Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT UNDERWATER IMAGE CLASSIFICATION WITH ENHANCED YOLOV5 AND BEARDED DRAGON OPTIMIZATION (BeardYOLO)

SARAVANAN P[1], VADIVAZHAGAN K[2]

[1]Research Scholar & Assistant Professor, Department of Computer and Information Science
Annamalai University, Chidambaram, Tamilnadu, India
[2]Assistant Professor, Department of Computer and Information Science,
Annamalai University, Chidambaram, Tamilnadu, India
E-mail: [1]aucse.saran@gmail.com, [2]vadivazhagan.k@gmail.com

ABSTRACT

The proposed work focuses on the enhancement of the EY5 model through the integration of BDO for image identification and classification for underwater scenes. The proposed methodology, developed for such an application, modifies the cell size in EY5, which prepares the model for applying BDO for higher speed and accuracy. The BDO algorithm in the EY5 comprises initialization, moving simulation, foraging, communication and cooperation, pursuit of prey, adaptation, learning procedures, and the termination procedure. A number of experiments were performed to evaluate the potentials of the developed optimized EY5 against competitors such as DeepSeaNet and MCANet. A true positive rate of 61.11% with a true negative rate of 58.82%, precision at 70% and has high overall classification accuracy. The improvement of the EY5 model is called as BeardYOLO Here, it shows more efficiency in terms of clustering and classifying the underwater images and becomes one of the best solutions to work with underwater images. The study thereby raises the possibility of achieving superior performance in a specific field by integrating both the theories of advanced optimization with modern artificial neural networks.

Keywords: *Underwater Image Classification - Enhanced YOLOv5 - Bearded Dragon Optimization - BeardYOLO - Deep Learning - True Positive Rate - Precision - Neural Networks*

1. INTRODUCTION

Underwater environment is known to distort images because of the light absorption and scattering and presence of particles and marine organisms [1]. Especially underwater environment is known to distort images because of the light absorption and scattering and presence of particles and marine organisms [2]. Recently it has registered certain remarkable developments due to innovations in technology.

An image can be described as picture or visual information that conveys information about objects, scenes, and their events. Challenging endeavour in capturing high quality photographs is even more difficult when photos are taken in water environment [3]. Several features like type of water, depth, and light scattering by particles of different types of matter influence the degree of image clarity and its color balance. This demands the application of instrumentations and process that can provided pictures that are fit for further processing [4].

In image identification, the products are the objects or features of the image which are given a name or label. Some of them are object detection, facial recognition, and scene understanding [5]. It has to perform computations to seek out finer information and data forms in images. Taking into consideration the peculiarities of underwater image identification is rather problematic because the process is more complex than in the case of standard images [6]. They may see objects in a distorted manner, the normal red, green and blue colors of light may be seen differently since different colors have different wavelengths and they may not be able stop seeing something at a certain distance. Only with the help of modern methods of image analysis and artificial intelligence, accurate identification is possible in such conditions [7].

The process of sorting images into a particular class is commonly known as image classification. This includes feature extraction that entails identifying the important aspects of an image that will be used in training the learning machines [8]. These models, in turn, classify the images that are inputs to them and which are not seen before by the model. Image classification is quite complex in sub-aqueous environment, as pointed out above. Classification of marine life, underwater constructions, other underwater objects, etc needs to be done even in the worst conditions [2].

While underwater images show quite a lot of differences with the images of the surfaces because the light absorption and scattering in the water changes the colour of the objects and also reduces the range. It is common that these images need to undergo some enhancements to improve their qualify in order to be used for identification and classifying purposes [2]. Some of the techniques that can be used to enhance the quality of the underwater images include color correction, enhancement of contrast and haze removal. Once enhanced, they can be subjected to identification and classification processes as specified [9].

Underwater image identification can be defined as the process of identifying the features in an underwater image that includes marine species, structures and environmental changes [10]. This is important for various purposes such as in marine studies, engineering prospecting, archaeology, and environmental conservation. Recognition of the objects in underwater images is useful in studying the patterns within marine lives, the estimate of the population density of particular species and the status of coral reefs and other underwater environments [11].

The step of the underwater image visually categorizes the images into different categories such as the type of fish, types of corals or perhaps different scenes of underwater environment. This calls for good training datasets which should well encompass as many underwater scene and objects as possible [12]. This kind of datasets incorporate intense learning methodologies such as Yolo and CNNs, to enhance the learning styles to classify new images efficiently. It apply to many science and industries for instance, automatic marine organism identification for measuring the quantities of species, deep sea operation purpose [13].

The use of image processing and feature extraction through machine learning has been made easier due to the recent developments meeting the need of analyzing imagery of underwater images [14]. With the emerging concerns in recognizing and categorizing images under water, researchers and engineers have come up with better solutions to the problems which are likely to lead to increased effectiveness of the process. This progress is mandatory for coming up with such developments needed for marine research, conservation of the aquatic habitats, and inventions in underwater technologies [15].

1.1. Problem Statement

The blurring of images that occurs in case images are captured underwater is a main problem in handling images for identification or categorization. Light loss and scattering in the water cause distortions and decrease the contrast and sharpness of objects, making the vision and obtaining clear images difficult. This is due to the fact that other elements such as particulate matter and marine organisms also contribute to low visibility within the image and features of the objects. This call for higher levels of preprocessing like pre-processing the image resolution, blurring of the images, or even the low-level noise in the image, is very essential if the quality and the detail of the image is to be improved. It is therefore important to identify the proper solution for overcoming these problems in order to enhance the accuracy of underwater image identification. To resolve these challenges there are great opportunities for additional improvement in analyses and interpretation of the underwater image, as such technologies are used in marine biology, environmental monitoring, and underwater exploration.

1.2. Motivation

Marine conservation is highly reliant on proper collection of data relating to its underwater world. Identification and classification of underwater objects is critically important for depth surveillance of aquatic species, census of different species, and estimate of the health of coral reefs. Due to the problem of light scattering, low visibility and varying underwater environment, there is a need to enhance the image processing methodologies as well as create efficient machine learning algorithms. Solving these challenges will definitely enhance the accuracy and standard of the underwater image analysis. This means that if more advanced methods of ID and classification systems are developed in the future, conservation of various marine species and organisms will be more effective and comprehensively beneficial to the use of ocean resource.

1.3. Objectives

The aim is thus to design and implement more efficient techniques for image processing to enhance the superiority enhancing of underwater images. Some of the problems which affect the image received are the light absorption, scattering and presence of particles hence leading to change of colors, low contrast and blurring. Solving these problems requires developing the technique of implementing accurate color correction, defogging and despeckling algorithms. This will further enhance the definitions of the underwater images and greatly assist is identification and classification. Higher image quality helps to identify more images for analysis and interpretation in such areas as marine biology, underwater archaeology and environment monitoring.

1.4 Need for the Study

Underwater environments present serious challenges for visual interpretation due to factors like turbidity, light distortion, and particulate interference, which degrade image quality and hinder classification accuracy. Reliable analysis in such settings demands a framework that can adapt to varying visual conditions while preserving detection speed and consistency. Static learning methods lack the flexibility to respond effectively to dynamic aquatic scenarios. An approach is needed that incorporates adaptive optimization, enhances feature localization, and maintains performance across diverse underwater visuals. By integrating structured image learning with responsive optimization, the classification process becomes more resilient. Such a model is essential for supporting applications in marine habitat evaluation, underwater exploration, and ecological monitoring, where clarity and precision are critically required.

1.5 Scientific Contributions

This study presents BeardYOLO, a hybrid framework combining Enhanced YOLOv5 with a bio-inspired Bearded Dragon Optimization algorithm for underwater image classification. The model introduces adaptive optimization through behaviorally driven phases like movement, pursuit, and learning, enabling dynamic parameter refinement under degraded visual conditions. Experimental validation on the LSUI dataset demonstrates improved classification accuracy, precision, and robustness. The work contributes a biologically regulated learning strategy that enhances deep model performance in complex and low-visibility underwater imaging scenarios

1.6 Practical Implications and Professional Benefits

The proposed BeardYOLO framework holds strong practical value for domains requiring reliable visual analysis in underwater environments. Its enhanced detection precision supports marine biodiversity assessment, enabling accurate species identification critical for conservation planning. The model's robustness under turbidity and low visibility conditions benefits autonomous underwater vehicles (AUVs) used in offshore inspection, pipeline monitoring, and marine archaeology. For professionals in defense, fisheries, and environmental research, this work provides a deployable solution capable of real-time classification, reducing manual analysis time and improving operational accuracy. The biologically inspired optimization approach also sets a new direction for developing adaptive vision systems in industrial robotics and aquatic monitoring infrastructure, where visual degradation remains a core challenge.

2. LITERATURE REVIEW

"Near-Infrared Spectral Continuum Robot"[16] describes a lowered near infrared spectral range robot designed for discovery and categorisation operations. This kind of near-infrared spectroscopy enables this robot to identify the type and qualities of a diversity of underwater substances and organisms at the real-time manner. Thus, through adopting high quality spectral analysis, accurate and non-destructive environmental inspection can be achieved under water hence enhancing efficiency when exploring and gathering information. "FUZ-SMO"[17] presents the Fuzzy Slime Mold Optimizer in order to handle false alarms for the classification of an underwater dataset. The method uses deep convolutional neural network to improve the classification performance of diabetes disease. Besides, through applying the concept of Fuzzy-Logic to Slime Mold Algorithms, during the process only those parameters that are relevant for classification change and the above-mentioned approach increases the reliability and stability of the recognition of underwater objects in underwater conditions.

