making process, which is crucial for applications in environmental science and the conservation of marine biodiversity. With the inclusion of WOA, the proposed approach gets an appropriate balance between feature reduction, computation, and classification accuracy, making it a robust choice for automated plankton classification and ecological studies.

3.7. Classification models

The two major approaches followed for classification in this work are an end-to-end deep learning model and machine learning classifiers on fused feature sets. Each of these is discussed in detail below.

3.7.1. End-to-end deep learning models

This approach entails directly training deep learning models on raw input images. Two architectures were used for this:

3.7.1.1. InceptionResNetV2. The InceptionResNetV2 model was used as an end-to-end classifier, a state-of-the-art convolutional neural network. This architecture enables the network to learn rich hierarchical representations of image data. The model is trained directly on raw input images. Each image went through several convolutional layers, batch normalization layers, pooling layers, and activation functions. Outputs from the network are fed to a softmax classifier that converts the logits to class probabilities, thereby giving the predicted class label for every input image.

It was fine-tuned on the target dataset with pre-trained weights from ImageNet. This end-to-end design ensures the network can automatically learn discriminative features optimized for the classification task without manual feature engineering.

3.7.1.2. DeepPlanktonNet End-to-End Model. A DeepPlanktonNet deep learning model has also been developed as an end-to-end classifier. Like InceptionResNetV2, the model is trained from scratch with raw input images. The DeepPlanktonNet model used convolution layers for feature extraction and a set of fully connected layers that map extracted features into class probabilities; classification is given by the softmax classifier at the last layer.

The different architectures of the neural networks and conventional machine learning classifiers form part of the classifier repertoire on the fused feature set. Each classifier shall be trained using suitable hyperparameters tuned to yield performance; cross-validation shall be carried out on these models to ensure that they generalize well and are, therefore, robust.

3.7.2. Machine learning classifiers

Narrow Neural Network (NNN): This is a simple feedforward neural network with a single hidden layer that contains only a few neurons. It is computationally efficient and works well for problems that are not too complex. However, these networks, though simpler and require less computational power, might fail to understand complicated links in the data due to their limited representation capability for nuanced characteristics.

Medium Neural Network (MNN): In a medium neural network, there is a balance in its architecture: neither few nor many neurons per hidden layer. That kind of network is usually tried for modeling most data that requires some degree of complexity. Compared to a narrow network, it may have one or two hidden layers with huge numbers of neurons inside them, which would enhance feature extraction and generalization.

Wide Neural Network (WNN): The feed-forward neural network will have one hidden layer, but it contains more neurons compared to a narrow network. Therefore, it has the ability to learn complex patterns; at the same time, the risk of overfitting, particularly when an input is sparse or noisy.

Bi-Layered Neural Network (BNN): It includes two hidden layers. This provides another degree of abstraction and brings increased capability of the network for modeling complex relationships within the data. Normally adopted in simple linear regression or binary classification problems, it is often simple in design. The usefulness of the model is limited because buried layers are absent; it cannot identify complex patterns.

Tri-Layered Neural Network (TNN): Three hidden layers, this neural network architecture will be much deeper with enhanced representational power. Because of the hiding layers, the model might learn nonlinear correlations in the data. This architecture is suitable for applications with nonlinear decision boundaries and provides more flexibility than the bi-layered network. In fact, this version of architecture works perfectly when features of datasets are high-dimensional.

Decision Trees: Nonparametric machine learning algorithms recursively partition the feature space into smaller features. This algorithm successively splits the data based on feature thresholds established by the tree, which is easy to interpret but allows overfitting when it becomes too deep or complex.

Linear Discriminant Analysis (LDA): LDA basically works on seeking a linear combination of features such that several classes are to be kept apart as far as possible. It assumes the data in each class to be normally distributed and identical covariance matrices. This is very powerful in feature extraction and classification problems, as it does feature dimensionality reduction by projecting data into lower-dimensional space that retains the largest class separability.

Support Vector Machine (SVM): A Support Vector Machine is sometimes known as a concept in machine learning for supervised regression and classification techniques. It finds the best hyperplane to represent to represent the data for classification into different classes. This model used the kernel function since the data represented was nonlinear in nature; hence, mapping such data into higher dimensions was possible.

Fine K-Nearest Neighbor (FKNN): F-KNN is the modification of well-known KNN algorithm, which is an improved variant by which classification of a sample in the feature space relies on the majority of its k-nearest neighbors. Thus, a "fine" version of KNN is sensitive to local patterns within the data due to small values of k.

3.8. LIME explainability

This paper introduces a new hybrid deep learning-based framework, DeepPlanktonNet, for detecting microalgae. Though DL has performed very promising predictive performance in the microalgae analysis, due to its intrinsic complexity, in most cases, the logic behind the predictions remains hard to explain. Many considered effective, black-box CNNs challenging due to the lack of transparency. Because these models are opaque, no field expert can fully trust or understand their decision-making processes; thus, the complete acceptability of such models within the operational field is difficult. This problem is solved using explainable artificial intelligence (XAI). XAI involves making complex AI systems transparent and interpretable. This study uses the LIME approach [41], a fundamental part of XAI, to close the interpretability gap and raise the proposed model's credibility. LIME provides a significant benefit by offering localized, intelligible explanations for predictions generated by complex machine-learning models, such as black-box CNNs. LIME has helped provide insight into important portions of the microalgae images that provided certain decisions of the DL model. It makes the model more interpretable, and field practitioners can be informed in detail about inner mechanics.

The advantages are that XAI improves model reliability and investigates complex systems, making this technology so important. Among Grad-CAM and occlusion, LIME had been chosen due to its better localization, accuracy, and model-agnostic nature [36]. This permits a range of inputs to be produced in perturbed samples using LIME, reflecting nuances in DeepPlanktonNet decision limits. This is further supported by the experimental results, which reveal how LIME explains Plankton classification easily and understandably. We improve overall interpretability in the model's behavior while analyzing microalgae images by carefully integrating LIME into other methods. Fig. 6: LIME-generated gradient maps using the jet color scheme visually highlight different degrees of relevance within microalgae images. These renderings use stronger, warmer tones, such as red and orange, to indicate more critical sites that are useful in guiding predictions by the model.

Cooler colors, for instance, blue, depict regions of low significance. Microalgae experts can locate high-value locations within the microalgae samples since these maps reveal an explicit visible image of where the model pays the most attention while classifying them. These images give a clear window into the model's decision-making process through visual highlighting of highly relevant areas in microalgae culture created by contrasting the actual scan with the model assessment. This approach has helped field practitioners in their diagnostic procedures and further enhanced the dependability of the model concerning the identification of microalgae.

Algorithm: Plankton Image Classification Framework.
Input: Plankton image dataset D.
Output: Classified plankton species with explanations.

Step 1: Pre-Processing
- 1.1 Resize input images $I \in D$ to standard dimensions (H, W)
- 1.2 Apply image augmentation (rotation R, flipping F, etc.)
- 1.3 Partition dataset D into training set D_train, validation set D_val, and test set D_test

Step 2: Deep Feature Extraction
- 2.1 Extract feature set F1 using InceptionResNetV2:
  F1 = InceptionResNetV2(D_train)
- 2.2 Extract feature set F2 using DeepPlanktonNet:
  F2 = DeepPlanktonNet(D_train)

Step 3: Feature Fusion
- 3.1 Concatenate feature vectors F1 and F2 to form F_combined:
  F_combined = F1 ⊕ F2

Step 4: Feature Selection using Whale Optimization Algorithm (WOA)
- 4.1 Initialize WOA parameters: population size N, iterations T, search agents S
- 4.2 Evaluate feature subsets Si and select the optimal feature set F_selected:
  F_selected = WOA(F_combined)

Step 5: Classification
- 5.1 Train classifiers (NNN, WNN, BNN, TNN, DT, LDA, KNN, SVM) using F_selected
- 5.2 Perform classification on test dataset D_test:
  y_pred = Classifier(F_selected)
- 5.3 Evaluate classification performance (Accuracy, F1-score, Precision, Recall)

Step 6: Model Training Optimization
- 6.1 Use Grid Search for hyperparameter tuning:
  θ_opt = GridSearch(θ)
- 6.2 Train optimized models InceptionResNetV2 and DeepPlanktonNet using θ_opt

Step 7: Model Explanation using LIME
- 7.1 Apply LIME to generate heatmaps for interpretability:
  LIME(y_pred) → heatmaps
- 7.2 Visualize important image regions contributing to classification

Return: Classification results with model interpretability

4. Results

This section provides a detailed overview of the study dataset used for the fine-grained plankton classification experiments, the proposed hybrid model, and its hyperparameter setting. It is further accompanied by an in-depth result analysis from several experiments conducted to test the efficacy of the model in question. The experimental settings have included details about the training aspects of the proposed hybrid model, together with the software platform used during this investigation.

4.1. Hyperparameters

Hyperparameter optimization is a crucial factor in impacting the performance of deep learning models [42]. For this research, a grid search and 5-fold cross-validation over the training data were used to learn two of the most influential hyperparameters: learning rate and epochs. Final testing was conducted only on the test set to avoid any biased performance measurement. The grid search was conducted by testing learning rates of 0.1, 0.01, and 0.001 and epochs of 5, 10, and 15. All the other hyperparameters were set manually using either empirical knowledge or default values to balance computational efficiency. The 5-fold cross-validation method helped prevent overfitting and ensure that selected hyperparameters would generalize well to new data. By strictly following the procedure of tuning hyperparameters in cross-validation over the training data, we preserve the integrity of our evaluation procedure and allow for an unprejudiced comparison among different setups. The final hyperparameters chosen, listed in Table 3, were selected considering both maximization of performance in the validation folds and achieving computationally viable possibilities.

