International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)
31 July - 2 August 2025, Rangpur, Bangladesh

Jellyfish Species Identification: A CNN Based Artificial Neural Network Approach

Md. Sabbir Hossen[a], Md. Saiduzzaman[a], Pabon Shaha[b*], and Mostofa Kamal Nasir[b],
[a]Bangladesh University, Dhaka-1207, Bangladesh
[c]Mawlana Bhashani Science & Technology University, Tangail-1902, Bangladesh
sabbirhossen5622@gmail.com, saiduzzamancse56bu@gmail.com, pabonshahacse15@gmail.com, kamal@mbstu.ac.bd

2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN) | 979-8-3315-9694-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/QPAIN66474.2025.11172004

***Abstract*—Jellyfish are a diverse category of marine gelatinous organisms that are vital to the balance of aquatic ecosystems, yet they present notable difficulties for biodiversity preservation and ecological management due to their rapid growth and environmental influence. Precisely identifying different types of jellyfish is vital for ecosystem observation and effective control. In this research, we introduced a convolutional neural network-based approach for detecting and classifying jellyfish types utilizing an underwater image dataset. This approach incorporates sophisticated feature extraction methods, such as VGG16, EfficientNetV2-B0, ResNet50, and MobileNetV3, in conjunction with three Feedforward Neural Network classifiers and seven conventional machine learning classifiers, to enhance species recognition accuracy. Moreover, we employed the softmax function for direct classification of jellyfish species through CNN architectures. The integration of the MobileNetV3 with Artificial Neural Network emerged as the most accurate configuration, attaining an impressive accuracy of 98%, and markedly surpassing other combinations of feature extractors and classifiers. This research demonstrates the potential of hybrid models and CNN architecuters in tackling biodiversity-related issues and improving species classification in aquatic ecosystems**

***Index Terms*—Jellyfish Detection, Underwater Object Detection, Multilayer Perceptron, Deep Learning, and Machine Learning**

I. INTRODUCTION

Jellyfish, classified under the Phylum Cnidaria, are soft-bodied marine creatures that inhabit oceans across the globe [1]. These organisms are essential to the marine ecosystem, functioning as both prey and predators, yet they also present serious obstacles to marine biodiversity and human activities [2], [3]. Although jellyfish blooms assist in cycling nutrients and serve as a food source for various aquatic creatures, they also create problems by disrupting fishing industries, obstructing power plant cooling systems, and stinging swimmers [4]. The precise classification of jellyfish is vital for monitoring biodiversity and regulating the environment, and reducing the harmful consequences of jellyfish blooms. Conventional identification techniques are often slow and susceptible to mistakes, particularly when processing underwater images influenced by environmental elements like low visibility and turbidity [5]. Newest developments in machine learning and computer vision provide effective solutions to automate and improve the detection and classification of jellyfish species [6]. This research presents a deep learning method for classifying jellyfish utilizing underwater imagery. The dataset utilized in this work includes 1080 jellyfish images taken underwater, sourced from publicly accessible platforms and specialized marine research institutes. The proposed approach integrates advanced CNN architectures for both direct classification and feature extraction, such as VGG16, EfficientNetV2-B0, ResNet50, and MobileNetV3. Additionally, the extracted features were combined by a hybrid method with three Feedforward Neural Network classifiers and traditional machine learning classifiers to achieve accurate and stable performance. The methods were assessed using traditional evaluation metrics like accuracy, precision, recall, and F1 score. This study's remaining sections are organized as follows: Chapter II provides an overview of current studies on underwater object and jellyfish detection as well as their environmental importance. Chapter III details the dataset, the preprocessing process, and the suggested model and framework. Chapter IV outlines the implementation procedure and experimental setup. Chapter V reports the experimental outcomes along with the comparison. An overview of the results and future prospects for jellyfish identification and environmental observation is finally provided in Chapter VI.

*Corresponding Author:

The backing for this study comes from "NextGen AI Lab and Bangladesh University"; we appreciate the resources, direction, and assistance the organization has given us.