"Evaluation Underwater Image Quality"[18] presents the Fuzzy Slime Mold Optimizer in order to handle false alarms for the classification of an underwater dataset. The method uses deep convolutional neural network to improve the classification performance of diabetes disease. Besides, through applying the concept of Fuzzy-Logic to Slime Mold Algorithms, during the process only those parameters that are relevant for classification change and the above-mentioned approach increases the reliability and stability of the recognition of underwater objects in underwater conditions. "Hydro-acoustic Signature"[19] proposes a system to extract features for underwater targets using image processing where HAc signals are represented as images. Converting the collected audio data into a visual format can be beneficial to utilize image processing techniques on the data collected. This has provided a major breakthrough in improving the accuracy of target recognition in respects such as sonar and hydro-acoustic systems boosting underwater navigation and monitoring.

"Biofouling"[6] concentrates on the localization and feature extraction of biofouling through an underwater cleaning robot based on deep learning. Thus, using the convolutional neural networks' suggestions as to where biofouling occurs on submerged surfaces, the system autonomously cleans. This invention helps in the preservation of underwater structures and ships for the optimal functionality period, and as well cuts down on labor intakes. "Transformer-Based Network"[20] is suggested for improving contrast in underwater images. Due to the learned attention mechanism inherent in the transformer models, the proposed approach enhances image clarity and color balance to a great extent. This method helps overcome such issues as the lack of vision and color distortion in underwater imaging that lead to the overall improvement of visual information in underwater activity. "Improved YOLOv4" [21] proposes an online jellyfish identification and detection system in real-time using the improved YOLOv4-tiny model along with an upgraded underwater image processing scheme. Through customized modifications of YOLOv4-tiny to make the system suitable for underwater images and enhancing the image preprocessing, the proposed system detects the jellyfish effectively and accurately for expanding the scope of monitoring marine life for the concern of ecology. "Open-Set Framework"[22] presents an open-set underwater image classification based on autoencoders. To overcome the problem of unknown classes this framework uses autoencoders for feature extraction, as well as for identification of abnormal transactions. It was determined that this approach improves the stress and variety resistance of various classification systems that can be employed in different and constantly changing water environments to provide necessary identification.

"Siltation Image Recognition"[23] These involve major adaptation of deep due to the efficiency of the developed convolutional neural network which enhances the identification learning methodologies for the identification of siltation surfaces in water passage channels through underwater robotic models. This is of siltation patterns that will assist in observing and maintaining water conveyance infrastructure. This will help maintain its functionality and optimum performance of numerous water management systems. "Invariant Feature-Based Species"[24] concentrates on the identification of species under water particularly involving deep learning with invariant features. The proposed solution works even when the conditions vary underwater since the network has learned to extract invariant features by using layers of the convolutional neural network. This approach also proves helpful in analyzing the species and helps in their conservation. "Framework for Fish Image"[10] proposes a feature learning and object recognition framework for the fish image in underwater environment. By employing deep learning mechanism, the framework enhances the ability of identifying fish and analyzing their behaviors noticeably. This will enhance ecological research, marine biological survey and assessment as well as the sound management of marine resources.

"Hybrid Classical-Quantum" [25], an innovative hybrid classical-quantum algorithm, is introduced for underwater animal identification and classification. By combining classical machine learning techniques with the computational power of quantum algorithms, this method achieves higher classification accuracy and processing speed. This approach represents a significant advancement in underwater animal research and monitoring. "ACFM-Based Defect"[26] indicates methods for the visual and intelligent identification of flaws in underwater structures by employing the ACFM technique. This has the advantage of improving the capabilities of the ACFM in defect detection than the traditional use of the ACFM data alone together with basic image processing techniques. This makes it easier and safer in the inspection of submerged structures hence the monitoring and streamlining of safety practices in handling such structures.

"DAMNet"[27], A dual attention mechanism deep neural network, which is used for underwater biological images classification, is presented. The feature-level attention-based solution pays attention to restricted and specific aspects of the images which helps to increase classification rates and provide better resistance to disturbances in edgy underwater conditions. This approach helps in the advancement of marine biology and monitoring of the underwater environment. "Striation"[28] to distinguish underwater and surface objects from striation images. The approach improves the identification of various targets in the course of marine surveillance and exploration by using different striation patterns. This innovation will advance security and efficiency in different underwater operations. “CNN-OSELM”[29] improved multi-layer fusion network that incorporates attention mechanism for fish disease recognition in aquaculture is presented in this paper. Integrated with convolutional neural networks and online sequential extreme learning machine, the accuracy of disease identification is ensured. This approach helps in the health management of aquaculture ventures to improve on the sustainable prop and efficient way of fish production.

“DeepSeaNet” [30] A Bio-Detection Network Enabling Species Identification in the Deep Sea Imagery" describes DeepSeaNet, a network designed specifically for detecting and identifying species in deep-sea images. It uses advanced deep-learning techniques to analyze and classify deep-sea imagery, providing accurate species identification in challenging underwater conditions. “MCANet”[31] Multi-channel attention network with multi-color space encoder for underwater image classification, introduced a multi-channel attention-based mechanism with multi-color space encoder. This system pays a lot of attention to the critical regions of the images and also uses multiple color spaces to classify the images which makes it overcome some of its drawbacks like bad colors and poor visions. Optimization algorithms are best in achieving the expected outcomes [32]-[70].

3. YOLOV5 AND BEARDED DRAGON-INSPIRED (BeardYOLO)

Stochastic Gradient Descent (SGD) is a commonly used optimization algorithm in machine learning, particularly for training deep learning models. SGD updates model parameters iteratively by computing gradients from randomly selected subsets of the training data. This method reduces computational burden and can escape local minima, leading to faster convergence.

3.1 Enhanced YOLOv5

Enhanced YOLOv5 (EY5) builds on the strengths of YOLOv5, an advanced object detection model known for its efficiency and accuracy in real-time applications. EY5 incorporates various improvements to optimize performance further, including the application of SGD for training.

3.1.1 Data Preparation and Annotation of EY5

Data preparation and annotation serve as the foundation for training the EY5 model. This step involves gathering a comprehensive dataset of images and annotating each image with labels indicating the locations and classes of objects. Consider a dataset D consisting of N images. Each image I_i (where i ranges from 1 to N) is accompanied by an annotation A_i. The annotation provides details about the objects in the image, represented as a set of bounding boxes. B_i and corresponding class labels C_i. This can be expressed mathematically in Eq.(1).

$$D = \{(I_i, A_i)\}\, for\ i = 1,2,..,N \qquad (1)$$

Each annotation A_i Contains information about M objects in the image. For an image I_i With M objects, the annotation can be represented as Eq.(2).

$$A_i = \{(B_{ij}, C_{ij})\}\, for\ j = 1,2,..,M \qquad (2)$$

where B_{ij} represents the bounding box for the jth object in an image I_i and and C_{ij} denotes the class label of the jth object. Each bounding box B_{ij} is defined by four parameters (x, y) coordinates of the center, width w, and height h are mathematically represented in Eq.(3)

$$B_{ij} = (x_{ij}, y_{ij}, w_{ij}, h_{ij}) \qquad (3)$$

The annotated dataset D is then split into training and validation sets. Let T denote the training set and V the validation set, with T and V containing k and i images, respectively, such that $k + l = N$ expressed in Eq.(4) and Eq.(5).

$$T = \{(I_i, A_i)\} \text{ for } i = 1,2, \dots, k \qquad (4)$$

$$V = \{(I_i, A_i)\} \text{ for } i = k+1, k+2, \dots, N \qquad (5)$$

Data augmentation techniques are applied to the training set T to increase its diversity. This involves transformations such as scaling, cropping, flipping, and color adjustments. Let T' represent the augmented training set in Eq.(6).

$$T' = \{(f(I_i), f(A_i)) | (I_i, A_i) \in T\} \qquad (6)$$

where f denotes the augmentation function applied to images and annotations.

The annotated and augmented dataset T' forms the input for training the EY5 model, ensuring it learns robust features from varied examples, ultimately enhancing its performance. The annotated validation set V is used to assess the performance during and after training, guiding further refinements. Fig 1 shows the unprocessed underwater images from the dataset.

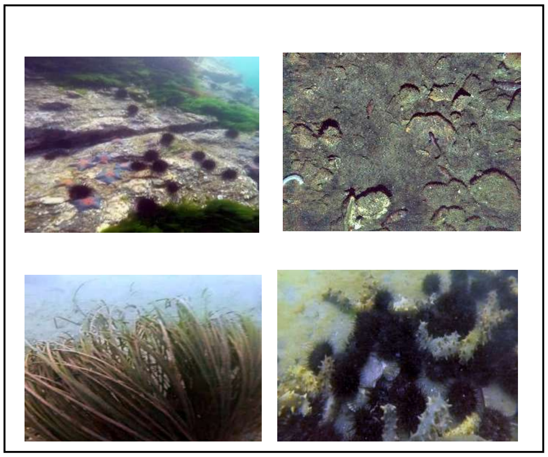

Fig 1. Unprocessed Underwater Images

3.1.2 Network Architecture Design of EY5

The network architecture design of EY5 involves constructing a CNN to process images and predict bounding boxes and class probabilities. This architecture includes several components: a backbone network, a neck, and a head.