Table 3
Final hyperparameters of inceptionresnetv2 and DeepPlanktonNet.

Parameter	Value
Activation function	softmax
Maximum Epochs	10
k-fold cross-validation	5-fold
Shuffle	Every Epoch
Optimization algorithm	Grid search
Learning rate	0.001
Validation frequency	30
Train size	0.9
Test size	0.1
Iteration per epoch	16

4.2. Experimental setting

The experiments were conducted using a computer system with the specifications listed in Table 4. A five-fold cross-validation technique was used to segment the dataset for a rigorous performance assessment, ensuring that training and validation subsets were consistently rotated over all folds in every experimental iteration.

4.3. Evaluation metrics

In this study, we employed accuracy, precision, recall, and F1-score to assess the performance of all deep neural networks. All the performance metrics were computed as follows:

$$Accuracy = \frac{TP + TN}{TS} \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

TN stands for true negative, TP for true positive, FP for false positive, FN for false negative, and TS denotes the whole number of samples.

4.4. Performance evaluation

Evaluating a machine learning model's performance on a task is known as performance evaluation. To make whether the model is accurate, dependable, and has good generalization to new, unseen data, it is an essential phase in the model-development lifecycle. These experiments aim to assess the performance of the DeepPlanktonNet deep learning framework. For these experiments, we used 6598 images from the WHOI dataset. This dataset consists of 22 classes, and we have 300 images in each class, except the "detritus" class, which has 298 images. The following experiments are performed in this experiment.

4.4.1. End-to-end classification using DeepPlanktonNet and inceptionresnetv2

The results of the end-to-end classification models, shown in Table 5, highlight the superior performance of the InceptionResNetV2 model compared to the DeepPlanktonNet model. InceptionResNetV2 achieved an accuracy of 92.33 %, a precision and recall of 90 % each, and an F1-score of 89 %. This strong performance can be attributed to its sophisticated architecture, combining multiple Inception modules for multi-scale feature extraction and residual connections that effectively support gradient flow.

Table 4
Details of experimental setup.

S. NO	Hardware Name	Specification
1	CPU of Computer system	Intel(R) Core(TM) i5-8365U CPU @ 1.60 GHz 1.90 GHz
2	RAM	16 GB
3	SSD	256 GB
4	Implementation tool	MATLAB R2024a- Academic use
5	Window	Windows 11 Pro

Table 5
Performance comparison of both end-to-end Inceptionresnetv2 and DeepPlanktonNet model.

Model	Accuracy	Precision	Recall	F1-Score
End-to-end Inceptionresnetv2	92.328	90	90	89
End-to-end DeepPlanktonNet model	79.21	76	76	23

In contrast, the DeepPlanktonNet model exhibited comparatively lower performance, with an accuracy of 79.21 % and a precision and recall of 76 % each. Notably, its F1-score was significantly lower at 23 %, indicating a lack of balance between precision and recall. While the DeepPlanktonNet model's higher precision demonstrates a tendency to minimize false positives, its simpler architecture constrained its ability to capture complex patterns, especially compared to the pre-trained InceptionResNetV2.

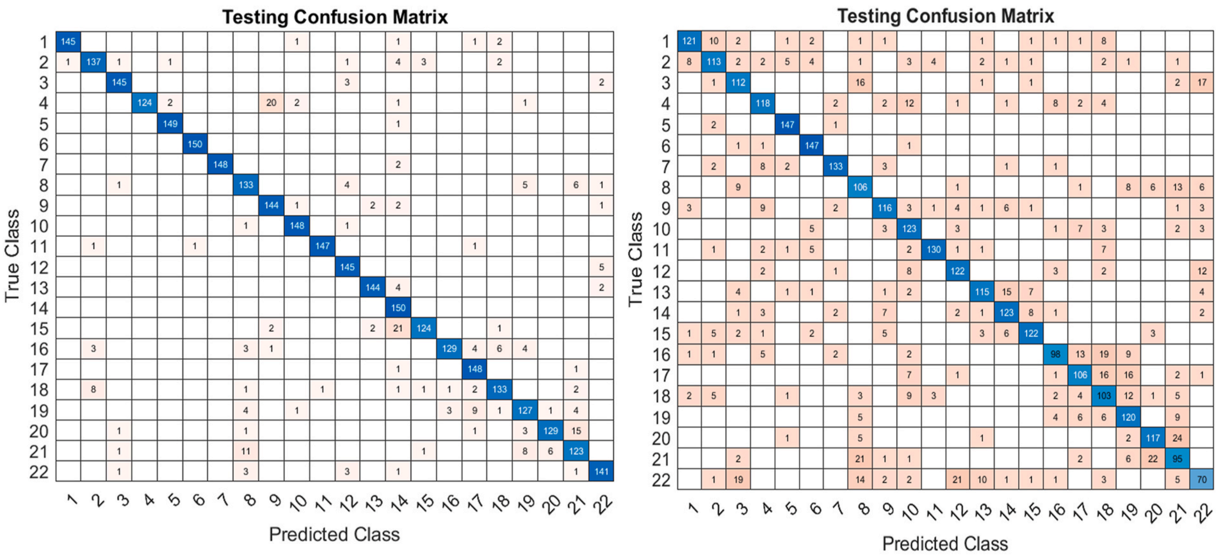
While these end-to-end models provide valuable insights, the limitations of relying solely on these approaches became evident. The suboptimal performance of the DeepPlanktonNet model and the inherent constraints of individual models motivated the use of a hybrid approach involving feature extraction and concatenation. By extracting features from both models and combining them into a single concatenated feature set, we aimed to leverage the strengths of both architectures. This merged feature set is then used to train machine learning classifiers to enhance performance.

By managing differences effectively between features and classifiers within this hybrid capacity, the improvements in representation help ensure computational efficiency, flexible responsibilities, and robustness of classification. Fig. 4 shows the testing confusion matrices for both models inceptionresnetv2 and DeepPlanktonNet. As discussed in the following sections, this approach demonstrates the potential for significant performance gains compared to using each end-to-end model independently.

4.4.2. Results for machine learning classifiers on fused feature set

As shown in Table 6, applying machine learning classifiers to the fused feature set makes significant improvements in classification performance compared to DeepPlanktonNet end-to-end models. Classifiers can leverage the complementarities between features learned from InceptionResNetV2 and DeepPlanktonNet using the fused feature set, resulting in consistent performance across a range of evaluation metrics. One of the key aspects of this study is the impact of feature selection using the Whale Optimization Algorithm (WOA) on computational efficiency. Table 6 presents the pre- and post-feature selection execution times comparison, the effect of dimensionality reduction on training and inference times. There is a significant decrease in computation time for all the classifiers except the SVM model, which supports the advantage of optimized feature selection in reducing computation overhead. The optimization procedure brings significant computational efficiency savings, particularly for decision trees, neural networks, and KNN classifiers, with modest classification performance losses.

The classifiers from the neural network show significant improvement in the execution efficiency after applying WOA-based feature selection. The optimal balance between classification performance and computational expense is achieved by the Medium Neural Network (MNN), which calculates an F1-score of 0.94 and an accuracy of 96.4 % while reducing the execution time to 453.94 s from 522.49 s, a reduction of 13.12 %. The Wide Neural Network (WNN) has the best accuracy of



Confusion Matrix for Inceptionresnetv2 Confusion Matrix for DeepPlanktonNet

Fig. 4. Comparison of Testing Confusion Matrices for InceptionResNetV2 and DeepPlanktonNet. The left matrix represents the performance of InceptionResNetV2, showing strong diagonal dominance indicating high classification accuracy. The right matrix represents DeepPlanktonNet, displaying more misclassifications, as evident from the off-diagonal values.

Table 6
Time comparison of the optimized and non-optimized fused feature.

Classifier name	Without Optimization Time on fused feature (sec)	With Optimization time of fused feature (sec)
Narrow neural network	495.23	428.15
Medium Neural Network	522.49	453.94
Wide Neural Network	568.91	493.52
Bilayered Neural Network	596.4	523.64
Trilayered Neural Network	692.54	579.96
Decision tree (Medium tree)	50.91	38.97
Linear discriminant	46.05	41.78
SVM (Linear Gaussian)	165.07	332.40
KNN	326.05	303.28

the neural network classifiers, which is 96.6 %, yet its execution time is longer, although the optimization reduces it from 568.91 s to 493.52 s with a 13.27 % improvement. Similarly, the Narrow Neural Network (NNN) is good, with an accuracy of 95.8 %, and its execution time is improved by 13.53 % from 495.23 s to 428.15 s. The Bi-layered Neural Network (BNN) and Tri-layered Neural Network (TNN) are also enhanced, with execution times lowered from 596.4 s to 523.64 s and from 692.54 s to 579.96 s, respectively. These reductions in execution time confirm that reducing input feature dimensionality has a beneficial effect on computational efficiency for deep learning models.