II. Related Work

The domain of jellyfish identification remains relatively unexplored, with only a few studies dealing with this field. Obstacles such as poor image clarity and underwater noise make it challenging to create accurate models. To mitigate these issues, researchers frequently reference adjacent areas such as fish identification, marine trash identification, and underwater object detection, which includes coral reef analysis. These developments offer adaptable strategies that are beneficial to jellyfish classification. This section examines the existing studies on jellyfish detection, along with the significant progress in object detection from underwater, to establish context and emphasize the importance of further study in this field. *S Roy et al.* [7] created an effective ML-based underwater object recognition system for detecting items in underwater imagery. The utilization of VGG-16 in conjunction with transfer learning was the main focus of the study. The pre-trained weights of VGG16 contributed useful information about fundamental image features. They used three datasets to train a CNN and a neural network classifier based on VGG-16. The accuracy of the VGG-16-based classifier was 84.89% using transfer learning. *J. Roy et* al. [8] employed YOLOv8 architectures that were improved with a MaxRGB filter for accurate underwater object detection and classification. The MaxRGB filter was used in their experiment to improve content separability in images. Their proposed method reached a classification accuracy of 99.6% and achieved a mean average precision of 98.6%. *T.-N. Pham et al.* [9] proposed the jellyfish classification method built on DL based object detection using optical imagery. The enhanced YOLOv5-nanomodel presented in this paper is based on the addition of GAM and the replacement of conventional Conv modules in the conventional structure's backbone with CoordinateCov modules. Experimental findings indicate that the proposed upgraded model surpasses the alternatives, achieving 89.1% mAP@0.5 and outperforming YOLOv8, YOLOv6, Faster R-CNN, SSD, and RetinaNet. *Y. Han et al.* [10] implemented DL techniques to identify various jellyfish types. The researchers utilized multiple DL models, including GoogLeNet and AlexNet. Their results showed that the jellyfish classification task using the GoogLeNet backbone achieved 96.21% accuracy, outperforming AlexNet in comparative analysis. Furthermore, the Faster R-CNN algorithm's detection performance was assessed using the two backbone networks for jellyfish identification. Results demonstrated that the GoogLeNet-based Faster R-CNN achieved an average detection accuracy of 74.96%, indicating superior performance. *U. Nawarathne et al.* [11] examined multiple DL architectures for classifying jellyfish, including YOLOv8, the most recent YOLO architecture, and several other CNNs like DenseNet, InceptionV3, MobileNet, MobileNetV2, and NASNet Mobile. YOLOv8 delivered the best results, with an impressive 99.5% accuracy in identifying and classifying jellyfish instances. *G.M Firdaus et al.* [12] addresses the challenge of classifying jellyfish images by leveraging the EfficientNetB3 model alongside the capabilities of DCNNs. The research involved a dataset consisting of 900 jellyfish images across 6 distinct categories and employed the EfficientNetB3 model, chosen for its effective balance between performance efficiency and architectural complexity. According to the classification accuracy evaluation, EfficientNetB3-based DCNNs outperformed traditional CNN models in image classification tasks, achieving a notable accuracy rate of 96.67%. *M. Vishwakarma et al.* [13] applied CNN models for feature extraction from images, employing the MobileNetV2 model with transfer learning, and presented a comprehensive approach to jellyfish identification. The CNN model was trained using several species of jellyfish. In addition, image features were analyzed by flattening and resizing using a classic ML approach, SVM. Three categories of jellyfish were used as the dataset for training the SVM model. Experimental results showed strong performance from both methods; the CNN obtained 97% accuracy on the training set, while the SVM model yielded good results on a separate test set.

III. MATERIALS & METHODS

This section of our study discusses the research methodology. Among the steps in this process are feature extraction, machine learning, selection of a multilayer perceptron model, dataset definition, preprocessing, and data augmentation. The suggested workflow diagram is shown in Figure 1.