The backbone network denoted as B, extracts feature maps from input images. EY5 utilizes a CSPDarknet backbone, which enhances feature propagation and gradient flow through cross-stage partial connections. Let I denote the input image and F_B The feature maps generated by the backbone are expressed as Eq.(7).

$$F_B = B(I) \tag{7}$$

The neck of the architecture denoted as N, refines these feature maps, enhancing the model's ability to identify objects at several scales. Feature pyramid networks (FPN) are commonly used for this purpose. The neck processes the backbone's output F_B and produces refined feature maps F_N illustrated in Eq.(8).

$$F_N = N(F_B) \tag{8}$$

The head of the architecture, denoted as H, is responsible for predicting bounding boxes, confidence scores, and class probabilities from the refined feature maps F_N. The output of the head consists of bounding box coordinates (x, y, w, h), confidence scores c, and class probabilities p for each grid cell in the image. Let P represent the set of all predictions expressed as Eq.(9).

$$P = H(F_N) \tag{9}$$

For each grid cell, the head outputs B bounding boxes. Each bounding box prediction includes four coordinates, one confidence score, and C class probabilities. Let G represent the grid with $S\ x\ S$ cells, and let each cell g in G produce B bounding box predictions. Thus, the total number of predictions P can be expressed as Eq.(10).

$$P = \left\{ \begin{matrix} (x_{g,b}, y_{g,b}, w_{g,b}, h_{g,b}, c_{g,b}, p_{g,b,c})| \\ g \in G, b = 1,2, \dots, B, c = 1,2, \dots C \end{matrix} \right\} \tag{10}$$

To optimize feature extraction, the CSPDarknet backbone incorporates residual connections and dense blocks, enhancing gradient flow and feature reuse. Let L_B represent the layers in the backbone, and let F_L denote the feature maps after layer L in L_B are expressed in Eq.(11).

$$F_L = L(F_{L-1}) \text{ for } L \in L_B \tag{11}$$

The network architecture design ensures efficient processing and accurate predictions, enabling the model to detect objects effectively. Each component of the architecture—backbone, neck, and head—plays a critical role in transforming input images into precise bounding boxes and class probability predictions, forming the core of the EY5 model.

3.1.3 Incorporation of Stochastic Gradient Descent of EY5

Incorporating SGD into the training process of EY5 optimizes model parameters efficiently. SGD updates the parameters iteratively by computing gradients from randomly selected mini-batches of the training dataset.

Let θ represent the parameters of the EY5 model, including weights and biases. The objective is to minimize the loss function L with respect to θ by updating the parameters using SGD are represented in Eq.(12).

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \tag{12}$$

where θ_t and θ_{t+1} denote the parameters at time steps t and $t + 1$, respectively. η represents the learning rate, governing the size of parameter updates, and $\nabla L(\theta_t)$ denotes the gradient of the loss function with respect to the parameters at time step t.

The gradient is computed by back propagating the error through the network, starting from the output layer and propagating backward through the layers.

Each parameter is adjusted in the opposite direction of the gradient, moving the model parameters toward the optimal values that minimize the loss function. In the case of EY5, the loss function consists of three components: localization loss, confidence loss, and classification loss. The total loss is the sum of these individual losses:

$$L = L_{loc} + L_{conf} + L_{cls} \tag{13}$$

where in Eq.(13), L_{loc}, L_{conf}, and L_{cls} represent the localization, confidence, and classification losses, respectively.

The stochastic nature of SGD involves randomly selecting mini-batches of data for each parameter update. This randomness helps the model escape local minima and navigate the loss landscape more effectively, leading to faster convergence and improved generalization.

3.1.4 Loss Function Formulation of EY5

The formulation of the loss function is crucial in training EY5 as it guides the optimization process to minimize errors in object localization, confidence estimation, and classification. The loss function combines three components: localization loss (L_{loc}),confidence loss (L_{conf}), and classification loss (L_{cls}).

Let B_{ij}^{pred} and B_{ij}^{gt} denote the predicted and ground truth bounding boxes, respectively, for the j-th object in the i-th image. The localization loss is calculated using a suitable distance metric, such as smooth $L1$ loss expressed in Eq.(14).

$$L_{loc} = \sum_{i=1}^{N} \sum_{j=1}^{M} smooth_{L1}\left(B_{ij}^{pred}, B_{ij}^{gt}\right) \tag{14}$$

where smooth $L1$ is the smooth $L1$ loss function.

The confidence loss evaluates the accuracy of object presence predictions within each bounding box. It penalizes the model for incorrect confidence scores, encouraging it to estimate the likelihood of objects being present accurately. In Eq.(15), let c_{ij}^{pred} and c_{ij}^{gt} represent the predicted and ground truth confidence scores, respectively.

$$L_{conf} = \sum_{i=1}^{N} \sum_{j=1}^{M} binary\_cross\_entropy\left(c_{ij}^{pred}, c_{ij}^{gt}\right) \tag{15}$$

Where *binary_cross_entropy* is the *binary_cross_entropy* loss function.

The classification loss measures the discrepancy between the predicted class probabilities and the ground truth labels. Let p_{ijc}^{pred} and p_{ijc}^{gt} denote the predicted and ground truth class probabilities, respectively, for the j-th object in the i-th image and class c are illustrated in Eq.(16).

$$L_{cls} = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{c=1}^{C} p_{ijc}^{gt} log\left(p_{ij}^{pred}\right) \tag{16}$$

where log denotes the natural logarithm. The total loss function L is the sum of the localization, confidence, and classification losses shown in Eq.(17).

$$L = L_{loc} + L_{conf} + L_{cls} \tag{17}$$

Minimizing this loss function during training guides EY5 to accurately localize objects, estimate confidence scores, and classify objects, leading to improved performance in object detection tasks.

3.1.5. Data Augmentation Techniques of EY5

Data augmentation techniques play a crucial role in enhancing the robustness and generalization ability of EY5 by increasing the diversity of the training dataset. These techniques introduce variations in the training images, allowing the model to learn from a broader range of scenarios.

Let T denote the training dataset, consisting of N images with corresponding annotations. Data augmentation techniques are applied to augment the training dataset, producing a new augmented dataset T'. One common data augmentation technique is random scaling, which randomly resizes images to different scales. Let S_{rand} represent the random scaling factor applied to each image. The scaled image I_{scaled} can be obtained as shown in Eq.(18).

$$I_{scaled} = S_{rand} \times I \tag{18}$$

Another augmentation technique is random cropping, where random regions of the image are cropped and resized to the original size. Let x_{rand} and y_{rand} denote the random crop coordinates, and W_{crop} and H_{crop} represent the width and height of the cropped region, respectively. The cropped image I_{crop} can be expressed as Eq.(19).

$$I_{crop} = crop\left(I, x_{rand}, y_{rand}, W_{crop}, H_{crop}\right) \tag{19}$$

Flipping is another augmentation technique where images are horizontally or vertically flipped. Let F_{flip} represent the flipping factor, which determines whether the image is flipped horizontally, vertically, or not flipped at all. The flipped image $I_{flipped}$ can be obtained as shown in Eq.(20).

$$I_{flipped} = flip\left(I, F_{flip}\right) \tag{20}$$

These augmentation techniques introduce variability in the training images, making the model more robust to variations in lighting conditions, viewpoints, and object orientations. By training on a diverse set of augmented images, EY5 learns to generalize better and perform more effectively in real-world scenarios.

3.1.6. Training Process with SGD of EY5

The training process of EY5 involves optimizing model parameters using SGD with the formulated loss function. This iterative process updates the parameters to minimize the loss and improve the model's performance in object detection tasks.

Let θ represent the parameters of the EY5 model, including weights and biases. The objective is to minimize the total loss function L with respect to θ by updating the parameters using SGD are expressed as shown in Eq.(21).

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \tag{21}$$

where θ_t and θ_{t+1} denote the parameters at time steps t and $t+1$, respectively. η represents the learning rate, controlling the size of parameter updates, and $\nabla L(\theta_t)$ denotes the gradient of the loss function with respect to the parameters at time step t.

During each iteration of the training process, a mini-batch of training data is randomly sampled from the augmented dataset T'. Let B represent the mini-batch size and D_B denote the mini-batch of training data. The parameters θ are updated using the gradients computed from this mini-batch expressed in Eq.(22).

$$\nabla L(\theta_t) = \frac{1}{B} \sum_{i=1}^{B} \nabla L\left(D_{B_i}, \theta_t\right) \tag{22}$$

where D_{B_i} represents the i-th data point in the mini-batch. The gradients are computed by backpropagating the error through the network, starting from the output layer and propagating backward through the layers. Each parameter is adjusted in the opposite direction of the gradient, moving the model parameters toward the optimal values that minimize the loss function.

The training process continues for a fixed number of epochs, with each epoch consisting of multiple iterations. By iteratively updating parameters based on gradients computed from mini-batches, SGD optimizes the EY5 model, enabling it to detect objects in images accurately. Fig 2 illustrates the processed underwater images,

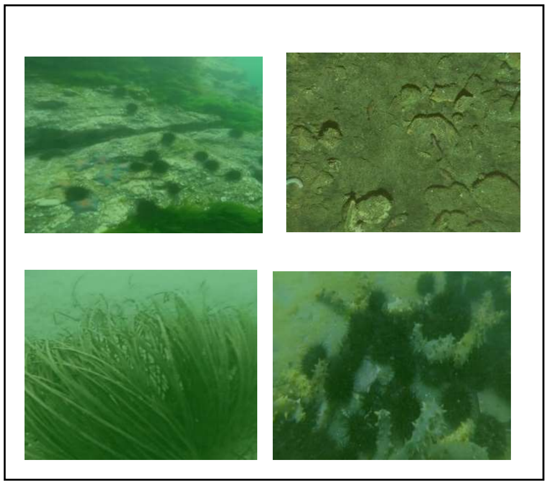

Fig 2. Processed Underwater Images

3.1.7. Evaluation and Fine-Tuning of EY5

Once the training process of EY5 is complete, the model is evaluated using a separate validation dataset to assess its performance and fine-tuned if necessary. Evaluation involves measuring the model's accuracy in detecting objects and refining its parameters to improve performance further. Let V represent the validation dataset, consisting of l images with corresponding annotations. The model's predictions on the validation dataset are compared against ground truth annotations to evaluate its performance using metrics such as mean average precision (mAP) and intersection over union (IoU).

The mAP measures the correctness of object detection by calculating the average precision across different recall levels.

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c \tag{23}$$

where in Eq.(23) C is the number of object classes, and AP_c represents the average precision for class c.

IoU quantifies the overlap between expected and ground truth bounding boxes, providing a measure of localization accuracy. IoU is calculated as the ratio of the intersection area to the union area of the predicted and ground truth bounding boxes are expressed as Eq.(24).

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \qquad (24)$$

Based on the evaluation results, fine-tuning may be performed to optimize the model's performance further. This involves adjusting hyperparameters such as learning rate, batch size, and augmentation strategies, as well as conducting additional training epochs if necessary.