Conventional classifiers also show enhanced computational efficiency after optimization. The Decision Tree (Medium Tree) has the highest computational gain, with a decrease in execution time by 23.4 %, from 50.91 s to 38.97 s, and maintaining the same accuracy of 62.4 %. The Linear Discriminant Analysis (LDA) achieves the highest classification accuracy among all classifiers, 97.9 %, and displays a decrease in execution time from 46.05 s to 41.78 s, a decrease of 9.29 %. This corroborates the efficiency of discriminant-based methods in handling optimized sets of features. The K-Nearest Neighbor (KNN) classifier also benefits from the smaller feature space, as its execution time reduces from 326.05 s to 303.28 s, with a 6.99 % improvement, demonstrating that fewer input features enhance nearest-neighbor searches and speed up classification.

However, a contrasting trend is observed with the Support Vector Machine (SVM) classifier and the Linear Gaussian kernel, which exhibits an increase in run time from 165.07 s to 332.40 s, an increase of 101.3 %. This trend suggests that feature dimensionality reduction results in a shift in the data distribution, and therefore, in the optimization process of the SVM kernel transformations. Since SVM is based on the separability of hyperplane, the optimization procedure may involve extra computational work for tightening the newly formed boundaries of decision, thus having increased computational cost. For this exceptionality, feature selection has a global positive effect when it comes to lowering computational overhead, as clearly indicated in Table 6.

Table 7 also gives a comparative analysis of classifiers in terms of accuracy and computational speed, showing how feature optimization impacts classification accuracy and time execution. The experiment indicates how the use of WOA for feature selection enables classifiers to achieve high accuracy while optimizing for computational efficiency. Of the classifiers, LDA, MNN, WNN, and KNN show strong classification

Table 7
Detailed Comparison of Classifiers Based on Accuracy and Computational Efficiency before optimization.

Classifier	Accuracy (%)	Precision	Recall	F1 score	Comp. time (sec)
Narrow neural network	95.81	93.60	93.59	92.68	495.23
Medium Neural Network	96.42	94.15	94.11	93.44	522.49
Wide Neural Network	96.61	94.34	94.32	93.60	568.91
Bilayered Neural Network	95.01	92.71	92.61	91.89	596.4
Trilayered Neural Network	94.63	92.47	92.47	91.73	692.54
Decision tree (Medium tree)	62.42	59.69	69.01	58.58	50.91
Linear discriminant	97.91	95.68	95.72	94.89	46.05
SVM (Linear Gaussian)	96.92	94.67	94.68	93.89	165.07
KNN	96.74	94.36	94.36	93.59	326.05

performance and significant reductions in execution time, confirming that selection of the most significant features allows models to perform better. The stringent scrutiny of computational performance, as illustrated via Tables 6 and 7, reaffirms the relevance of feature selection in machine learning pipelines for plankton classification. By eliminating redundant features, WOA-based optimization ensures classifiers are more efficient, reducing training as well as inference times while preserving classification accuracy. These findings confirm the advantages of integrated feature selection methods towards improved computational efficiency for plankton classification on large scales.

4.4.3. Optimized fused features classification using ML

The experiment aimed to evaluate the performance of various machine learning classifiers on the feature fusion set, refined by the feature selection approach for handling the problem of redundancy and irrelevance. Feature fusion was extracted from the dimensionalities of InceptionResNetV2 and the DeepPlanktonNet model. Such challenges may involve redundant and irrelevant features that degrade classification performance and computation efficiency. Feature selection applied to the fused feature set overcomes these issues by enabling the selection of the most relevant features for classification. Then, the selected feature subset was fed into machine learning classifiers to find out the effect of the optimization of features on the performance of the classification. It was done so to improve the predictive accuracy with data complexity reduction and enhancement in computational efficiency. Fused features create a combined feature representation from both InceptionResNetV2 and the DeepPlanktonNet model to give a richer representation of the data. Optimizing involves a fine-tuning process for machine learning classifiers with respect to hyperparameters for improved performance. This experiment will, therefore, be comparing the different classification results with and without the application of WOA to see its impact on accuracy, precision, recall, and F1-score metrics.

The results, summarized in the second table, reveal a significant improvement in the performance of nearly all classifiers after the application of the Whale Optimization Algorithm. Accuracy for the Narrow Neural Network rose to 95.90 % from 95.81 %. Precision increased to 96.09 % from 93.60 %, recall from 93.59 % to 95.89 %, and an F1-score increased similarly from 92.68 % to 95.89 %, hence proving WOA indeed optimized this simpler neural network architecture. Similarly, the Medium Neural Network performed very well by increasing accuracy from 96.42 % to 97.72 %, precision from 94.15 % to 97.83 %, recall from 94.11 % to 97.72 %, and F1-score from 93.44 % to 97.78 %. In other words, it shows how this optimizer supports medium-sized architecture very well. On the Wide Neural Network (WNN), the improvement observed accuracy increased from 96.61 % to 97.88 %. Accuracy increased only from 94.34 % to 95.60 %, with a slight reduction in recall from 94.32 % to 94.80 %. Still, the increase reflects positively on the optimization. Results for the BNN were highly improved, as accuracy jumped from 95.01 % to 98.79 %, precision from 92.71 % to 98.81 %, recall from 92.61 % to 98.77 %, and F1-score from 91.89 % to 98.79 %. It proves that WOA is very effective for complex networks containing multiple layers.

In the case of TNN, accuracy increased from 94.63 % to 97.57 %. Similarly, precision, recall, and F1-score significantly increased from 92.47 %, 92.47 %, and 91.73 %–97.68 %, 97.57 %, and 97.62 %, respectively. This indicates that WOA efficiently tunes the hyperparameters of more complicated neural networks to result in superior classification performance. Decision Tree-Medium Tree showed the most improvement among all the models, where the accuracy for this model drastically increased from 62.42 % to 96.05 %. It improves accuracy from 59.69 % to 96.39 %, recalls from 69.01 % to 96.04 %, and the F1-score from 58.58 % to 96.21 %. The massive uplift in performance indeed shows that WOA has the ability to surmount inherent deficiencies of a decision tree like overfitting and suboptimal split point selection through optimization of the parameters. In the case of a Linear Discriminant classifier, WOA improved accuracy from 97.91 % to 98.18 %. Precision and recall also increased to 95.68 %, 95.72 %, and 94.89 % from 98.29 %, 98.17 %, and 98.23 %, respectively. After optimization, the accuracy of SVM (Linear Gaussian) improved considerably, increasing from 96.92 % to 97.72 %. The optimized value of precision, recall, and F1-score reached an almost perfect 99.69 % compared with 94.67 %, 94.68 %, and 93.89 %, respectively. The exceptional rise in the performance of SVM identifies the capability of WOA for fine-tuning kernel parameters along with regularization terms in the classification task. KNN classifier results show a marked increase, too, from an accuracy of 96.74 %–97.72 %, precision rising from 94.36 % to 97.80 %, recall rising from 94.36 % to 97.72 %, and F1-score rising from 93.59 % to 97.76 %. The improvements suggest that WOA successfully optimized the neighborhood size and distance metric, leading to better performance in KNN.

The results in Table 8 demonstrate that applying Whale Optimization to fused features significantly enhances the performance of all classifiers. The most pronounced improvement was observed for the Decision Tree and SVM classifiers, dramatically increasing their metrics post-optimization. Fig. 5 compares the accuracy of all classifiers with and without Whale optimization. Overall, WOA proved to be an effective tool for optimizing machine learning models, particularly for complex and multi-layered classifiers. By fine-tuning hyperparameters, the algorithm successfully improved the classifiers' accuracy, precision, recall, and F1 score, showcasing its utility in tasks involving fused feature sets.

4.4.4. Analyzing accuracy and computational efficiency trade-offs

To better illustrate the trade-off between model performance and computational expense, a heatmap visualization was employed, as in Fig. 6. The heatmap effectively demonstrates the impact of optimization on classification performance and computational cost. The outcomes indicate that all models improved in accuracy after optimization, with the Bilayered Neural Network having the highest accuracy of 98.79 %. Specifically, the Decision Tree improved most significantly, from 62.42 % to 96.05 %, which means that optimization significantly enhanced its predictive power. In computational efficiency, all models demonstrated reduced running time after optimization, which verifies the conclusion that the optimization process is effective. Nevertheless, the analysis also identifies trade-offs, such as the SVM model that had higher accuracy but with longer computational time 165.07 s prior to optimization vs. 332.40 s after optimization. Meanwhile, the Linear Discriminant Analysis and Decision Tree models remained the most computationally light, with computational times less than 50 s. The heatmap is a handy summary of the trade-offs in this work, providing great insight into selecting models under demands, either by trading accuracy for computational

Table 8
Quantitative assessment of classifiers post-optimization.