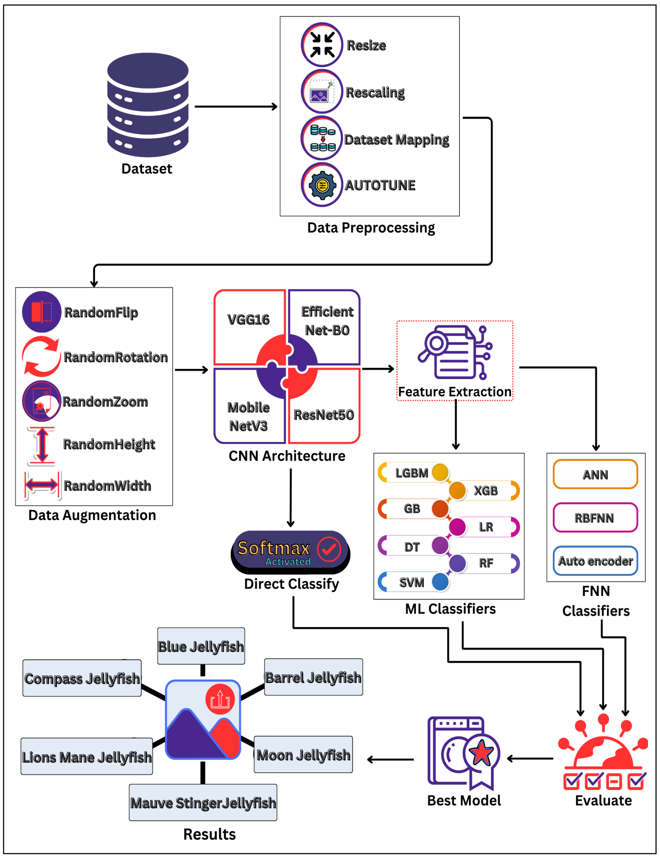


Fig. 1. The suggested workflow diagram for identifying jellyfish classes

A. Dataset

The dataset employed in this study, titled "Jellyfish Image Dataset," was obtained from Kaggle [14]. The dataset contains high-resolution deep-sea images of jellyfish taken in diverse aquatic environments. It comprises six unique jellyfish types: Moon Jellyfish, Mauve Stinger Jellyfish, Lion's Mane Jellyfish, Compass Jellyfish, Blue Jellyfish, and Barrel Jellyfish. For every class, there are 150 training images, 15 testing images, and 15 validation images, totaling 1,080 pictures across all species. Each image is accurately annotated with its species label, ensuring precise and reliable supervised learning. Figure 2 shows three examples from each species of jellyfish images.

B. Data Preprocessing

The preprocessing phase applied various techniques to guarantee high-quality inputs for the models. Initially, normalization was performed by scaling pixel values from their original range of [0, 255] to [0, 1], which standardizes inputs and speeds up model convergence. Moreover, all images were resized to 224x224 pixels for consistency and compatibility with the model architecture. Dataset mapping was used to carry out this preprocessing using TensorFlow's map() function, enabling efficient per-sample transformations. To boost processing performance, the AUTOTUNE parameter was used to parallelize data loading and preprocessing, fully utilizing hardware capabilities.

(a) Moon (b) Moon (c) Moon (d) Mauve Stinger (e) Mauve Stinger (f) Mauve Stinger

(g) Lions Mane (h) Lions Mane (i) Lions Mane (j) Compass (k) Compass (l) Compass

(m) Blue (n) Blue (o) Blue (p) Barrel (q) Barrel (r) Barrel

Fig. 2. Sample images representing various jellyfish types from the dataset, such as Moon jellyfish, Mauve Stinger jellyfish, Lion's Mane jellyfish, Blue jellyfish, Compass jellyfish, and Barrel jellyfish, highlighting differences in color, form, and texture.

C. Data Augmentation

To improve the variety and strength of datasets, data augmentation methods were employed to synthetically enlarge the training dataset by producing 10,000 new samples. Random transformations were applied during the augmentation process, such as RandomRotation, which applies image rotations within a specified angle range to manage orientation changes, and RandomFlip, which performs horizontal flips to replicate different viewing perspectives. Moreover, RandomZoom was utilized to simulate differences in object distance, while RandomWidth and RandomHeight adjustments modified aspect ratios to reflect real-world distortion effects. These augmentation techniques not only expanded the dataset volume but also enabled the architectures to better generalize by training on diverse image patterns. The result was increased robustness in handling varied visual appearances across jellyfish images.

D. Convolutional Neural Network Architectures

In this study, we implemented multiple CNN architectures for feature extraction and classification of various jellyfish species from underwater imagery. In particular, we adopted VGG-16 [15], EfficientNetV2-B0 [16], ResNet-50 [17], and MobileNetV3 [18] to leverage their cutting-edge capabilities in image processing. Each architecture was first trained independently to perform direct classification of jellyfish species. Afterward, the same models were employed as feature extractors, and the obtained features were input into multilayer perceptrons and traditional ML classifiers for further classification tasks.