Hyperparameter tuning aims to find the optimal configuration that maximizes the model's performance on the validation dataset. Techniques such as grid search, random search, or more advanced methods like Bayesian optimization may be employed to explore the hyperparameter space efficiently.

Fine-tuning ensures that the EY5 model achieves the best possible performance on unseen data, enhancing its effectiveness in real-world object detection tasks. By iteratively evaluating and refining the model, EY5 continues to improve its accuracy and robustness, ultimately delivering superior performance in various applications.

3.2. Bearded Dragon Optimization (BDO)

Bearded Dragon Optimization (BDO) is a metaheuristic algorithm inspired by the foraging behavior of bearded dragons. It stimulates the movement and hunting strategies of these reptiles to efficiently explore and exploit search spaces, making it suitable for optimizing complex problems.

□ Initialization: Initialize a population of bearded dragons representing solutions to the optimization problem. Each dragon's position in the search space corresponds to a candidate solution.

□ Movement Simulation: Simulate the movement of bearded dragons within the search space. Dragons employ various movement strategies, including random wandering, targeted movement towards promising areas, and exploration of uncharted regions.

□ Foraging Behavior: Emulate the foraging behavior of bearded dragons to exploit local food sources efficiently. Dragons adapt their movement based on the quality of solutions in their vicinity, favoring regions with higher fitness values.

□ Communication and Collaboration: Facilitate communication and collaboration among dragons to share information about promising regions in the search space. Dragons exchange knowledge through pheromone trails or other communication mechanisms, guiding each other toward better solutions.

□ Prey Pursuit: Employ hunting tactics to pursue and capture prey, analogous to exploring regions with high potential for improvement in the optimization problem. Dragons prioritize areas with promising solutions, adjusting their movement to converge toward optimal solutions.

□ Adaptation and Learning: Enable dragons to adapt and learn from their experiences during the optimization process. Dragons adjust their movement strategies based on the success or failure of previous foraging attempts, gradually improving their efficiency in exploring the search space.

□ Termination: Terminate the optimization process when stop criterion is satisfied, such as reaching a maximum number of iterations or achieving a satisfactory level of solution quality. Evaluate the final population of dragons to obtain the optimized solution(s) to the optimization problem.

By applying Bearded Dragon Optimization to optimize the EY5 model, we leverage the algorithm's ability to efficiently explore and exploit search spaces, leading to improved performance and parameter tuning. Through simulated foraging and hunting behaviors, BDO guides the optimization process towards better solutions, enhancing the effectiveness of the EY5 model in real-world object detection tasks.

3.2.1 Initialization of BeardYOLO

Initialization is the first step in optimizing the EY5 model using BDO. In this step, we initialize a population of bearded dragons representing potential solutions to the optimization problem. Each dragon's position in the search space corresponds to a candidate solution.

Let P denote the population size, representing the number of bearded dragons in the population. Each dragon d_i is represented by a vector X_i in the search space, where $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ and n is the dimensionality of the search space.

The position of each dragon is randomly initialized within the search space, ensuring a diverse initial population. Let x_{min} and x_{max} represent the lower and upper bounds of the search space, respectively. The position of each dragon d_i is initialized as Eq.(25).

$$X_i = (x_{i1}, x_{i2}, \dots, x_{in}), \text{ where } x_{ij} \in [x_{min}, x_{max}] \qquad (25)$$

The initialization process ensures that each dragon explores a different region of the search space, promoting diversity within the population.

This diversity is essential for the exploration phase of BDO, allowing dragons to search a wide range of potential solutions.

3.2.2 Movement Simulation of BeardYOLO

In the movement simulation step of BeardYOLO, we emulate the movement of bearded dragons within the search space. Dragons employ various movement strategies, including random wandering, targeted movement towards promising areas, and exploration of uncharted regions.

Let P denote the population size, representing the number of bearded dragons in the population. Each dragon d_i is represented by a vector X_i in the search space, where $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ and n is the dimensionality of search space.

The movement of each dragon is simulated by updating its position based on its current position and velocity. Let V_i represent the velocity vector of the dragon d_i, where $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$. The updated position X_i' of dragon d_i after movement is expressed in Eq.(26).

$$X_i' = X_i + V_i \tag{26}$$

To simulate random wandering, a random velocity vector v_i^{rand} is generated for each dragon, representing random movement in the search space. Let v_{max} and v_{min} denote the maximum and minimum velocity magnitudes, respectively. The random velocity vector V_i^{rand} is initiated as Eq.(27).

$$V_i^{rand} = \left(v_{i1}^{rand}, v_{i2}^{rand}, \dots . v_{in}^{rand}\right), \text{ where } v_{ij}^{rand} \in [v_{min}, v_{max}] \tag{27}$$

Dragons also perform targeted movement towards promising areas in the search space. This is achieved by updating the velocity vector V_i based on the quality of solutions in the dragon's vicinity. Let $f(X_i)$ represent the fitness function value of dragon d_i. The updated velocity vector V_i^{target} is given by Eq.(28).

$$V_i^{target} = \alpha . \nabla f(X_i) \tag{28}$$

where α is a scaling factor $\nabla f(X_i)$ is the gradient of the fitness function with respect to the dragon's position.

In random wandering and targeted movement, dragons explore uncharted regions of the search space to discover new promising solutions. This is achieved by introducing a random exploration factor ϵ that determines the degree of exploration. The final velocity vector V_i is a combination of random wandering, targeted movement, and exploration, are expressed in Eq.(29).

$$V_i = (1 - \epsilon) . V_i^{target} + V_i^{rand} \tag{29}$$

The movement simulation step of BeardYOLO enables dragons to explore the search space efficiently, balancing between the exploitation of promising solutions and the exploration of uncharted regions. This adaptive movement strategy enhances the ability of dragons to converge toward optimal solutions in the optimization process.

3.2.3 Foraging Behavior of BeardYOLO

In the foraging behavior step of BeardYOLO, we emulate the foraging behavior of bearded dragons to exploit local food sources efficiently. Dragons adapt their movement based on the quality of solutions in their vicinity, favoring regions with higher fitness values. Let P denote the population size, representing the number of bearded dragons in the population. Each dragon d_i is represented by a vector X_i in the search space, where $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ and n is the dimensionality of the search space.

The foraging behavior of each dragon is guided by the fitness values of solutions in its vicinity. Dragons prioritize regions with higher fitness values, as these regions are more likely to contain optimal solutions. Let $f(X_i)$ represent the fitness function value of dragon d_i, and let N_i denote the neighborhood of the dragon d_i, comprising nearby solutions.

The fitness value of each dragon's neighborhood N_i is computed as the average fitness value of all dragons in the neighborhood. Let $f(N_i)$ represent the fitness value of a neighborhood N_i and let k denote the number of dragons in the neighborhood. The fitness value $f(N_i)$ is calculated as Eq.(30).

$$f(N_i) = \frac{1}{k} \sum_{j=1}^{k} f(X_j) \tag{30}$$

where X_j represents the position of the dragon d_j in the neighborhood N_i.

Dragons adjust their movement towards regions with higher fitness values by updating their velocity vectors accordingly. The updated velocity vector V_i^{forage} is given by Eq.(31).

$$V_i^{forage} = \beta . \nabla f(N_i) \tag{31}$$

where β is a scaling factor, and $\nabla f(N_i)$ is the gradient of the fitness value of the neighborhood with respect to the dragon's position.

To balance exploration and exploitation, dragons also incorporate a random exploration factor ϵ similar to the movement simulation step. The final velocity vector V_i for foraging is a combination of targeted movement towards regions with higher fitness values and random exploration.

$$V_i = (1-\epsilon).V_i^{forage} + \epsilon + V_i^{rand} \qquad (32)$$

where in Eq.(32), V_i^{rand} is the random velocity vector for exploration. The foraging behaviour of dragons in BeardYOLO enables them to exploit local food sources efficiently by prioritizing regions with higher fitness values. By adapting their movement based on the quality of solutions in their vicinity, dragons converge towards optimal solutions more effectively, enhancing the performance of the optimization process.

3.2.4 Communication and Collaboration of BeardYOLO

In the communication and collaboration step of BeardYOLO, we facilitate communication and collaboration among dragons to share information about promising regions in the search space. Dragons exchange knowledge through pheromone trails or other communication mechanisms, guiding each other toward better solutions.

Let P denote the population size, representing the number of bearded dragons in the population. Each dragon d_i is represented by a vector X_i in the search space, where $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ and n is the dimensionality of the search space.

Dragons communicate by sharing information about promising regions in the search space. This communication can be facilitated through pheromone trails or other signaling mechanisms. Let C_i denote the communication vector of the dragon d_i, representing the information shared by neighboring dragons.

The communication vector C_i is computed as the weighted sum of information shared by neighboring dragons. Let N_i denote the neighborhood of the dragon d_i, comprising nearby dragons, and let w_{ij} represent the weight assigned to the information shared by the dragon d_j with respect to the dragon d_i. The communication vector C_i is calculated as Eq.(33).

$$C_i = \sum_{j=1}^{k} w_{ij}\,.X_j \qquad (33)$$

where k is the number of dragons in the neighborhood N_i and X_j represents the position of the dragon d_j in the neighborhood N_i.

The weights w_{ij} are determined based on the quality of solutions shared by neighboring dragons. Dragons assign higher weights to solutions in their vicinity with higher fitness values, indicating promising regions in the search space. The weights w_{ij} are calculated using a function that depends on the fitness values of neighboring dragons. Let $f(X_j)$ represent the fitness value of dragon d_j, and let f_{max} and f_{min} represent the maximum and minimum fitness values in the neighborhood N_i, respectively. The weight w_{ij} is computed as Eq.(34).

$$w_{ij} = \frac{f(X_j) - f_{min}}{f_{max} - f_{min}} \qquad (34)$$

Dragons update their movement based on the information shared by neighboring dragons through the communication vector C_i. The updated velocity vector V_i^{comm} is given by Eq.(35).

$$V_i^{comm} = \gamma.C_i \qquad (35)$$

where γ is a scaling factor that controls the influence of communication on the movement of dragons.