Classifier Name	Optimized Accuracy (%)	Precision	Recall	F1-score	Opt. model time (sec)
Narrow neural network	95.90	96.09	95.89	95.89	428.15
Medium Neural Network	97.72	97.83	97.72	97.78	453.94
Wide Neural Network	97.88	95.60	94.80	95.19	493.52
Bilayered Neural Network	98.79	98.81	98.77	98.79	523.64
Trilayered Neural Network	97.57	97.68	97.57	97.62	579.96
Decision tree (Medium tree)	96.05	96.39	96.04	96.21	38.97
Linear discriminant	98.18	98.29	98.17	98.23	41.78
SVM (Linear Gaussian)	97.47	94.13	94.13	97.42	332.40
KNN	97.72	97.80	97.72	97.76	303.28

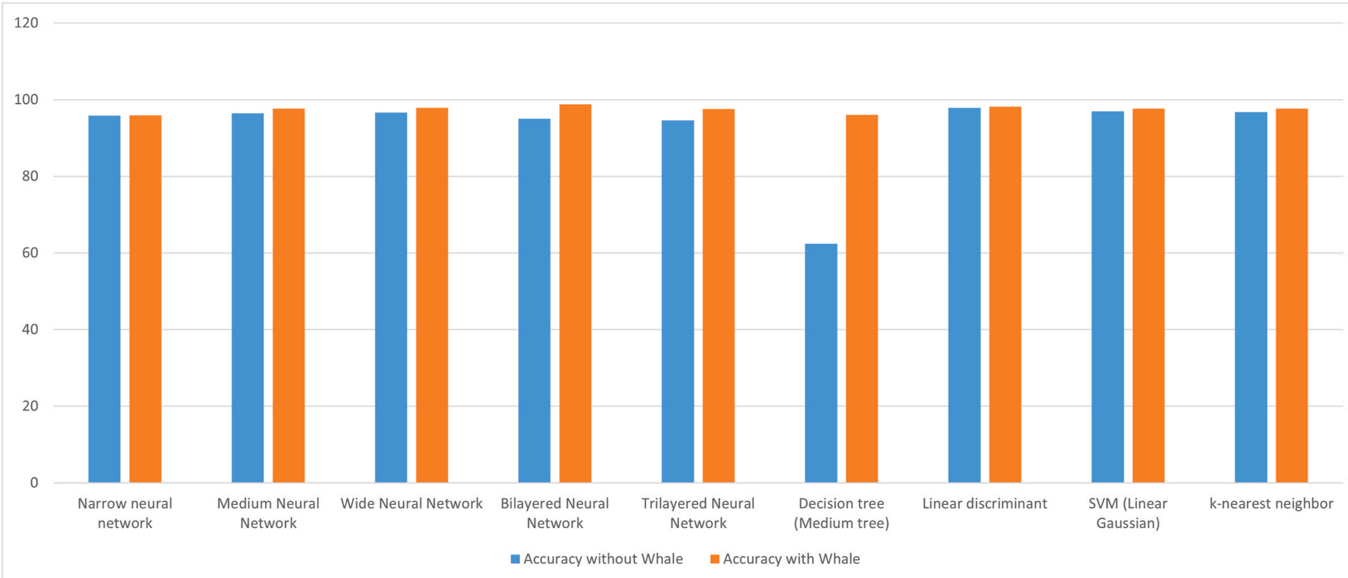


Fig. 5. Comparison of classification accuracies across different models with and without the 'Whale' class. The blue bars represent accuracy without the Whale class, while the orange bars indicate accuracy with the Whale class included. The accuracy drop in some models suggests that the Whale class introduces additional classification challenges. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

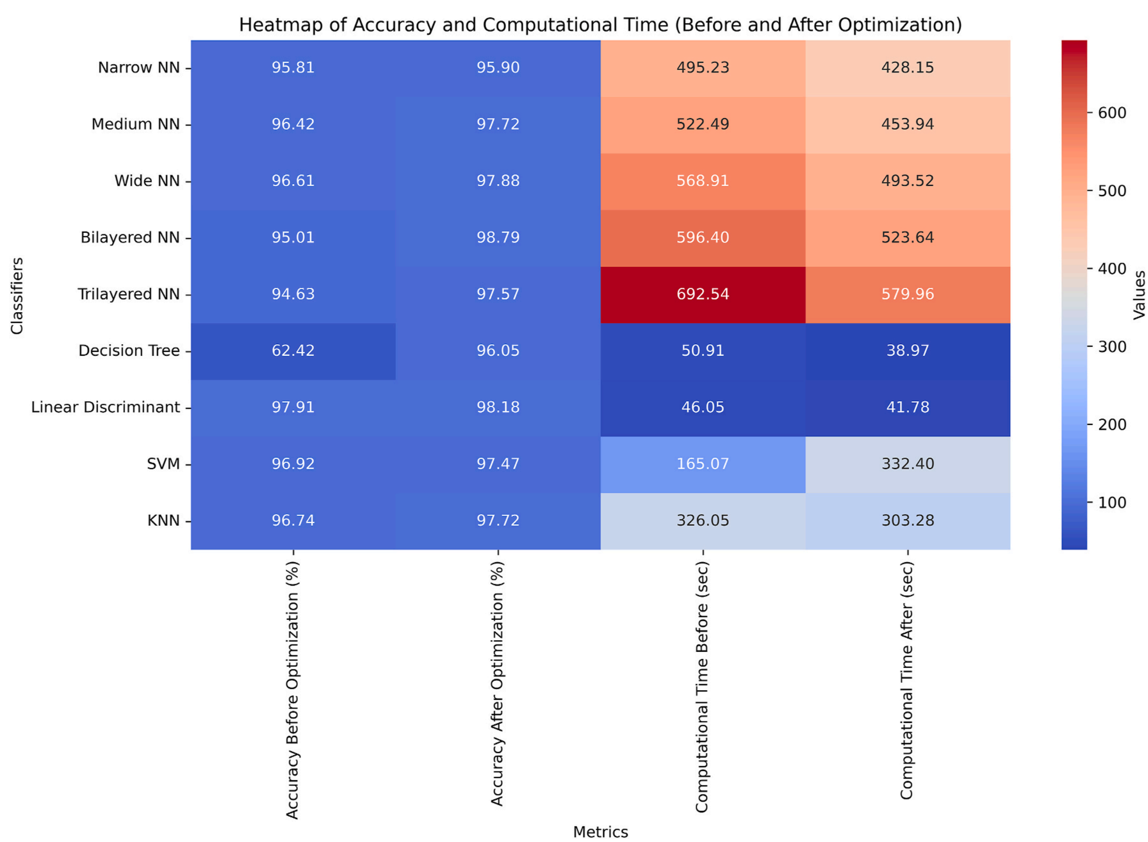


Fig. 6. Accuracy and computational time trade-offs pre and post-optimization.

speed or vice versa. The visualization also addresses the reviewer's concern head-on by adding to clarity beyond variation in model performance and informing classifier selection decisions.

4.4.5. Comparison with SOTA deep learning models

To better depict the comparison of performance, there is a need to frame the comparison contextually by putting significance on the most significant metric applied in measuring the performance of the models in comparison. One of the basic indicators of the general model's ability to predict instances correctly, the average accuracy, becomes critically relevant when there is a balanced dataset. But accuracy as a single-metric measure doesn't capture the full picture for a model in class-imbalanced scenarios; therefore, precision and recall are also equally important. Precision measures how many the model marks as true positives without marking false negatives as positives, which is very important in applications where false positive penalties could be severe. On the contrary, recall captures how accurately the model detects all positive samples, and it is of utmost importance if missing positive samples are costly or very undesirable. F1-score is the balanced average of recall and precision that provides in one measure how effectively the model is doing both ways, particularly in case of class imbalance. Architectures such as InceptionResNetV2, ShuffleNet, DenseNet201, and ResNet101 are strong models defined by their deep architecture and extended training on large datasets. They are very efficient in feature extraction and have been widely applied where high accuracy is required. However, they usually come at the expense of gigantic computational costs in training time and inference speed, which may be a drawback in resource-constrained environments.

The proposed approach includes InceptionResNetV2 and DeepPlanktonNet architectures since they were selected based on their ability to manage the peculiarity of the target task of classification efficiently. InceptionResNetV2 brings the best from Inception and ResNet architectures by proposing an efficient deep neural model that harmoniously strikes depth with computational practicability to manage high-accuracy tasks. A particular network, DeepPlanktonNet, improves the ability of the model to handle extremely complex datasets with high accuracy in plankton image classification. The hybrid model enhances the accuracy, precision, and recall without loss in the efficiency of the model at all times, delivering better outcomes at no loss to computational cost, as can be seen from Table 9. The proposed model goes on adding these strong models together to come up with an integral, productive, and extremely accurate solution and thus the best for the required task.

For more clarity, we utilized 5-fold cross-validation as a data split technique, decreasing overfitting and ensuring that reported performance metrics can be generalized. Under this practice, the available data is divided into five pieces, where all parts except one are used as a validation while the other one is used to train. Repeat this cycle five times to make sure there is at least one validation set per data point. The performance improvement, as evident from Tables 9 and is because of the synergistic strengths of InceptionResNetV2 and DeepPlanktonNet in handling complex datasets, thus providing greater accuracy, precision, recall, and F1-score than the individual models. The hybrid model ensures both model efficiency and improved classification results without any compromise on computational cost. Fig. 7 shows the comparison of confusion matrices for ShuffleNet, DenseNet201, and ResNet101 highlighting classification performance across 22 classes.