1) Convolutional Neural Network-Based Classification

This method relied on CNNs to carry out end-to-end classification, where the model concurrently learns both feature extraction and jellyfish species prediction. Raw input images are passed through several convolutional layers, which detect essential visual cues, like textures, edges, and geometric patterns, that are important for classification. These convolutional layers were subsequently followed by pooling operations, which reduced the spatial dimensions while maintaining significant feature information. The resulting feature maps were flattened and forwarded through a series of dense layers. A softmax activation function was utilized in the output dense layer to output class probabilities, enabling the network to assign a likelihood score to each species. The class corresponding to the highest probability score was selected as the final prediction.

2) Feature Extraction

To extract features, pre-trained CNN models were used for their deep hierarchical understanding of image structures. These models identify features at various levels, ranging from low to high, such as edges, texture, and complex object parts, through their convolutional and pooling layers. The intermediate feature outputs from convolutional layers were used instead of the final classification layers. These features were flattened and then used as inputs for separate FNN or ML classifiers, such as Artificial Neural Network (ANN) or Support Vector Machine (SVM), which handled the final prediction task.

E. Model Selection

To detect jellyfish species, we analyzed both advanced feedforward neural network (FNN) models [19] [20] and standard machine learning classifiers [21], [22]. Features extracted using pre-trained CNNs were utilized as inputs to these classifiers, supporting accurate and robust species classification.

1) Machine Learning Classifiers

We tested various machine learning algorithms on the extracted features for jellyfish classification. The models include Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Gradient Boosting (GB), and Support Vector Machine (SVM). Each model offers benefits when handling structured datasets and high-dimensional inputs. SVM constructs decision boundaries in complex spaces, while RF and DT capture non-linear dependencies. LR offers a strong baseline for multi-class classification. Ensemble models such as LightGBM, GB, and XGBoost, are renowned for their predictive power and robustness.

2) Feedforward Neural Network Classifiers

Alongside conventional ML approaches, we employed sophisticated FNN models to achieve improved detection accuracy. These include an Autoencoder-based classifier, a Radial Basis Function Neural Network (RBFNN) using an SVM RBF kernel, and an Artificial Neural Network (ANN). The ANN used several hidden layers and activation functions to learn complex feature interactions. The RBFNN combined kernel techniques and neural networks to define nonlinear decision boundaries by utilizing SVM's RBF kernel. The autoencoder-based model learned compressed feature representations in an unsupervised manner and fine-tuned them for the final classification task. These FNN models offered greater flexibility in recognizing deep patterns and contributed to achieving top-level accuracy.

Algorithm 1: Jellyfish Species Identification

```
Initialize: DS, PP, DA, CNN, FE, ML, FNN ; // Set up all required modules
DS ← Dataset (6 Distinct Species) ;
PP ← [ AUTOTUNE, Data Mapping, Rescaling, Resizing] ;
DA ← [Width, Height, zoom, rotation, Random Flip] ;    // Apply Augmentation (Generate 10,000 extra samples)
CNN ← [VGG16, EfficientNetV2-B0, ResNet50, MobileNetV3] ;
  // Use of Pre-trained CNN Architectures
FE ← Extract Features (Excluding Classification Head) ;
ML ← [LGBM, XGB, GB, LR, DT, RF, SVM] ;
FNN ← [Autoencoder-based, RBFNN, ANN] ;
X ← DS[Images], Y ← DS[Labels] ;
for i ← 1 to |CNN| do
    // Classification Using CNN Directly
    CNN[i].add(Softmax Layer) ;
    CNN[i].compile(optimizer=Adam) ;
    CNN[i].fit(X, Y, epochs=20, batch_size=32) ;
    Metrics: [Accuracy, Precision, Recall, F1-Score] ;
    // Extracted Feature-Based ML Classification
    FV ← FE(CNN[i], X) ;                  // Derive Features
    for j ← 1 to |ML| do
        X_T, Y_T, x_t, y_t ← Split Data (70:30)(FV, Y) ;
        ML[j].fit(X_T, Y_T) ;             // Train ML Classifier
        y_pred ← ML[j].predict(x_t) ;     // Generate Predictions
        Metrics: [Accuracy, Precision, Recall, F1-Score] ;
    // Extracted Feature-Based FNN Classification
    for k ← 1 to |FNN| do
        X_T, Y_T, x_t, y_t ← Split Data (70:30)(FV, Y) ;
        FNN[k].fit(X_T, Y_T) ;            // Train Neural Network
        y_pred ← FNN[k].predict(x_t) ;    // Generate Predictions
        Metrics: [Accuracy, Precision, Recall, F1-Score] ;
Deinitialize: DS, PP, DA, CNN, FE, ML, FNN ;   // Cleanup
```