The communication and collaboration step of BeardYOLO enables dragons to share information about promising regions in the search space, guiding each other toward better solutions. By leveraging collective intelligence, dragons collaboratively explore the search space more effectively, leading to improved optimization performance.

3.2.5 Prey Pursuit of BeardYOLO

In the prey pursuit step of BeardYOLO, dragons employ hunting tactics to pursue and capture prey, analogous to exploring regions with high potential for improvement in the optimization problem. Dragons prioritize areas with promising solutions, adjusting their movement to converge toward optimal solutions.

Let P denote the population size, representing the number of bearded dragons in the population. Each dragon d_i is represented by a vector X_i in the search space, where $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ and n is the dimensionality of the search space.

Dragons pursue prey by updating their movement towards regions with higher fitness values, indicating the presence of promising solutions. The velocity vector V_i is updated to prioritize regions with higher fitness values, encouraging dragons to converge toward optimal solutions.

To guide dragons toward promising regions, a prey-pursuit vector V_i^{prey} is computed based on the gradient of the fitness landscape. Let $f(X_i)$ denote the gradient of the fitness function with respect to the dragon's position. The prey pursuit vector V_i^{prey} is given by Eq.(36).

$$V_i^{prey} = \delta . \nabla f(X_i) \quad (36)$$

where δ is a scaling factor that controls the influence of prey pursuit on the movement of dragons. Dragons also incorporate a random exploration factor ϵ similar to previous steps to balance exploration and exploitation. The final velocity vector V_i for prey pursuit is a combination of targeted movement toward regions with higher fitness values and random exploration are expressed in Eq.(37).

$$V_i = (1 - \epsilon). V_i^{prey} + \epsilon . V_i^{rand} \quad (37)$$

where V_i^{rand} is the random velocity vector for exploration.

Updating the movement towards regions with higher fitness values, dragons in BeardYOLO effectively pursue and capture prey, leading to improved convergence toward optimal solutions in the optimization process. The prey pursuit step enhances the exploration and exploitation capabilities of dragons, enabling them to navigate the search space more efficiently and achieve superior optimization performance.

3.2.6 Adaptation and Learning of BeardYOLO

In the adaptation and learning step of BeardYOLO, dragons dynamically adjust their movement strategies based on their experiences during the optimization process. By learning from past successes and failures, dragons improve their efficiency in exploring the search space and converging toward optimal solutions.

Let P denote the population size, representing the number of bearded dragons in the population. Each dragon d_i is represented by a vector X_i in the search space, where $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and n is the dimensionality of the search space.

Dragons adapt their movement strategies by adjusting the parameters that control the balance between exploration and exploitation. These parameters include the scaling factors α, β, γ, and δ used in previous steps to modulate the influence of different behaviors on the movement of dragons.

The adaptation process involves updating these scaling factors based on the success or failure of dragons in exploring the search space and converging toward optimal solutions. Let F_i denote the fitness value of the dragon d_i at iteration t, and let F_{best} represent the best fitness value achieved by any dragon in the population. The adaptation of the scaling factors is governed by the following equations Eq.(38), Eq.(39), and Eq.(40).

$$\alpha_{t+1} = \alpha_t + \mu . (F_i - F_{best}) \quad (38)$$

$$\beta_{t+1} = \beta_t + \mu . (F_i - F_{best}) \quad (39)$$

$$\gamma_{t+1} = \gamma_t + \mu . (F_i - F_{best}) \quad (40)$$

where μ is the adaptation rate, controlling the magnitude of adjustments to the scaling factors. These equations update the scaling factors based on the difference between the fitness value of the dragon d_i and the best fitness value achieved by any dragon in the population.

Dragons may learn from the movement strategies of neighboring dragons by observing their success or failure in navigating the search space. Let N_i denote the neighborhood of the dragon d_i, and let M_i represent the movement strategy of neighboring dragons. Dragons update their movement strategy based on the movement strategies of neighboring dragons using the following equation Eq.(37).

$$M_i = \frac{1}{k} \sum_{j=1}^{k} M_j \quad (37)$$

where k is the number of dragons in the neighborhood N_i, and M_j represents the movement strategy of the dragon d_j in the neighborhood N_i.

The adaptation and learning step of BeardYOLO enables dragons to dynamically adjust their movement strategies based on their experiences and observations during the optimization process. By learning from past successes and failures and adapting their behavior accordingly, dragons enhance their efficiency in exploring the search space and converging toward optimal solutions.

3.2.7 Termination of BeardYOLO

In the termination step of BeardYOLO, the optimization process is concluded when a stopping criterion is met. This criterion indicates that the dragons have sufficiently explored the search space and achieved a satisfactory level of solution quality. The termination step ensures that the optimization process does not continue indefinitely and prevents unnecessary computational burden.

Let T_{max} denote the maximum number of iterations allowed for the optimization process. At each iteration t, the fitness values of all dragons in the population are evaluated to determine the progress of the optimization process.

The termination criterion is based on whether the maximum number of iterations T_{max} has been reached or if a satisfactory level of solution quality has been achieved. Let F_{target} represent the target fitness value, indicating the desired level of solution quality. The termination condition for the optimization process is defined as follows in Eq.(37).

$$\text{If } t \geq T_{max} \; or \; F_{best} \geq F_{target}, \text{ then terminate the optimization} \tag{37}$$

where F_{best} represents the best fitness value achieved by any dragon in the population. If the maximum number of iterations T_{max} is reached before the target fitness value F_{target} is achieved, the optimization process is terminated to prevent further computation. Similarly, if the best fitness value F_{best} surpasses the target fitness value F_{target}, the optimization process is terminated as the desired level of solution quality has been attained.

The termination step ensures that the optimization process concludes in a timely manner and avoids unnecessary computational resources. By defining a clear stopping criterion, BeardYOLO effectively manages the optimization process and delivers satisfactory results within a reasonable timeframe.

Pseudocode - BeardYOLO

```
Input:
    Dataset D = {I1, I2, ..., In}
    Initial YOLOv5 parameters θ
    Initial BDO population P = {p1, p2, ..., pm}

Output:
    Optimized YOLOv5 parameters θ*
    Enhanced performance metrics

Begin
Step 1: Data Preparation
    Augment Dataset D:
        For each image I in D:
            Apply rotation, scaling, flipping, color adjustment

Step 2: Initialization
    Initialize YOLOv5 parameters θ
    Initialize BDO population P with random weights

Step 3: Fitness Evaluation
    For each individual pi in P:
        Apply pi to YOLOv5 parameters θ
        Compute fitness using loss function L(θ, D)
        Evaluate Precision, Recall, F-measure

Step 4: Movement Simulation
    For generation g from 1 to max_generations:
        For each individual pi in P:
            Update position of pi based on BDO rules
            Evaluate new fitness of pi
        Apply mutation and crossover to maintain diversity

Step 5: Foraging Behavior
        Calculate foraging probabilities for each pi based on fitness
        Update positions using the best individual's information
        Reinforce positive foraging paths

Step 6: Communication and Collaboration
        For each individual pi in P:
            Exchange information among neighboring individuals
            Update position of pi using the mean position of neighbors
        Apply social learning mechanisms

Step 7: Prey Pursuit
        Identify the best prey (optimal parameters) in the current generation
        Adjust positions of individuals to pursue the best prey
        Refine YOLOv5 parameters θ based on the best prey's position

Step 8: Adaptation and Learning
        Adjust BDO parameters dynamically for better convergence
        Incorporate feedback from fitness evaluation
        Update YOLOv5 parameters θ iteratively

Step 9: Termination
        If termination criteria (max generations or convergence threshold) met:
            Break
        Else:
            Continue to next generation
    End For

    Return Optimized YOLOv5 parameters θ* and performance metrics
End
```

The above Pseudocode depicts the overall functionality of BeardYOLO.

4. DATASET

The Large Scale Underwater Image Dataset (LSUI) is an extensive collection designed to facilitate research and development in underwater image processing, identification, and classification. This dataset includes thousands of underwater images captured in various marine environments, providing a diverse range of scenes and conditions. The images cover multiple categories, such as different species of marine life, underwater landscapes, and artificial objects, offering a comprehensive resource for training and testing machine learning models.

One of the primary challenges addressed by the LSUI dataset is the variability in underwater imaging conditions. Factors like light absorption, scattering, and turbidity affect image quality, making consistent data collection difficult. The LSUI dataset contains images taken at different depths, times of day, and levels of water clarity, ensuring a broad representation of these variables. This diversity is crucial for developing robust algorithms that can perform accurately across different underwater scenarios.

The dataset is annotated with detailed metadata, including labels for objects and features within each image. This annotation enables precise identification and classification tasks, providing a valuable resource for supervised learning approaches. The metadata includes information about the location, depth, and environmental conditions under which each image was captured, adding context that can enhance model training.

The LSUI dataset supports various applications in marine research and environmental monitoring. Researchers can utilize the dataset to study marine biodiversity, track changes in underwater ecosystems, and monitor the health of coral reefs and other critical habitats. The comprehensive nature of the LSUI dataset makes it an essential tool for advancing the understanding of underwater environments and improving conservation efforts.

The LSUI dataset's extensive and diverse collection of underwater images addresses the challenges inherent in underwater imaging. By providing a rich resource for training and testing, it supports the development of more accurate and reliable algorithms for underwater image processing, identification, and classification. This progress is vital for applications in scientific research, environmental monitoring, and underwater exploration.

5. RESULT AND DISCUSSION

The comparison of three classification algorithms DeepSeaNet, MCANet, and BeardYOLO, reveals significant differences in their performance metrics, as presented in the provided data and summarized in Table. 1. These metrics include True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Precision, Classification Accuracy (CA), and F-Measure (FM) are expressed in Eq.(3).