4.4.6. Lime explainability results

The instances shown in Figs. 8–10 represent three different classes: Ditylum, Licmophora, and Pleurosigma. They are analyzed using the LIME approach to interpret the predictions of two deep learning models: InceptionResNetV2 and DeepPlanktonNet. For the class Ditylum, the LIME visualizations produced by InceptionResNetV2 focused attention on distinctive subregions, especially the edge and central structures; this would propose that the network relies in classification on fine-grained texture and shape features. In contrast, DeepPlanktonNet has a much wider and distributed focus that might be viewed as generalized features extracted with lower interpretability since focused regions are missing.

Table 9
Performance metrics of pre-trained networks compared to the proposed approach.

Pre-Trained Networks	Avg. Accuracy (%)	Precision	Recall	F1-Score
Inceptionresnetv2	92.32	90.54	90.59	89.59
Shufflenet	90.81	89.46	90.30	89.00
DenseNet201	91.96	90.32	90.75	89.63
Resnet101	90.80	91.90	92.17	91.12
Proposed approach	**98.79**	**98.81**	**98.77**	**98.79**

InceptionResNetV2 highlights discriminative areas for the Licmophora class in the top-right and bottom-left of the structure by focusing on distinctive shape or texture features. Meanwhile, DeepPlanktonNet spreads attention over the plankton body, which suggests an overall feature extraction where the key regions may not have that much importance. For Pleurosigma, InceptionResNetV2 emphasizes compact and central regions that are most probably targeted for symmetry and other internal patterns characteristic of the species. DeepPlanktonNet, however, evidences a more general approach, focusing rather equally on the structure. Overall, InceptionResNetV2 produces sharper and more interpretable attributions, fitting applications requiring stronger explainability. At the same time, DeepPlanktonNet has a wide focus on its features, hence probably more suitable for generalization but at the cost of interpretability. The choice between which model to apply would, thus, depend upon how these explanations meet domain-specific insight and task needs.

4.4.7. Comparison with SOTA approaches

The proposed approach has demonstrated reflective and insightful results in plankton classification. The InceptionResNetV2 and DeepPlanktonNet CNN deep learning models and the machine learning classifiers from the fused features both with and without optimization exhibit significantly improved performance against state-of-the-art (SOTA) approaches. Among the current methods, the paper in Ref. [43] contains a well-formulated transfer learning system with a goal of striking a balance between the classification performance and computational complexity. This work investigates the impacts of in-domain and out-of-domain transfer learning by utilizing three pipelines: (1) in-domain pre-training, wherein models are first pre-trained on a large-scale plankton dataset and subsequently fine-tuned on a target dataset, (2) out-of-domain pre-training, wherein general-purpose large-scale datasets like ImageNet1K or ImageNet22K are utilized as the source domain for pre-training before fine-tuning is done on plankton datasets, and (3) two-stage fine-tuning, where both methods are synthesized by pre-training on ImageNet first and then fine-tuning on a large-scale plankton dataset before lastly fine-tuning on the target dataset. The experiments in Ref. [43] demonstrate that out-of-domain transfer learning with ImageNet22K pre-training works excellently compared to in-domain training, giving an average accuracy gain of around 6 %. It also compares different deep learning architectures and verifies that Vision Transformers (ViTs) and ConvNeXt models can achieve similar or superior performance compared to CNN ensembles without being computationally expensive. The use of ViTs and ConvNeXt in Ref. [43] reached 96.6 % accuracy, which is one of the best in the plankton classification field.

Although the method in Ref. [43] maximizes transfer learning methods and model choice, the developed framework in this paper maximizes classification accuracy even more by integrating feature fusion and optimization techniques. In particular, the InceptionResNetV2 and DeepPlanktonNet CNN combination yields a deep feature representation, and using the Whale Optimization Algorithm (WOA) for feature selection yields a significant accuracy of 98.79 %, higher than the 96.6 % accuracy in Ref. [43]. The most important causes of such enhancement are: (1) Feature Fusion, which enhances the representational power of the model by combining the power of two complementary networks and both local and global patterns more effectively representing them; (2) Feature Selection via WOA, which eliminates redundant information and allows classifiers to focus on the most significant features, and it increases accuracy as well as computation efficiency; and (3) Classifier Diversity, in which multiple classifiers in machine learning are explored to ensure that the optimized and fused features perform effectively with various learning paradigms. These advancements all combine to deliver top-of-the-line performance in plankton classification.

Table 10 shows a comparative study between our proposed method and existing SOTA methods. The results show that the earlier highest

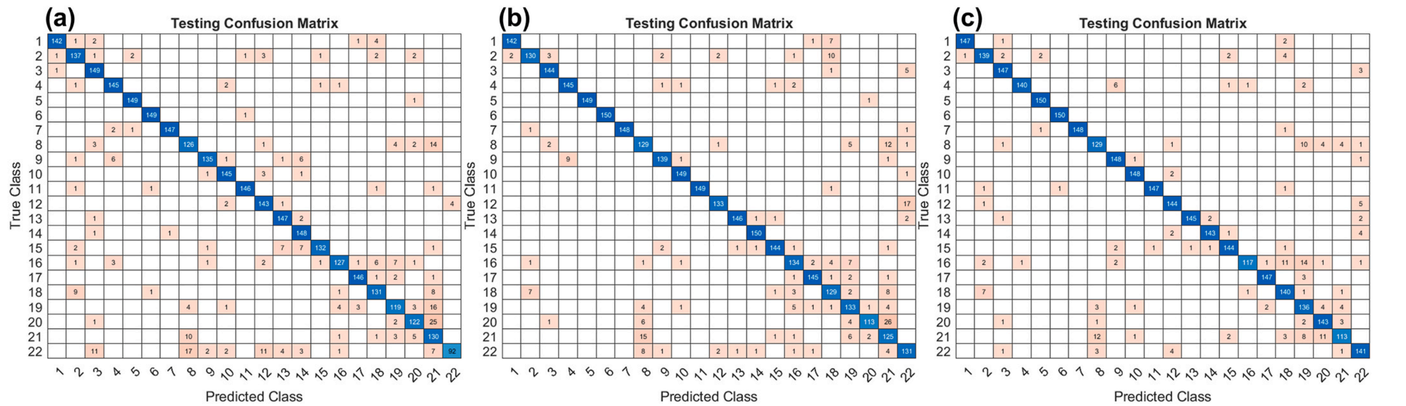


Fig. 7. Testing confusion matrices of state-of-the-art deep learning models (a) ShuffleNet, (b) DenseNet201, and (c) ResNet101 demonstrating their classification accuracy and error distribution across 22 target classes.

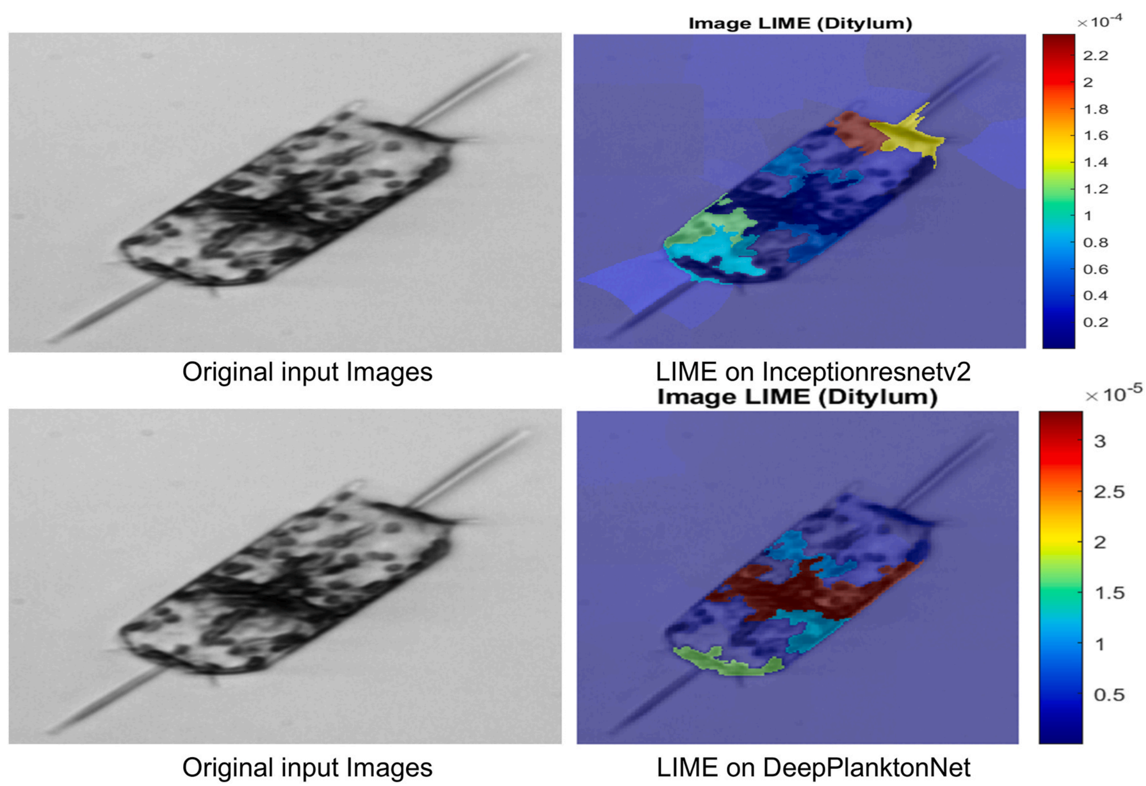


Fig. 8. LIME visualizations comparing InceptionResNetV2 and a DeepPlanktonNet on plankton classification. The left column shows the original input images, while the right column highlights important regions influencing model predictions using LIME. The color scale represents feature importance, with red indicating the most influential areas for *Ditylum* Class. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

reported accuracy was 97.9 % by a transfer learning ensemble method [44] and a custom CNN-based local and global feature extraction method [45] with an accuracy of 96.9 %. In the clustering-based approaches, the X-Mean clustering method [46] had an accuracy of 90.53 %, which was significantly lower than deep learning approaches. However, other models, such as ResNet18 with Adam optimization [47] and one-shot classification [37], performed far worse, with 72.9 % and 58.8 % accuracy, respectively.