IV. EXPERIMENTAL SETUP & IMPLEMENTATION

A. Experimental Setup

Every experiment was conducted using a 64-bit version of Windows 11 operating system running on a system equipped with an Intel Core i5 8th Gen processor (1.60–3.90 GHz), 12 GB of DDR4 RAM, and an NVIDIA GeForce MX250 GPU.. Anaconda Navigator was used to access the Jupyter Notebook, which was used for all tasks, including training models and processing images.. The complete evaluation and training phase lasted over ten hours, underscoring the computational workload involved.

B. Implementation Details

The jellyfish species classification workflow followed a well-structured and modular implementation strategy. First, the dataset was retrieved from a public platform and processed through several normalization and resizing steps to ensure consistency. Various augmentation techniques, like rotating, cropping, and flipping, were applied to enrich the training data and improve model generalization. The classification system was executed in three separate stages.

First, four CNN models—VGG16, EfficientNetV2-B0, ResNet50, and MobileNetV3—were used for direct jellyfish classification, employing the softmax activation for probabilistic output. Second, feature vectors were extracted using the same CNN models, which were then fed into three Feedforward Neural Network (FNN) models: an Autoencoder-based classifier, an RBFNN with an SVM kernel, and an Artificial Neural Network (ANN). Third, we employed seven ML classifiers to predict the jellyfish class. Model evaluation was conducted using common performance indicators, including accuracy, precision, recall, and F1-score, which enabled a comparative study to determine the most efficient classifier for jellyfish detection. The overall classification methodology is presented in Algorithm 1.

V. RESULT ANALYSIS & DISCUSSION

This section presents a thorough assessment of the different methods and models employed for jellyfish types detection. The effectiveness of the models is measured using metrics including Accuracy (A), Precision (P), Recall (R), and F1-score (F1), which help assess their capability to differentiate among jellyfish species. The evaluation encompasses comparisons of Convolutional Neural Network architectures, the hybrid multilayer perceptron, and machine learning classifiers, emphasizing their advantages and drawbacks. The findings are showcased through visualization and detailed tables, providing clear insights into the most effective jellyfish detection methods.

A. Result Analysis

This I provides a summary of the effectiveness of four CNN architectures that were assessed for jellyfish species classification utilizing the Softmax activation function: EfficientNetV2-B0, ResNet50, MobileNetV3, and VGG16. MobileNetV3 demonstrated the best results across all metrics, attaining 93% accuracy, precision, recall, and F1-score. Both ResNet50 and EfficientNet-B0 showed comparable outcomes, with 92% scores across all metrics. VGG16 recorded the lowest results, achieving 89% on each metric. EfficientNet-B0 and ResNet50 both performed similarly, scoring 92% on all measures. Out of all the models, VGG16 performed the worst, scoring 89% on each metric.