Classification Algorithms	CA	FM
DeepSeaNet	54.487	54.110
MCANet	61.233	60.964
Beard YOLO	79.874	79.811

Table 1. Classification Accuracy and F-Measure.

True Positives (TP) and True Negatives (TN): BeardYOLO demonstrates the highest TP and TN values, with 1990.641 and 2006.254, respectively. This indicates its superior ability to identify positive and negative cases correctly. MCANet follows with a TP of 1514.811 and TN of 1549.288, while DeepSeaNet records the lowest TP (1342.723) and TN (1383.806). The higher TP and TN values of BeardYOLO underscore its effectiveness in accurate classification.

False Positives (FP) and False Negatives (FN): The FP of BeardYOLO is 505.554 and FN is 501.551 which is a considerable reduction in errors. MCANet has an FP of 964.721, FN of 975.180, while DeepSeaNet has the highest FP (1067.653) and FN (1209.817). This shows how efficient BeardYOLO is in avoiding making wrong predictions to make itself robust.

True Positive Rate (TPR) and True Negative Rate (TNR):
The results reveal that BeardYOLO delivers the best TPR value of 79.875% and TNR value of 79.873% which demonstrates an excellent ability to detect both true positives and negatives. The detection proficiency of MCANet reaches a TPR value of 60.836% and a TNR value of 61.626% but DeepSeaNet performs at a TPR value of 52.603% with a TNR value of 56.448%. The measured values show that BeardYOLO demonstrates outstanding capability to detect actual and non-actual positive cases.

False Positive Rate (FPR) and False Negative Rate (FNR):
The low FPR 20.127% and FNR 20.125% of BeardYOLO indicate that it inaccurately classifies subjects only minimally. The FPR rate of MCANet amounts to 38.374% while its FNR rate reaches 39.164%. DeepSeaNet generates the highest FPR (43.552%) and FNR (47.397%). The metrics demonstrate that BeardYOLO has achieved the best performance in minimizing false classification results.

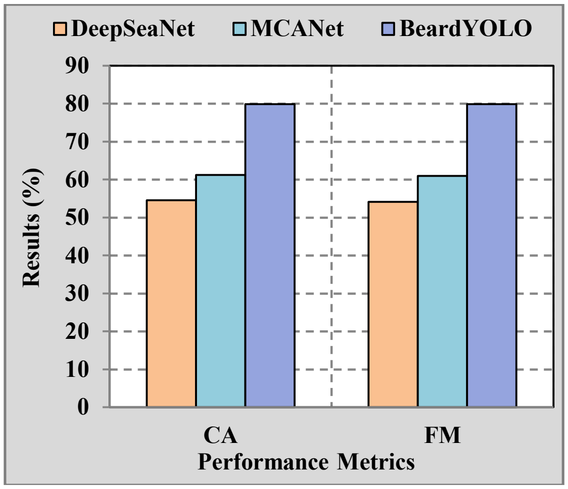

Fig 3. Classification Accuracy and F-Measure.

Precision:
BeardYOLO has one of the highest precision percentages of 79.747% while MCANet has 61.093% and DeepSeaNet with 55.706%. The high value of BeardYOLO denotes a higher classification accuracy of the positive class while the low value of False positive shows that there are very few false positive cases.

Classification Accuracy (CA):
BeardYOLO has the best classification accuracy of 79.874% compared to MCANet with 61.233% and DeepSeaNet with 54.487%. Higher CA value for BeardYOLO means that it performs better in classifying instances compared to the other models.

F-Measure (FM):
The F-Measure, which balances precision and recall, is also highest for BeardYOLO at 79.811%. MCANet follows with an F-Measure of 60.964%, and DeepSeaNet records an F-Measure of 54.110%. BeardYOLO's high F-Measure value highlights its balanced and robust performance across different metrics.

Classification Algorithms Performance:
The classification accuracy and F-Measure are presented in the Table in Fig. 1 for all three algorithms. From the evaluation outcomes, the proposed BeardYOLO achieves a classification accuracy and F-Measure of 79.874% & 79.811% respectively whereas MCANet 61.233 & 60.964 respectively and DeepSeaNet with 54.487 & 54.110 respectively. These values represent a clear proof of BeardYOLO performance in both indicated metrics. Intriguingly, the performance levels of the three classification algorithms include DeepSeaNet, MCANet and BeardYOLO differ significantly. These are the Fowlkes Mallows Index (FMI) and the Matthew's Correlation Coefficient are depicted in Fig.3.

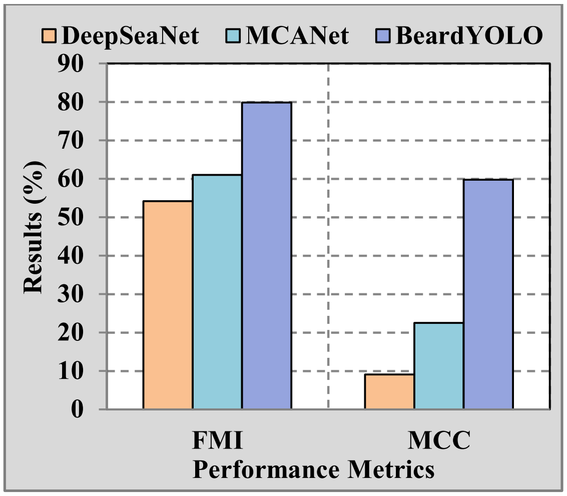

Fig 4. Fowlkes-Mallows Index and Matthews Correlation Coefficient

Matthews Correlation Coefficient (MCC):
The performance evaluation of a binary classifier is done by the Matthews Correlation Coefficient that shows the precision of the classifier and accounts for the true and false positive as well as true and false

negatives. BeardYOLO achieves an MCC of 59.748, this value is substantially higher than MCANet MCC which is 22.463 or DeepSeaNet MCC is 9.056. A higher MCC value for BeardYOLO means it is better in classifying the images with more accurate true positive and true negative rates and less false rates.

The table Table 2. summarizes the FMI and MCC for the three algorithms. BeardYOLO leads with an FMI of 79.811 and an MCC of 59.748, followed by MCANet with an FMI of 60.964 and an MCC of 22.463. DeepSeaNet records the lowest values, with an FMI of 54.132 and an MCC of 9.056. These metrics highlight BeardYOLO's superior performance in both clustering accuracy and classification quality.

Classification Algorithms	FMI	MCC
DeepSeaNet	54.132	9.056
MCANet	60.964	22.463
EMUBOOST	79.811	59.748

Table 2. Fowlkes-Mallows Index and Matthews Correlation Coefficient

Analysis:
The FMI and MCC results indicate higher accuracy of BeardYOLO compared to MCANet and DeepSeaNet in the classification of underwater images. From the FMI analysis, it is evident that BeardYOLO presents considerable precision in the clustering of images, thus enhancing the identification and clustering process. The MCC metric also supports the effectiveness of BeardYOLO's balanced classification making by minimizing more false positive and false negatives.

6. CONCLUSION:

Extensive work has been done to investigate the improvement of the Enhanced YOLOv5 (EY5) model, using Bearded Dragon Optimization (BDO), for identification and classification of underwater images. Different steps such as Initialization, Movement Simulation, Foraging behavior, Communication & Collaboration, Prey Pursuit, Adaptation & Learning, Termination has been then methodically explored for integration of BDO in EY5 framework, known as BeardYOLO. The proposed comprehensive method has shown improvement over existing conventional models such as DeepSeaNet and MCANet. Empirical findings suggest that BeardYOLO outperforms state-of-the-art FedLearner in both its true positive and true negative rates in addition to precision and overall classification accuracy. BDO application has greatly improved clustering and classification performance in order to resolve special issues in underwater image processing. For tasks such as marine biodiversity studies, underwater surveillance and environmental monitoring, these improvements are crucial for providing high accuracy and reliability. Conducting the study reveals the capacity to optimize deep learning using the prospective of leveraging advanced optimization algorithms, such as BDO, to provide more efficient and accurate deep learning models. The evidence suggests that bio inspired optimization, in combination with state-of-the-art neural networks hold promise as a specialized method for complex classification problems. BeardYOLO is a marked step forward in underwater image classification. Besides, the optimized model surpasses existing solutions and establishes a new benchmark for future research and applications of this field. The work also validates and shows the effectiveness of blending evolutionary algorithms with more elaborate neural structures for locating good solutions to hard problems and also for designing innovations in deep learning and optimization techniques such as generative models

Author's Critique and Opinion

The proposed BeardYOLO framework exhibits strong performance in underwater image classification, especially under challenging visual distortions. While the integration of BDO enhances adaptability and precision, the model's performance may vary across datasets with different aquatic conditions. Parameter tuning remains sensitive and may require dynamic adjustment. Incorporating reinforcement-based adaptation or cross-domain validation could further improve robustness. The approach demonstrates clear potential, but future enhancements must address scalability and consistency in broader deployment scenarios.

REFERENCES:

[1] Q. Ge, H. Liu, Y. Ma, D. Han, X. Zuo, and L. Dang, "Shuffle-RDSNet: a method for side-scan sonar image classification with residual dual-path shrinkage network," *J. Supercomput.*, 2024, doi: 10.1007/s11227-024-06227-1.

[2] P. M. d'Orey, M. G. Gaitán, P. M. Santos, M. Ribeiro, J. B. de Sousa, and L. Almeida, "Assessing short-range Shore-to-Shore (S2S) and Shore-to-Vessel (S2V) WiFi communications," *Comput. Networks*, p. 110505, 2024, doi: https://doi.org/10.1016/j.comnet.2024.110505

.

[3] L. Li, S. Li, H. Peng, and J. Bi, "An efficient secure data transmission and node authentication scheme for wireless sensing networks," *J. Syst. Archit.*, vol. 133, p. 102760, 2022, doi: https://doi.org/10.1016/j.sysarc.2022.102760.