In addition to high classification accuracy, another equally crucial consideration is the availability of interpretability in deep neural network models. Techniques like InceptionResNetV2 and DeepPlanktonNet are “black-box” models, and it is difficult to interpret their decisions. This consideration is particularly relevant to real-world applications where trust and accountability are necessary, e.g., healthcare [48] and finance [49]. To address this challenge, the proposed framework integrates Local Interpretable Model-Agnostic Explanations (LIME) to enable explainability. LIME provides a human-interpretable explanation of the model’s predictions by recognizing crucial input features that contribute to classification results. The benefits of integrating LIME include: (1) Explaining Model Decision-Making, as LIME illustrates which areas of input images contribute most to predictions; (2) Trusting Model Reliability, as it allows researchers to make sure that the model is grounded on interpretable features rather than unrelated correlations; and (3) Trust and Transparency Improvement, allowing for the deployment of AI-driven classification systems for applications in sensitive areas [50].

To further validate the integrity of our proposed approach, feature contributions will be rigorously verified by applying LIME to both

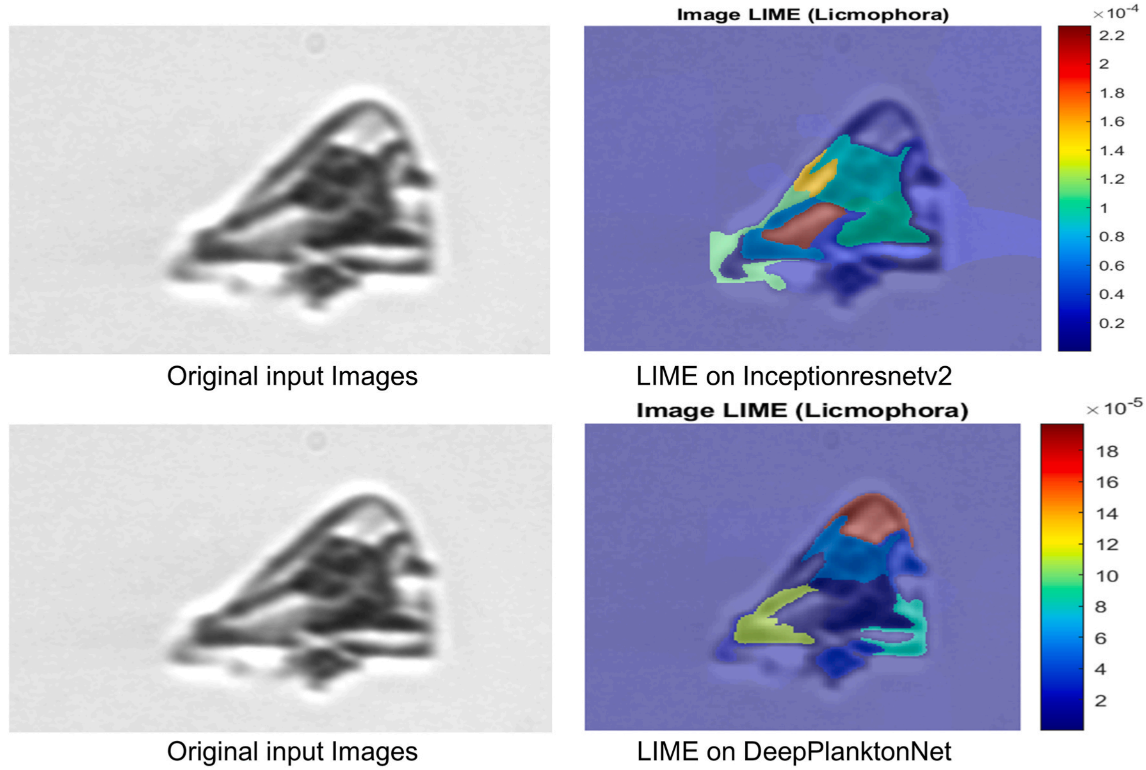


Fig. 9. LIME visualizations comparing InceptionResNetV2 and a DeepPlanktonNet on plankton classification. The left column shows the original input images, while the right column highlights important regions influencing model predictions using LIME. The color scale represents feature importance, with red indicating the most influential areas for *Licmophora* Class. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

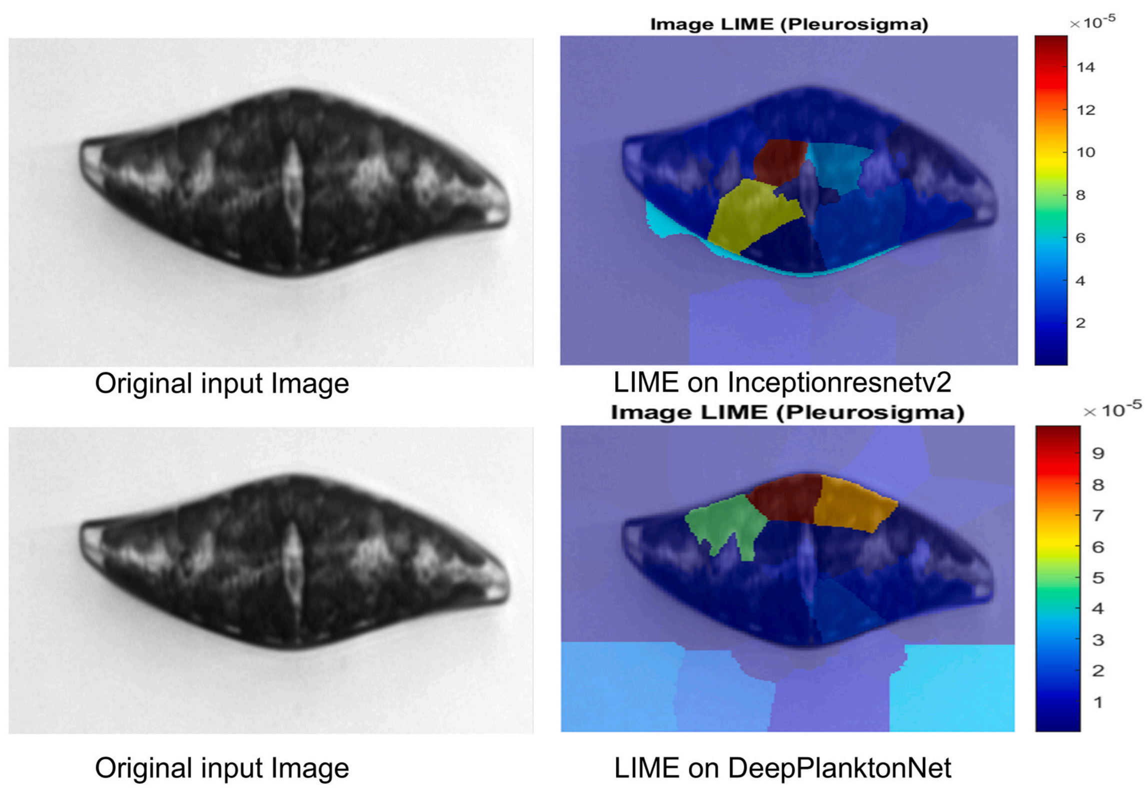


Fig. 10. LIME visualizations comparing InceptionResNetV2 and a DeepPlanktonNet on plankton classification. The left column shows the original input images, while the right column highlights important regions influencing model predictions using LIME. The color scale represents feature importance, with red indicating the most influential areas for *Pleurosigma* Class. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Table 10
Comparison of proposed framework with State-of-the-art (SOTA) approaches.

References	Year	Approach	Feature Optimization	XAI Method	Accuracy (%)
[43]	2023	TL Resnet50	–	–	96.6
[44]	2021	TL Ensemble	–	–	97.9
[37]	2020	One shot classification	–	–	58.8
[47]	2021	Resnet18	–	–	72.9
[46]	2021	X-Mean clustering	–	–	90.53
[45]	2017	Custom CNN local and global feature extraction	–	–	96.90
Proposed	**2025**	**DeepPlanktonNet + InceptionresnetV2**	**WOA**	**LIME**	**98.79**

DeepPlanktonNet and InceptionResNetV2 CNN. This step is critical to ensuring the excellent performance that will be demonstrated in our research is achieved without bias or overfitting to non-discriminative features. By leveraging explainable AI (XAI) techniques, our approach not only achieves state-of-the-art classification accuracy but also models fairness and transparency, rendering it ever more suitable for application in real-world scenarios.