TABLE I
PERFORMANCE OF PRE-TRAINED MODELS WITH SOFTMAX ACTIVATION

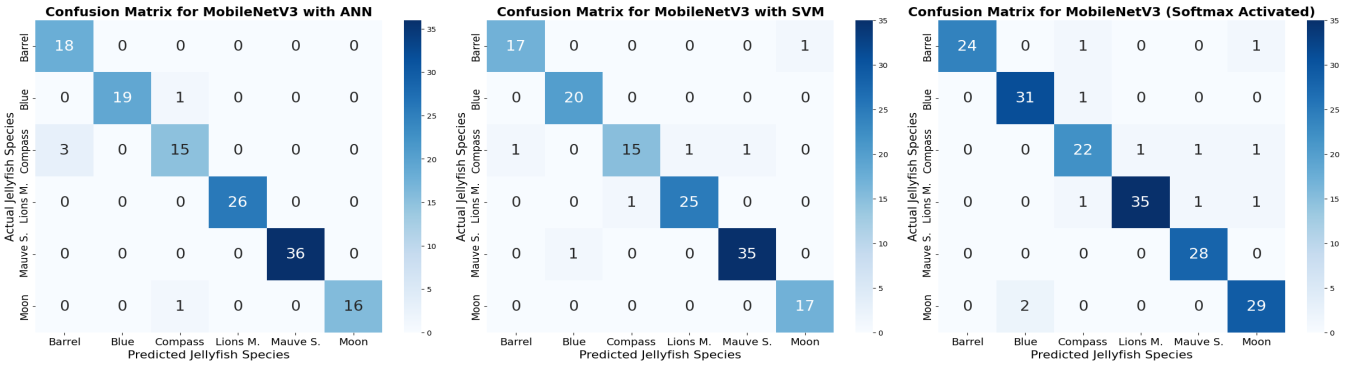
Architectures	Accuracy	Precision	Recall	F1-score
EfficientNetV2-B0	92%	92%	92%	92%
ResNet50	92%	91%	91%	91%
MobileNetV3	93%	93%	93%	93%
VGG16	89%	89%	89%	89%

TABLE II
Analysis of the Performance of Machine Learning Classifiers Utilizing Features Obtained from CNN Architectures

ML Classifier	Feature Extraction Techniques															
	VGG16				MobileNetV3				ResNet50				EfficientNetV2-B0			
	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1
LGBM	80%	80%	84%	80%	88%	88%	91%	88%	87%	88%	91%	88%	76%	79%	81%	78%
XGB	76%	76%	79%	76%	88%	87%	89%	88%	89%	88%	88%	88%	77%	79%	82%	78%
GB	80%	81%	83%	81%	88%	88%	89%	89%	86%	85%	86%	85%	77%	79%	79%	78%
LR	88%	88%	90%	88%	96%	97%	96%	96%	92%	91%	91%	91%	88%	88%	90%	89%
DT	59%	59%	61%	59%	60%	61%	63%	61%	60%	60%	61%	60%	57%	57%	59%	57%
RF	81%	81%	84%	81%	83%	83%	86%	84%	88%	88%	88%	88%	79%	81%	83%	80%
SVM	87%	86%	89%	87%	97%	97%	97%	97%	96%	96%	96%	96%	90%	90%	92%	90%

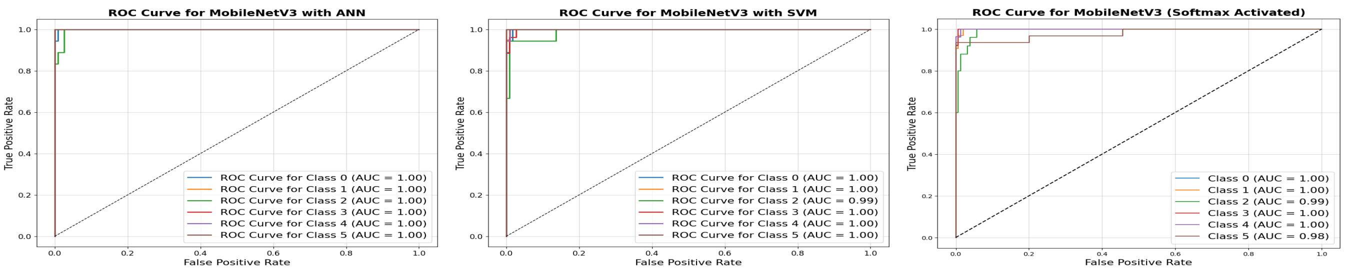
TABLE III
Analysis of the Performance of FNN Classifiers Utilizing Features Obtained from CNN Architectures

MLP Classifier	Feature Extraction Techniques															
	VGG16				MobileNetV3				ResNet50				EfficientNet-B0			
	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1
Autoencoder	81%	81%	84%	82%	96%	97%	96%	97%	91%	91%	94%	91%	89%	89%	90%	89%
RBFNN	85%	85%	88%	86%	90%	90%	91%	90%	89%	89%	93%	90%	87%	87%	89%	88%
ANN	86%	86%	88%	87%	98%	98%	98%	98%	90%	90%	92%	91%	89%	89%	91%	89%



(a) (b) (c)

Fig. 3. Confusion matrices evaluate the effectiveness of three distinct methods for classifying jellyfish species using MobileNetV3: (a) feature extraction in conjunction with artificial neural networks, (b) feature extraction in conjunction with SVM, and (c) softmax activation.