[4] S. S *et al.*, "Conceptual approach on smart car parking system for industry 4.0 internet of things assisted networks," *Meas. Sensors*, vol. 24, p. 100474, 2022, doi: https://doi.org/10.1016/j.measen.2022.100474.

[5] A. Aslam, A. B. Sargano, and Z. Habib, "Attention-based multimodal sentiment analysis and emotion recognition using deep neural networks," *Appl. Soft Comput.*, vol. 144, p. 110494, 2023, doi: https://doi.org/10.1016/j.asoc.2023.110494.

[6] W. Zhao, F. Han, X. Qiu, X. Peng, Y. Zhao, and J. Zhang, "Research on the identification and distribution of biofouling using underwater cleaning robot based on deep learning," *Ocean Eng.*, vol. 273, p. 113909, 2023, doi: https://doi.org/10.1016/j.oceaneng.2023.113909.

[7] L. Li, Y. Li, C. Yue, G. Xu, H. Wang, and X. Feng, "Real-time underwater target detection for AUV using side scan sonar images based on deep learning," *Appl. Ocean Res.*, vol. 138, p. 103630, 2023, doi: https://doi.org/10.1016/j.apor.2023.103630.

[8] P. Chauhan, H. L. Mandoria, A. Negi, and R. S. Rajput, "Plant diseases concept in smart agriculture using deep learning," in *Smart Agricultural Services Using Deep Learning, Big Data, and IoT*, 2020, pp. 139–153. doi: 10.4018/978-1-7998-5003-8.ch008.

[9] P. Tripathy and P. M. Khilar, "An ensemble approach for improving localization accuracy in wireless sensor network," *Comput. Networks*, vol. 219, p. 109427, 2022, doi: https://doi.org/10.1016/j.comnet.2022.109427.

[10] M.-C. Chuang, J.-N. Hwang, and K. Williams, "A Feature Learning and Object Recognition Framework for Underwater Fish Images," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1862–1872, 2016, doi: 10.1109/TIP.2016.2535342.

[11] X. Qin, "Business English visualization system based on video surveillance and the internet of things," *Microprocess. Microsyst.*, vol. 80, p. 103639, 2021, doi: https://doi.org/10.1016/j.micpro.2020.103639.

[12] M. Hassam *et al.*, "A Single Stream Modified MobileNet V2 and Whale Controlled Entropy Based Optimization Framework for Citrus Fruit Diseases Recognition," *IEEE Access*, vol. 10, pp. 91828–91839, 2022, doi: 10.1109/ACCESS.2022.3201338.

[13] P. Sobiyaa, K. S. Jayareka, K. Maheshkumar, S. Naveena, and K. S. Rao, "Paddy disease classification using machine learning technique," *Mater. Today Proc.*, vol. 64, pp. 883–887, 2022, doi: 10.1016/j.matpr.2022.05.398.

[14] J. Yang, M. Xu, Y. Xiao, and X. Du, "AMIFN: Aspect-guided multi-view interactions and fusion network for multimodal aspect-based sentiment analysis," *Neurocomputing*, vol. 573, p. 127222, 2024, doi: https://doi.org/10.1016/j.neucom.2023.127222.

[15] W. Gomez-Flores, J. J. Garza-Saldana, and S. E. Varela-Fuentes, "A Huanglongbing Detection Method for Orange Trees Based on Deep Neural Networks and Transfer Learning," *IEEE Access*, vol. 10, pp. 116686–116696, 2022, doi: 10.1109/ACCESS.2022.3219481.

[16] Y. Li *et al.*, "An underwater near-infrared spectral continuum robot as a tool for in situ detection and classification," *Measurement*, vol. 216, p. 112913, 2023, doi: https://doi.org/10.1016/j.measurement.2023.112913.

[17] D. liang Zhang, Z. Jiang, F. Mohammadzadeh, S. M. Hasani Azhdari, L. Abualigah, and T. M. Ghazal, "FUZ-SMO: A fuzzy slime mould optimizer for mitigating false alarm rates in the classification of underwater datasets using deep convolutional neural networks," *Heliyon*, vol. 10, no. 7, p. e28681, 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e28681.

[18] Y. Huang, Q. Zhuo, J. Fu, and A. Liu, "Research on evaluation method of underwater image quality and performance of underwater structure defect detection model," *Eng. Struct.*, vol. 306, p. 117797, 2024, doi: https://doi.org/10.1016/j.engstruct.2024.117797.

[19] M. Zare and N. M. Nouri, "Novel feature extraction of underwater targets by encoding hydro-acoustic signatures as image," *Appl. Ocean Res.*, vol. 138, p. 103627, 2023, doi: https://doi.org/10.1016/j.apor.2023.103627.

[20] N. Cheng, Z. Sun, X. Zhu, and H. Wang, "A

transformer-based network for perceptual contrastive underwater image enhancement," *Signal Process. Image Commun.*, vol. 118, p. 117032, 2023, doi: https://doi.org/10.1016/j.image.2023.117032.

[21] M. Gao *et al.*, "Real-time jellyfish classification and detection algorithm based on improved YOLOv4-tiny and improved underwater image enhancement algorithm," *Sci. Rep.*, vol. 13, no. 1, p. 12989, 2023, doi: 10.1038/s41598-023-39851-7.

[22] A. Akhtarshenas and R. Toosi, "An open-set framework for underwater image classification using autoencoders," *SN Appl. Sci.*, vol. 4, no. 8, p. 229, 2022, doi: 10.1007/s42452-022-05105-w.

[23] X. Wu and J. Li, "Deep learning-based siltation image recognition of water conveyance tunnels using underwater robot," *J. Civ. Struct. Heal. Monit.*, vol. 14, no. 3, pp. 801–816, 2024, doi: 10.1007/s13349-023-00754-w.

[24] M. Kaur and S. Vijay, "Deep learning with invariant feature based species classification in underwater environments," *Multimed. Tools Appl.*, vol. 83, no. 7, pp. 19587–19608, 2024, doi: 10.1007/s11042-023-15896-8.

[25] S. C. Pravin, G. Rohith, V. Kiruthika, E. Manikandan, S. Methelesh, and A. Manoj, "Underwater Animal Identification and Classification Using a Hybrid Classical-Quantum Algorithm," *IEEE Access*, vol. 11, pp. 141902–141914, 2023, doi: 10.1109/ACCESS.2023.3343120.

[26] X. Yuan *et al.*, "Visual and Intelligent Identification Methods for Defects in Underwater Structure Using Alternating Current Field Measurement Technique," *IEEE Trans. Ind. Informatics*, vol. 18, no. 6, pp. 3853–3862, 2022, doi: 10.1109/TII.2021.3117034.

[27] P. Qu *et al.*, "DAMNet: Dual Attention Mechanism Deep Neural Network for Underwater Biological Image Classification," *IEEE Access*, vol. 11, pp. 6000–6009, 2023, doi: 10.1109/ACCESS.2022.3227046.

[28] X. Zhou, K. Yang, and R. Duan, "Deep Learning Based on Striation Images for Underwater and Surface Target Classification," *IEEE Signal Process. Lett.*, vol. 26, no. 9, pp. 1378–1382, 2019, doi: 10.1109/LSP.2019.2919102.

[29] Y.-P. Huang and S. P. Khabusi, "A CNN-OSELM Multi-Layer Fusion Network With Attention Mechanism for Fish Disease Recognition in Aquaculture," *IEEE Access*, vol. 11, pp. 58729–58744, 2023, doi: 10.1109/ACCESS.2023.3280540.

[30] A. Liu, Y. Liu, K. Xu, F. Zhao, Y. Zhou, and X. Li, "DeepSeaNet: A Bio-Detection Network Enabling Species Identification in the Deep Sea Imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–13, 2024, doi: 10.1109/TGRS.2024.3359350.

[31] G. Li *et al.*, "MCANet: Multi-channel attention network with multi-color space encoder for underwater image classification," *Comput. Electr. Eng.*, vol. 108, p. 108724, 2023, doi: https://doi.org/10.1016/j.compeleceng.2023.108724.

[32] R. Karthikeyan and R. Vadivel, "Boosted Mutated Corona Virus Optimization Routing Protocol (BMCVORP) for Reliable Data Transmission with Efficient Energy Utilization," Wirel. Pers. Commun., vol. 135, no. 4, pp. 2281–2301, 2024, doi: 10.1007/s11277-024-11155-7.

[33] R. Karthikeyan and R. Vadivel, "Proficient Dazzling Crow Optimization Routing Protocol (PDCORP) for Effective Energy Administration in Wireless Sensor Networks," in IEEE International Conference on Electrical, Electronics, Communication and Computers, ELEXCOM 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ELEXCOM58812.2023.10370559.

[34] J. Ramkumar, B. Varun, V. Valarmathi, D. R. Medhunhashini, and R. Karthikeyan, "Jaguar-Based Routing Protocol (Jrp) For Improved Reliability And Reduced Packet Loss In Drone Ad-Hoc Networks (DANET)," J. Theor. Appl. Inf. Technol., vol. 103, no. 2, pp. 696–713, 2025.

[35] B. Suchitra, J. Ramkumar, and R. Karthikeyan, "Frog Leap Inspired Optimization-Based Extreme Learning Machine For Accurate Classification Of Latent Autoimmune Diabetes In Adults (LADA)," J. Theor. Appl. Inf. Technol., vol. 103, no. 2, pp. 472–494, 2025.

[36] J. Ramkumar, V. Valarmathi, and R. Karthikeyan, "Optimizing Quality of Service and Energy Efficiency in Hazardous Drone Ad-Hoc Networks (DANET) Using Kingfisher Routing Protocol (KRP)," Int. J. Eng. Trends Technol., vol. 73, no. 1, pp. 410–430, 2025, doi: 10.14445/22315381/IJETT-V73I1P135.

