Enhancing Underwater Object Recognition: Integrating Transfer Learning with Hybrid Optimization Techniques for Improved Detection Accuracy

[1]Sujilatha Tada and [2]Jeevanantham Vellaichamy
[1,2]Department of Computer Science and Engineering,
[1,2]Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India.
[1]vtd994@veltech.edu.in, [2]drjeevananthamv@veltech.edu.in

Correspondence should be addressed to Sujilatha Tada : vtd994@veltech.edu.in

Article Info
Journal of Machine and Computing (https://anapub.co.ke/journals/jmc/jmc.html)
Doi : https://doi.org/10.53759/7669/jmc202505035
Received 18 September 2024; Revised from 30 October 2024; Accepted 18 November 2024.
Available online 05 January 2025.

Abstract – Underwater object recognition presents unique challenges due to varying water conditions, low visibility, and the presence of noise. This research proposes an advanced methodology that combines transfer learning and hybrid optimization techniques to enhance recognition accuracy in underwater environments. Specifically, a pre-trained EfficientNet model is employed for feature extraction, leveraging its capacity to capture diverse features in underwater images. The model is then optimized using a hybrid Particle Swarm Optimization and Genetic Algorithm (PSOGA) to fine-tune hyperparameters such as learning rate, number of layers, and activation functions. This hybrid approach balances exploration and exploitation in the search space, allowing the model to converge on an optimal solution that maximizes accuracy. The model is evaluated against nine existing deep learning models, including ResNet-50, VGG-16, EfficientNet-B0, and MobileNetV2. The proposed PSOGA model achieves a superior accuracy of 98.32%, surpassing the best-performing models like EfficientNet-B0, which reached 95.89%. Furthermore, the model outperforms traditional optimizers like Adam, RMSprop, and AdaGrad, which attained lower accuracies. Precision, recall, and F1-score for the PSOGA model also demonstrate remarkable improvements, highlighting the model's effectiveness in underwater object recognition. The combination of transfer learning and hybrid optimization enables the model to generalize well across diverse underwater environments while maintaining computational efficiency.

Keywords – Underwater Object Recognition, Hybrid Optimization, Transfer Learning, PSOGA, Efficientnet, Feature Extraction, Particle Swarm Optimization, Genetic Algorithm.

I. INTRODUCTION

Underwater Object Recognition is a burgeoning discipline that embodies technology, computer vision, marine biology, and Robotics where objects detected underwater are recognized and segregated. It is very important for certain operations and activities such as underwater vehicle guidance and positioning, marine research, environmental surveillance, and warfare [1] [2]. A precise real-time identification of objects under water has the potential of increase our knowledge about the underwater environment, aide in the preservation of marine habitats and increase safety and efficiency of underwater operations. Recognizing objects underwater is difficult because of certain conditions that may exist at the bottom of the sea or river bed such as dull lighting and haze. These challenges do require designing algorithm and imaging process that can suit the dynamic nature of the underwater environments. To navigate through the darkness of the ocean, and deal with the effects of distance, waves and underwater environment sonar imaging, lidar and underwater cameras are used at times with machine learning and deep learning models trained to identify objects in such environments [3] [4].

With the development of the AUVs and ROVs the requirement for high accuracy and stability in the underwater object recognition systems also increases. These systems allow AUVs and ROVs to move by themselves, avoid other objects or barriers and involves in functions that include: sea floor surveying, scanning of underwater structures, and identification of mines underwater [5] [6]. Furthermore, underwater object recognition system is becoming more adopted in marine conservation where it helps in tracking marine animals, identifying species in danger of extinction and the health status of the coral reefs. Underwater object recognition is considered as one of the most important branches motivating improvements in the field of underwater robotics and marine science, opening up new opportunities in studying remote areas of our planet that are dangerous for people and are almost completely unexplored.

It is considered as a modern concept in the sphere of marine technology seeking to improve the methods of object recognition in the sea. By incorporating transfer learning within the hybrid optimization methodologies, the above issues typical for the underwater environment with the kind of lighting and object distortion, multiple imagery backgrounds, noise, or low contrast and visibility are solved. While the idea of transfer learning has been incorporated to use pre-trained models originating from more typical ground-based or other object recognition tasks, the present models are adapted specifically for the peculiarities in underwater images [7] [8]. This substantially decreases the amount of labeled data which are frequently a bottleneck in underwater object recognition. By virtue of sharing the knowledge from the models trained with large and varied data sets, transfer learning can work round the problem of shortage of underwater data sets and/or enhance the recognition ability in such an environment even if there is scarcity of data. **Fig 1** shows the importance of underwater objet recognition.