(a) (b) (c)

Fig. 4. The performance of MobileNetV3 is compared using the receiver-operating characteristic (ROC) curve utilizing three distinct methods: (a) feature extraction combined with an artificial neural network, (b) feature extraction combined with SVM, and (c) softmax activation. The ROC Curve and values for the six classes—0, 1, 2, 3, 4, and 5—are displayed in the figures.

The table II shows the evaluation results of seven ML algorithms using features derived from four CNN architectures. The combination of SVM and MobileNetV3 reached the top results, scoring 97% across all evaluation metrics. The SVM also produced strong outcomes, with ResNet-50 reaching 96% Accuracy. On the other hand, the Decision Tree yielded the poorest performance with all convolutional neural network backbones, showing an Accuracy as low as 59% when combined with ResNet50 and VGG16. All things considered, MobileNetV3 delivered better performance than other feature extractor models for most classifiers, while EfficientNetV2-B0 offered close results, though slightly lower, especially when used with LGBM and RF classifiers.

The table III displays the evaluation of three Feedforward Neural Network classifiers using extracted features from the four CNN models. The pairing of MobileNetV3 with an ANN produced the best outcome, achieving 98% Accuracy across all performance evaluation metrics. Autoencoder-based and RBFNN models also performed well when paired with MobileNetV3, achieving accuracy scores of 96% and 90%, respectively. Autoencoder-based model and ANN paired with ResNet50 likewise delivered strong performance, attaining an Accuracy of 91% and 90%, respectively. Conversely, the Autoencoder produced weaker results, with VGG16 attaining an accuracy of just 81%. Overall, MobileNetV3 outperformed all other Feedforward Neural Network classifiers in terms of feature extraction.

Fig. 3 visualizes the confusion matrices of various classification methods for detecting jellyfish species using MobileNetV3. The MobileNetV3 model with Softmax activation delivers high detection accuracy, successfully detecting most jellyfish types with minimal error. The ANN and SVM models, built on extracted features, further improve performance by reducing classification errors. Among these, the Artificial Neural Network provides the most consistent results, with near-perfect classification accuracy across almost all species.

Fig. 4 presents the ROC curves that evaluate MobileNetV3's classification performance for jellyfish types. All models display outstanding accuracy, achieving nearly perfect Area Under the Curve (AUC) scores for all six species. In terms of overall effectiveness, the ANN-based model outperformed the others. The ANN model consistently achieved an AUC of 1.00 in every class, showcasing its superior classification capability. Although both the Softmax and SVM-based models performed well, with most AUC scores approaching 1.00, the most accurate model for identifying jellyfish species was the Artificial Neural Network.

B. Discussion & Limitations

In this work, we investigated several machine learning approaches, assessing them through various evaluation metrics. The findings indicate that the combination of Artificial Neural Network and MobileNetV3 consistently surpassed all other approaches. This model achieved top scores in Accuracy, Precision, Recall, F1, and AUC, confirming its reliability for automatic jellyfish identification. Nonetheless, this study faces some limitations. The dataset used is comparatively limited in size, which could hinder the model's ability to generalize across diverse ecological circumstances and broader jellyfish species. Additionally, our method relies solely on visual features from images, while incorporating ecological parameters, such as seasonal patterns, salinity, and water temperature, could potentially enhance accuracy. Another constraint involves the computational requirements of CNN architectures, which may limit deployment in low-resource or real-time environments.

VI. CONCLUSION & FUTURE WORK

Precise classification of jellyfish species is essential in preserving ecosystem balance and marine biodiversity. This research emphasizes the effectiveness of AI-powered classification systems in marine biology, utilizing hybrid machine learning and deep learning approaches for species detection and identification. The suggested models achieve excellent performance by fusing strong classifiers with CNN-based feature extraction, with the MobileNetV3 and Artificial Neural Network combination reaching 98% Accuracy and excellent scores in Precision, Recall, and F1. These contributions advance marine species recognition efforts and support AI integration in ecological monitoring. For future directions, extending the dataset to cover additional jellyfish types, improving real-time inference capabilities, and integrating contextual ecological data—such as environmental conditions and seasonal cycles—could further enhance model performance. Moreover, utilizing more recent deep learning techniques, such as self-supervised learning and transformer-based vision models, could enhance classification accuracy and improve generalization.