5. Discussion

The proposed approach illustrates an excellent contribution to the literature of plankton image classification via the synergy between InnceptionResNetv2, DeepPlanktonNet architecture, feature fusion, optimization techniques, and explainability techniques. The end-to-end accuracy of the InceptionResNetV2 model exhibits its strength by achieving a splendid accuracy of 92.32 %. This verifies the efficacy of transfer learning with pre-trained models optimized over a domain-specific data set. Nevertheless, the DeepPlanktonNet model, which was trained from scratch, fared worse at 79.21 % accuracy. This difference illustrates the challenge of training models from scratch with sparse labeled data as they might not generalize as much as pre-trained models.

Feature fusion was conducted to overcome the shortcomings of the individual models by combining the extracted features of both InceptionResNetV2 and DeepPlanktonNet. It enhanced feature representation and, at the same time, also gave a remarkable enhancement in performance using machine learning classifiers. Among them, a linear discriminant and bilayered neural network yielded an accuracy of 98.79 % and 98.18 %, respectively. Integration of Whale Optimization Algorithm (WOA) further developed feature selection and also proved its utility to enhance accuracy and computational efficiency. This was realized in the accuracy of the bilayered neural network, which, when optimized, achieved an accuracy rate of 98.79 %, verifying a very high synergy between WOA and deep feature fusion. Moreover, WOA optimization realized considerable computation reduction overhead, as indicated in Tables 6 and 7, making the framework scalable to large applications.

Scalability in the deployment of machine learning models for extensive ecological studies is needed so that there is a demand for real-time or near-real-time classification when monitoring marine biodiversity. The described framework, being optimized using feature selection based on WOA, has considerable savings in computational costs with high-quality classification, as shown in the results. Real-time classification depends on the hardware specifications, however, as much as on data communication rate and parallelization of models. While the current strategy is adequate for batch processing in high-performance computing applications, further optimizations such as hardware acceleration and distributed computing need to be explored to enable real-time classification in field use. The execution of models on embedded edge devices equipped with specialized AI accelerators (e.g., NVIDIA Jetson or Coral Edge TPU) would also enable real-time classification and allow for large-scale automated monitoring in marine ecosystems.

While edge AI accelerators and high-performance computing can enhance real-time classification, federated learning offers another promising avenue for scalability enhancement. Traditional cloud-based AI models entail continuous data uploading to central servers, which can be unfeasible for large-scale ecological research in remote areas. Federated learning offers a decentralized solution through the ability of models being trained on multiple edge devices directly without sharing raw data. This not only ensures data privacy but also facilitates unbroken learning from diverse environmental settings. Through the integration of federated learning into the proposed plankton classification framework, local ocean research stations can train models on local datasets, allowing for greater responsiveness to varying ecological conditions. Moreover, this decentralized framework reduces computational and bandwidth overheads involved in cloud-based model updates, a factor that makes it an ideal solution for mass-scale marine biodiversity monitoring. Subsequent studies should be aimed at creating federated learning approaches specific to ecological use cases, solving problems like communication-efficient model updates, heterogeneous data distribution, and security issues due to adversarial attacks in decentralized learning settings.

LIME is the requisite layer of explainability given to what is generally referred to as the Achilles' heel of deep learning models: "black boxes." LIME's feature contribution visualization enabled one to understand why the model made a particular prediction. This interpretability is sorely required in plankton classification since explainable outputs are required for ecological insights and decision-making in ocean science. Moreover, the ablation experiments with other pre-trained networks, such as DenseNet and ResNet101, also highlight the accuracy and computational cost superiority of the proposed approach.

5.1. Limitations of the proposed approach

Even with these encouraging results, there are certain limitations. DeepPlanktonNet's performance was not as good as pre-trained models, and the highest performance was observed when larger labeled datasets were utilized. Apart from that, even with the remarkable improvements by the fusion feature set and WOA optimization, the computational cost of the optimization process was not overlooked and reflects the need to advance optimization algorithms further. A second limitation of this research might be the intrinsic dependence on an open-source dataset that potentially does not capture the plankton species variation over different types of water bodies. Additionally, one of the biggest limitations is from a feature importance analysis in the traditional sense. Since deep learning models learn high-dimensional abstract features rather than human-understandable attributes, ranking the feature importance of each is challenging. Although we utilize Local Interpretable Model-agnostic Explanations (LIME) to highlight important image regions for classification, it is still difficult to measure how well these explanations align with domain knowledge. In contrast to table data with structure where feature importance can be immediately assessed, image classification based on deep learning draws on abstract representation that may or may not have direct equivalence with known predefined expert biological features. To date, there are no standard interpretability metrics for plankton classification against which alignment to domain knowledge is measurable.

5.2. Future work

To address this gap, our future work will explore cooperative collaborations with marine biologists to generate benchmark datasets with expert annotation of biologically informative image regions. We also intend to add attention mechanisms and other explainability techniques, such as SHapley Additive Explanations (SHAP) and neuron attribution methods, to boost interpretability further. These methods will measure to what degree the model's decision-making process aligns with expert knowledge, making deep learning more credible and useable in marine ecology. We also proposed some other future directions for this study, include adding domain adaptation techniques to generalize the model to unseen datasets with minimal fine-tuning.

Exploring self-supervised or semi-supervised learning techniques may also solve the issue of small labeled datasets better by utilizing unlabeled data. Also, coupling WOA with other metaheuristic techniques and hybrid optimization techniques may reduce computational cost without compromising on high performance. Incorporating temporal and spatial data into the LIME framework for plankton research may also enhance its application in dynamic ocean ecosystems. In addition, the application of federated learning techniques would further increase scalability and flexibility so that models could learn distributed data without compromising data privacy. Lastly, real-time implementation of the proposed methodology in the automatic monitoring of aquatic ecosystems would be an intriguing area for future research. Therefore, the methods presented here take an unprecedented step toward realizing state-of-the-art performance, computational efficiency, and explainability in plankton classification. The methodology can be widely applied to broader usage in environmental monitoring and marine biology due to the potential removal of the demerits mentioned and identification of directions for the future.

6. Conclusion

Plankton is a critical element of marine ecosystems and a valuable bioindicator of ecosystem health. However, traditional plankton taxonomy procedures remain time-consuming, labor-intensive, and prone to human error. In this paper, this work exploits recent advances in deep learning and optimization algorithms to propose a lightweight and interpretable automated plankton classification system. From the WHOI dataset, we explore InceptionResNetV2 and DeepPlanktonNet with feature fusion for better representation and Whale Optimization Algorithm (WOA) for feature selection to enhance accuracy and computation, and then classification of the optimized features by different machine learning classifiers and use of the LIME explainability technique to enhance model interpretability. The framework developed achieved a classification accuracy of 98.79 %, outperforming earlier state-of-the-art solutions.

In addition to accuracy, real-time deployability is a critical factor for large-scale marine research. The method proposed has the advantage of WOA-driven feature selection, which lowers computational complexity and thus makes it more suitable for real-time deployment. Yet, additional optimizations like model compression and edge computing deployment would be required to realize large-scale, real-time classification.

Subsequent work will explore feature selection using reinforcement learning to enhance classification methods adaptively. Data augmentation using generative AI models such as GANs is also feasible to address class imbalance and increase model generalizability even further.

The proposed framework has enormous real-world applications, like ecological surveys, climate change monitoring, and water quality analysis. By incorporating autonomous underwater vehicles (AUVs) and remote sensing technologies, it can be employed to perform large-scale automated monitoring of marine ecosystems. In addition to enhancing plankton classification, this research lays a foundation for further innovation in environmental AI applications.

CRediT authorship contribution statement

Muhammad Hassan: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Giovanna Salbitani:** Writing – review & editing, Validation, Supervision, Conceptualization. **Simona Carfagna:** Writing – review & editing, Validation, Conceptualization. **Javed Ali Khan:** Writing – review & editing, Validation, Methodology, Formal analysis, Conceptualization.

Ethical statement

The work does not violate any ethical guidelines of the journal.

Declaration of competing interest

The authors declare that there are no competing conflicts of interest regarding the manuscript.