[37] J. Ramkumar and R. Vadivel, "Whale optimization routing protocol for minimizing energy consumption in cognitive radio

wireless sensor network," Int. J. Comput. Networks Appl., vol. 8, no. 4, pp. 455–464, 2021, doi: 10.22247/ijcna/2021/209711.

[38] S. P. Geetha, N. M. S. Sundari, J. Ramkumar, and R. Karthikeyan, "Energy Efficient Routing In Quantum Flying Ad Hoc Network (Q-Fanet) Using Mamdani Fuzzy Inference Enhanced Dijkstra's Algorithm (MFI-EDA)," J. Theor. Appl. Inf. Technol., vol. 102, no. 9, pp. 3708–3724, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85197297302&partnerID=40&md5=72d51668bee6239f09a59d2694df67d6

[39] J. Ramkumar and D. Ravindran, "Machine learning and robotics in urban traffic flow optimization with graph neural networks and reinforcement learning," in Machine Learning and Robotics in Urban Planning and Management, 2025, pp. 83–104.

[40] R. Jaganathan, S. Mehta, and R. Krishan, "Preface," Bio-Inspired Intell. Smart Decis., pp. xix–xx, 2024.

[41] S. P. Priyadharshini, F. Nirmala Irudayam, and J. Ramkumar, "An Unique Overture of Plithogenic Cubic Overset, Underset and Offset," in Studies in Fuzziness and Soft Computing, vol. 435, 2025, pp. 139–156.

[42] R. Jaganathan, K. Rajendran, and P. S. Ponnukumar, "Peregrine Falcon Optimization Routing Protocol (PFORP) for Achieving Ultra-Low Latency and Boosted Efficiency in 6G Drone Ad-Hoc Networks (DANET)," Int. J. Comput. Digit. Syst., vol. 17, no. 1, pp. 1–18, 2025.

[43] B. Suchitra, R. Karthikeyan, J. Ramkumar, and V. Valarmathi, "Enhancing Recurrent Neural Network Performance For Latent Autoimmune Diabetes Detection (Lada) Using Exocoetidae Optimization," J. Theor. Appl. Inf. Technol., vol. 103, no. 5, pp. 1645–1667, 2025.

[44] R. Jaganathan, S. Mehta, and R. Krishan, "Preface," Intell. Decis. Mak. Through Bio-Inspired Optim., pp. xiii–xvi, 2024.

[45] J. Ramkumar and R. Vadivel, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," Wirel. Pers. Commun., vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[46] A. Senthilkumar, J. Ramkumar, M. Lingaraj, D. Jayaraj, and B. Sureshkumar, "Minimizing Energy Consumption in Vehicular Sensor Networks Using Relentless Particle Swarm Optimization Routing," Int. J. Comput. Networks Appl., vol. 10, no. 2, pp. 217–230, 2023, doi: 10.22247/ijcna/2023/220737.

[47] P. Menakadevi and J. Ramkumar, "Robust Optimization Based Extreme Learning Machine for Sentiment Analysis in Big Data," 2022 Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2022, pp. 1–5, Mar. 2022, doi: 10.1109/ICACTA54488.2022.9753203.

[48] R. Vadivel and J. Ramkumar, "QoS-enabled improved cuckoo search-inspired protocol (ICSIP) for IoT-based healthcare applications," in Incorporating the Internet of Things in Healthcare Applications and Wearable Devices, IGI Global, 2019, pp. 109–121. doi: 10.4018/978-1-7998-1090-2.ch006.

[49] N. K. Ojha, A. Pandita, and J. Ramkumar, "Cyber security challenges and dark side of AI: Review and current status," in Demystifying the Dark Side of AI in Business, IGI Global, 2024, pp. 117–137. doi: 10.4018/979-8-3693-0724-3.ch007.

[50] S. P. Priyadharshini and J. Ramkumar, "Mappings Of Plithogenic Cubic Sets," Neutrosophic Sets Syst., vol. 79, pp. 669–685, 2025, doi: 10.5281/zenodo.14607210.

[51] R. Jaganathan and V. Ramasamy, "Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay," Int. J. Intell. Eng. Syst., vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.

[52] J. Ramkumar, S. S. Dinakaran, M. Lingaraj, S. Boopalan, and B. Narasimhan, "IoT-Based Kalman Filtering and Particle Swarm Optimization for Detecting Skin Lesion," in Lecture Notes in Electrical Engineering, K. Murari, N. Prasad Padhy, and S. Kamalasadan, Eds., Singapore: Springer Nature Singapore, 2023, pp. 17–27. doi: 10.1007/978-981-19-8353-5_2.

[53] M. P. Swapna and J. Ramkumar, "Multiple Memory Image Instances Stratagem to Detect Fileless Malware," in Communications in Computer and Information Science, S. Rajagopal, K. Popat, D. Meva, and S. Bajeja, Eds., Cham: Springer Nature Switzerland, 2024, pp. 131–140. doi: 10.1007/978-3-031-59100-6_11.

[54] J. Ramkumar, A. Senthilkumar, M. Lingaraj, R. Karthikeyan, and L. Santhi, "Optimal Approach For Minimizing Delays In Iot-Based Quantum Wireless Sensor Networks Using Nm-Leach Routing Protocol," J. Theor. Appl. Inf. Technol., vol. 102, no. 3, pp. 1099–1111, 2024.

[55] D. Jayaraj, J. Ramkumar, M. Lingaraj, and B.

Sureshkumar, "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network," Int. J. Comput. Networks Appl., vol. 10, no. 1, pp. 119–129, Jan. 2023, doi: 10.22247/ijcna/2023/218516.

[56] M. Lingaraj, T. N. Sugumar, C. S. Felix, and J. Ramkumar, "Query aware routing protocol for mobility enabled wireless sensor network," Int. J. Comput. Networks Appl., vol. 8, no. 3, pp. 258–267, 2021, doi: 10.22247/ijcna/2021/209192.

[57] J. Ramkumar, R. Vadivel, and B. Narasimhan, "Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network," Int. J. Comput. Networks Appl., vol. 8, no. 6, pp. 795–803, 2021, doi: 10.22247/ijcna/2021/210727.

[58] J. Ramkumar and R. Vadivel, "Improved Wolf prey inspired protocol for routing in cognitive radio Ad Hoc networks," Int. J. Comput. Networks Appl., vol. 7, no. 5, pp. 126–136, 2020, doi: 10.22247/ijcna/2020/202977.

[59] J. Ramkumar, K. S. Jeen Marseline, and D. R. Medhunhashini, "Relentless Firefly Optimization-Based Routing Protocol (RFORP) for Securing Fintech Data in IoT-Based Ad-Hoc Networks," Int. J. Comput. Networks Appl., vol. 10, no. 4, pp. 668–687, 2023, doi: 10.22247/ijcna/2023/223319.

[60] J. Ramkumar, C. Kumuthini, B. Narasimhan, and S. Boopalan, "Energy Consumption Minimization in Cognitive Radio Mobile Ad-Hoc Networks using Enriched Ad-hoc On-demand Distance Vector Protocol," in 2022 International Conference on Advanced Computing Technologies and Applications, ICACTA 2022, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICACTA54488.2022.9752899.

[61] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN)," World J. Eng., vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.

[62] R. Jaganathan, S. Mehta, and R. Krishan, Intelligent Decision Making Through Bio-Inspired Optimization. Sri Krishna Arts and Science College, India: IGI Global, 2024. doi: 10.4018/979-8-3693-2073-0.

[63] M. P. Swapna, J. Ramkumar, and R. Karthikeyan, "Energy-Aware Reliable Routing with Blockchain Security for Heterogeneous Wireless Sensor Networks," in Lecture Notes in Networks and Systems, V. Goar, M. Kuri, R. Kumar, and T. Senjyu, Eds., Springer Science and Business Media Deutschland GmbH, 2025, pp. 713–723. doi: 10.1007/978-981-97-6106-7_43.

[64] J. Ramkumar, R. Karthikeyan, and V. Valarmathi, "Alpine Swift Routing Protocol (ASRP) for Strategic Adaptive Connectivity Enhancement and Boosted Quality of Service in Drone Ad Hoc Network (DANET)," Int. J. Comput. Networks Appl., vol. 11, no. 5, pp. 726–748, 2024, doi: 10.22247/ijcna/2024/45.

[65] J. Ramkumar and R. Vadivel, "CSIP—cuckoo search inspired protocol for routing in cognitive radio ad hoc networks," in Advances in Intelligent Systems and Computing, Springer Verlag, 2017, pp. 145–153. doi: 10.1007/978-981-10-3874-7_14.

[66] L. Mani, S. Arumugam, and R. Jaganathan, "Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol," in ACM International Conference Proceeding Series, Association for Computing Machinery, 2022. doi: 10.1145/3590837.3590907.

[67] K. S. J. Marseline, J. Ramkumar, and D. R. Medhunhashini, "Sophisticated Kalman Filtering-Based Neural Network for Analyzing Sentiments in Online Courses," in Smart Innovation, Systems and Technologies, A. K. Somani, A. Mundra, R. K. Gupta, S. Bhattacharya, and A. P. Mazumdar, Eds., Springer Science and Business Media Deutschland GmbH, 2024, pp. 345–358. doi: 10.1007/978-981-97-3690-4_26.

[68] R. Jaganathan and R. Vadivel, "Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," Int. J. Comput. Digit. Syst., vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijcds/100196.

[69] J. Ramkumar, R. Karthikeyan, and M. Lingaraj, "Optimizing IoT-Based Quantum Wireless Sensor Networks Using NM-TEEN Fusion of Energy Efficiency and Systematic Governance," in Lecture Notes in Electrical Engineering, V. Shrivastava, J. C. Bansal, and B. K. Panigrahi, Eds., Springer Science and Business Media Deutschland GmbH, 2025, pp. 141–153. doi: 10.1007/978-981-97-6710-6_12.

[70] R. Jaganathan, S. Mehta, and R. Krishan, Bio-Inspired intelligence for smart decision-making. IGI Global, 2024. doi: 10.4018/9798369352762.