REFERENCES

[1] M. Avian and A. Ramšak, "Phylum cnidaria: Classes scyphozoa, cubozoa, and staurozoa," in *Invertebrate Zoology*, pp. 149–172, CRC Press, 2021.

[2] T. K. Doyle, G. C. Hays, C. Harrod, and J. D. Houghton, "Ecological and societal benefits of jellyfish," *Jellyfish blooms*, pp. 105–127, 2014.

[3] D. Pauly, W. Graham, S. Libralato, L. Morissette, and M. Deng Palomares, "Jellyfish in ecosystems, online databases, and ecosystem models," *Hydrobiologia*, vol. 616, pp. 67–85, 2009.

[4] J. E. Purcell, S.-i. Uye, and W.-T. Lo, "Anthropogenic causes of jellyfish blooms and their direct consequences for humans: a review," *Marine Ecology Progress Series*, vol. 350, pp. 153–174, 2007.

[5] P. L. Lee, M. N. Dawson, S. P. Neill, P. E. Robins, J. D. Houghton, T. K. Doyle, and G. C. Hays, "Identification of genetically and oceanographically distinct blooms of jellyfish," *Journal of the Royal Society Interface*, vol. 10, no. 80, p. 20120920, 2013.

[6] A. Gauci, A. Deidun, and J. Abela, "Automating jellyfish species recognition through faster region-based convolution neural networks," *Applied Sciences*, vol. 10, no. 22, p. 8257, 2020.

[7] S. Roy, P. Ghosh, T. Goto, and M. Sen, "Obdtl: Under water object classification using transfer learning," in *2024 4th International Conference on Computer, Communication, Control & Information Technology (C3IT)*, pp. 1–6, 2024.

[8] J. Roy and K. H. Talukder, "Under water objects detection and classification using deep learning technique," in *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*, pp. 1–5, IEEE, 2024.

[9] T.-N. Pham, V.-H. Nguyen, K.-R. Kwon, J.-H. Kim, and J.-H. Huh, "Improved yolov5 based deep learning system for jellyfish detection," *IEEE Access*, 2024.

[10] Y. Han, Q. Chang, S. Ding, M. Gao, B. Zhang, and S. Li, "Research on multiple jellyfish classification and detection based on deep learning," *Multimedia Tools and Applications*, pp. 1–16, 2022.

[11] U. Nawarathne, H. Kumari, and H. Kumari, "Comparative analysis of jellyfish classification: A study using yolov8 and pre-trained models," in *2024 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, vol. 7, pp. 1–6, IEEE, 2024.

[12] G. M. Firdaus, M. D. Sulistiyo, and N. M. Z. Hashim, "An improved jellyfish image classification using the efficientnetb3-architectured dcnn," in *2024 12th International Conference on Information and Communication Technology (ICoICT)*, pp. 314–319, IEEE, 2024.

[13] M. Vishwakarma, A. M. Yadav, and R. Ganesh, "Jellyfish recognition by using convolutional neural networks and support vector machine," *IRE Journal*, vol. 7, pp. 10–16, 2024.

[14] A. Tanwar, "Jellyfish images dataset, kaggle," 2023. Available at: https://bit.ly/42sDWVL.

[15] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[18] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

[19] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 7, pp. 579–588, 2009.

[20] M. S. Hossen, M. Saiduzzaman, and P. Shaha, "Social media sentiments analysis on the july revolution in bangladesh: A hybrid transformer based machine learning approach," in *Proceedings of the 17th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, 2025.

[21] M. S. Hossen, P. Shaha, M. Saiduzzaman, M. Shovon, A. K. Akhi, and M. S. Iqbal, "An explainable ai driven machine learning approach for maternal health risk analysis," in *27th International Conference on Computer and Information Technology (ICCIT)*, IEEE, 2024.

[22] M. S. Hossen, P. Shaha, and M. Saiduzzaman, "A hybrid machine learning approach utilizing cnn feature extraction with traditional classifier to identify strawberry leaf diseases," in *4th International Conference on Electrical, Computer and Communication Engineering (ECCE)*, IEEE, 2025.