References

[1] A.Z. Worden, M.J. Follows, S.J. Giovannoni, S. Wilken, A.E. Zimmerman, P. J. Keeling, Rethinking the marine carbon cycle: factoring in the multifarious lifestyles of microbes, Science 347 (6223) (2015) 1257594.
[2] D.P. Häder, K. Gao, Interactions of anthropogenic stress factors on marine phytoplankton, Front. Environ. Sci. 3 (2015) 14.
[3] P.L. Williams, J.I. Robertson, A serious inhibition problem from a Niskin sampler during plankton productivity studies, Limnol. Oceanogr. 34 (7) (1989) 1300–1305.
[4] C.T. Taggart, W.C. Leggett, Efficiency of large-volume plankton pumps, and evaluation of a design suitable for deployment from small boats, Can. J. Fish. Aquat. Sci. 41 (10) (1984) 1428–1435.
[5] W. Aron, E.H. Ahlstrom, B.M. Bary, A.W.H. Be, W.D. Clarke, Towing characteristics of plankton sampling gear, Limnol. Oceanogr. 10 (3) (1965) 333–340.
[6] D. Sosa-Trejo, A. Bandera, M. González, S. Hernández-León, Vision-based techniques for automatic marine plankton classification, Artif. Intell. Rev. (2023) 1–32.
[7] A. Mitra, D.A. Caron, E. Faure, K.J. Flynn, S.G. Leles, P.J. Hansen, U. Tillmann, The Mixoplankton Database (MDB): diversity of photo-phago-trophic plankton in form, function, and distribution across the global ocean, J. Eukaryot. Microbiol. (2023) e12972.
[8] S.E. Mountcastle, N. Vyas, V.M. Villapun, et al., Biofilm viability checker: an open-source tool for automated biofilm viability analysis from confocal microscopy images, npj Biofilms Microbiomes 7 (2021) 44, https://doi.org/10.1038/s41522-021-00214-7.
[9] M.R. Hipsey, D.P. Hamilton, P.C. Hanson, C.C. Carey, J.Z. Coletti, J.S. Read, B. W. Ibelings, F.J. Valesini, J.D. Brookes, Predicting the resilience and recovery of aquatic systems: a framework for model evolution within environmental observatories, Water Resour. Res. 51 (9) (2015) 7023–7043.
[10] B.H. Davis, [Review of freedom and tenure in the academy: the fiftieth anniversary of the 1940 statement of principles, in: W.W. Van Alstyne (Ed.), Academe 78 (3) (1992) 67–69, https://doi.org/10.2307/40250336.
[11] S. Nishizawa, Photographic study of suspended matter and plankton in the sea, 北大彙報 5 (1954) 36–40.
[12] H.E. Edgerton, L.D. Hoadley, Cameras and lights for underwater use, SMPTE Journal 64 (7) (1955) 345–350, 1955.
[13] R. Schröder, Untersuchungen Uber Die Planktonverteilung Mit Hilfe Der Unterwasser-Fernsehanlage und Des Echographen (1961).
[14] P.B. Ortner, S.R. Cummings, R. Aftring, H.E. Edgerton, Silhouette photography of oceanic zooplankton, Nature 277 (5691) (1979).
[15] P.B. Ortner, L.C. Hill, H.E. Edgerton, In-situ silhouette photography of gulf stream zooplankton. Deep sea research Part A, Oceanograph. Res. Papers 28 (12) (1981) 1569–1576.
[16] R. Froese, K.G. Barthel, W. Welsch, M. Rolke, C. Schubert, B. Hermann, D. Weg, Development of an underwater video system for recording of ichthyoplankton and zooplankton, ICES CM 50 (1990) 90.
[17] G. Gorsky, M. Picheral, L. Stemmann, Use of the underwater video profiler for the study of aggregate dynamics in the north mediterranean, Estuar. Coast Shelf Sci. 50 (1) (2000) 121–128.
[18] S. Samson, T. Hopkins, A. Remsen, L. Langebrake, T. Sutton, J. Patten, A system for high-resolution zooplankton imaging, IEEE J. Ocean. Eng. 26 (4) (Oct. 2001) 671–676, https://doi.org/10.1109/48.972110.
[19] M. Ciranni, V. Murino, F. Odone, V.P. Pastore, Computer vision and deep learning meet plankton: milestones and future directions, Image Vis Comput. 143 (2024) 104934.
[20] E.C. Orenstein, D. Ratelle, C. Briseño-Avena, M.L. Carter, P.J. Franks, J.S. Jaffe, P. L. Roberts, The Scripps Plankton Camera system: a framework and platform for in situ microscopy, Limnol Oceanogr. Methods 18 (11) (2020) 681–695.
[21] E. Merz, T. Kozakiewicz, M. Reyes, C. Ebi, P. Isles, M. Baity-Jesi, F. Pomati, Underwater dual-magnification imaging for automated lake plankton monitoring, Water Res. 203 (2021) 117524.

[22] J.Y. Luo, J.O. Irisson, B. Graham, C. Guigand, A. Sarafraz, C. Mader, R.K. Cowen, Automated plankton image analysis using convolutional neural networks, Limnol Oceanogr. Methods 16 (12) (2018) 814–827.

[23] D.J. Matuszewski, Computer Vision for Continuous Plankton Monitoring, Doctoral dissertation, Universidade de São Paulo, 2014.

[24] S.S. Rawat, A. Bisht, R. Nijhawan, A deep learning based CNN framework approach for plankton classification, in: 2019 Fifth International Conference on Image Information Processing (ICIIP), IEEE, 2019, November, pp. 268–273.

[25] X. Li, Z. Cui, Deep residual networks for plankton classification, in: OCEANS 2016 MTS/IEEE Monterey, IEEE, 2016, September, pp. 1–4.

[26] C. Wang, X. Zheng, C. Guo, Z. Yu, J. Yu, H. Zheng, B. Zheng, Transferred parallel convolutional neural network for large imbalanced plankton database classification, in: 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), IEEE, 2018, May, pp. 1–5.

[27] A. Lumini, L. Nanni, Deep learning and transfer learning features for plankton classification, Ecol. Inform. 51 (2019) 33–43.

[28] A. Rachman, A.S. Suwarno, S. Nurdjaman, Application of deep (machine) learning for phytoplankton identification using microscopy images, in: 7th International Conference on Biological Science (ICBS 2021), Atlantis Press, 2022, May, pp. 213–224.

[29] T. Kerr, J.R. Clark, E.S. Fileman, C.E. Widdicombe, N. Pugeault, Collaborative deep learning models to handle class imbalance in flowcam plankton imagery, IEEE Access 8 (2020) 170013–170032.

[30] A. Lumini, L. Nanni, Deep learning and transfer learning features for plankton classification, Ecol. Inform. 51 (2019) 33–43.

[31] Alessandra Lumini, Loris Nanni, Gianluca Maguolo, Deep learning for plankton and coral classification, Appl. Comput. Inform. 19 (3/4) (2020) 265–283.

[32] Py Ouyang, Hu Hong, Shi Zhongzhi, Plankton classification with deep convolutional neural networks. 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, 2016. IEEE.

[33] N.A. Samee, E.S.M. El-Kenawy, G. Atteia, M.M. Jamjoom, A. Ibrahim, A. A. Abdelhamid, M.Y. Shams, Metaheuristic optimization through deep learning classification of COVID-19 in chest X-ray images, Comput. Mater. Continua (CMC) 73 (2) (2022).

[34] M. Ciranni, F. Odone, V.P. Pastore, Anomaly detection in feature space for detecting changes in phytoplankton populations, Front. Mar. Sci. 10 (2024) 1283265.

[35] L. Nanni, A. Loreggia, L. Barcellona, S. Ghidoni, Building ensemble of deep networks: convolutional networks and transformers, IEEE Access 11 (2023) 124962–124974.

[36] N. Ullah, M. Hassan, J.A. Khan, M.S. Anwar, K. Aurangzeb, Enhancing explainability in brain tumor detection: a novel DeepEBTDNet model with LIME on MRI images, Int. J. Imag. Syst. Technol. 34 (1) (2024) e23012.

[37] A.L. Teigen, A. Saad, A. Stahl, Leveraging similarity metrics to in-situ discover planktonic interspecies variations or mutations, in: Global Oceans 2020: Singapore–US Gulf Coast, 2020, October, pp. 1–8. IEEE.

[38] H.M. Sosik, R.J. Olson, Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry, Limnol. Ocean. Meth. 5 (6) (2007) 204–216.

[39] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, Proc. AAAI Conf. Artif. Intell. 31 (1) (2017, February).

[40] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Software 95 (2016) 51–67.

[41] M.T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you?" Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, August, pp. 1135–1144.

[42] L. Liao, H. Li, W. Shang, L. Ma, An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks, ACM Trans. Software Eng. Methodol. 31 (3) (2022) 1–40.

[43] A. Maracani, V.P. Pastore, L. Natale, L. Rosasco, F. Odone, In-domain versus out-of-domain transfer learning in plankton image classification, Sci. Rep. 13 (1) (2023) 10443.

[44] S.P. Kyathanahally, T. Hardeman, E. Merz, T. Bulas, M. Reyes, P. Isles, M. Baity-Jesi, Deep learning classification of lake zooplankton, Front. Microbiol. 12 (2021) 746297.

[45] J. Dai, Z. Yu, H. Zheng, B. Zheng, N. Wang, A hybrid convolutional neural network for plankton classification, in: Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III 13, Springer International Publishing, 2017, pp. 102–114.

[46] A. Venkataramanan, M. Laviale, C. Figus, P. Usseglio-Polatera, C. Pradalier, Tackling inter-class similarity and intra-class variance for microscopic image-based classification, in: International Conference on Computer Vision Systems, Springer International Publishing, Cham, 2021, September, pp. 93–103.

[47] J. Guo, J. Guan, Classification of marine plankton based on few-shot learning, Arabian J. Sci. Eng. 46 (9) (2021) 9253–9262.

[48] J. Cálem, C. Moreira, J. Jorge, Intelligent systems in healthcare: a systematic survey of explainable user interfaces, Comput. Biol. Med. 180 (2024) 108908.

[49] N. Ullah, J.A. Khan, I. De Falco, G. Sannino, Explainable artificial intelligence: importance, use domains, stages, output shapes, and challenges, ACM Comput. Surv. 57 (4) (2024) 1–36.

[50] K. Liao, B. Yan, Z. Ding, J. Huang, X. Fan, S. Wu, H. Li, X-scPAE: an explainable deep learning model for embryonic lineage allocation prediction based on single-cell transcriptomics revealing key genes in embryonic cell development, Comput. Biol. Med. 188 (2025) 109787.