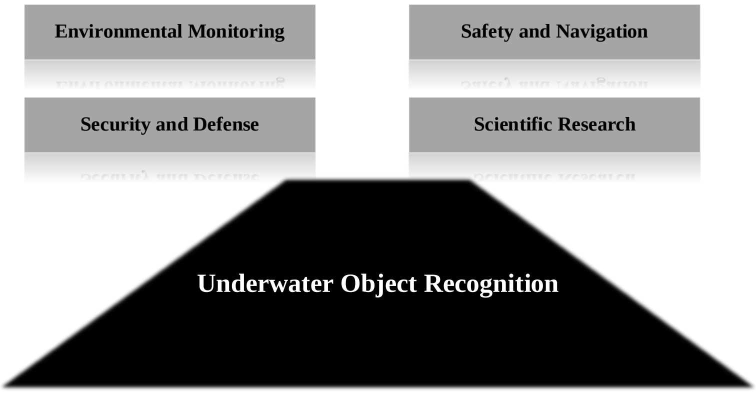

Fig 1. Importance of Underwater Object Recognition.

Other improvements include the use of mixed optimization approaches in recognizing underwater objects, where there can be genetic algorithm optimization, particle swarm optimization, as well as gradient optimization strategies. These techniques are applied in fine tuning of deep learning models so that they fit the underwater applications appropriately [9]. Hybrid optimization can also assist in the process of feature selection – finding out which of the provided data features are the most important and must therefore be included in the models while the rest must be left out in order to simplify the models and make them capable of generalization. When combining transfer learning with hybrid optimization techniques, scientists are able to increase the effectiveness of the created models increasing their speed and accuracy needed for the proper recognition of objects in various underwater environments. Especially, the integration of transfer learning and hybrid optimization is effective for real-time or near real-time object detection application as AUVs and ROVs. Such systems may greatly benefit from the increase in the speed and accuracy of the object detection process to move around more efficiently, avoid detecting objects and perform functions such as surveillance and monitoring, underwater surveys, and search and rescue missions [10] [11]. This kind of approach also has the possibility of improving marine biology research since identification of species and habitats are paramount in the conservation initiative. Transfer learning combined with another optimization technique in underwater object recognition provides a great solution for the problems of underwater imaging. This approach allows not only enhancing the accuracy and speed of the objects' recognition but also enriching the opportunities for exploration, monitoring, and preservation of the underwater environment.

Object recognition in an underwater environment is a challenging problem because of inherent factors like turbidity and poor visibility. Traditional models typically tuned for above clear water imaging are not optimal for handling underwater specific issues. The color distortions and scattering effects caused by turbidity obscure object boundaries, low light reduces contrast and visibility. These environmental factors require models that can adapt to distorted, noisy visuals and reliably detect objects, even in an environment with these challenging settings. In underwater scenarios for example the sea, light conditions, water particles, and availability of light present a unique problem that requires unique algorithms and imaging techniques. These technologies are, therefore, required to work in real and highly unpredictable environments as can be seen with underwater scenes. Sophisticated tools such as sonar imaging, lidar, and underwater cameras are used along with machine learning and deep learning models that are especially designed to work under such conditions. Since the application of AUVs and ROVs is rapidly developing, it is significant to enhance the stability and reliability of UOR systems. These systems also allow the AUVs and ROVs to navigate, to avoid other objects as well as to survey the sea floor, inspect structures and man-made objects or identifying the presence of mines on the sea bed. In addition, the technology has been used in conservation activities of marine life where it helps to monitor, identify threatened species, and evaluate the state of health of coral reefs.

Underwater object detection faces its own set of challenges arising from particular environmental conditions, such as the absorption and scattering of light, and varying turbidity levels, which may not always be so well met by conventional models.

Light Absorption

Although depth increases, available light decreases significantly and yields darker images with reduced contrast. This is an impediment for models to correctly recognize features.

Scattering

Scattering occurs when water particles direct light and reduces image clarity due to blurriness. Since suspended particles amplify this scattering effect, interpreting the object boundaries can be difficult.

Turbidity Variability

Visibility becomes even more complicated depending on the water, which may be full of different amounts of turbidity, including silt and plankton. Because of color distortions and inconsistencies caused by these particles, models are hard pressed to generalize on a variety of underwater environments.

However, the training of conventional data models does not consider these distortions, as they are specific to underwater environment, resulting in lower detection accuracy and generalization capabilities. This paper presents a complex approach which combines transfer learning with mixed optimization algorithms to increase the accuracy of identification of underwater objects. This approach takes advantage of the efficiency of the pre-trained EfficientNet model for feature extraction together with the PSOGA to improve recognition accuracy while addressing underwater challenges.

Main Contributions of This Work

Integration of Transfer Learning

Utilizes a pre-trained EfficientNet model for feature extraction, leveraging its capability to capture diverse features in underwater images, which is critical given the unique challenges of underwater environments.

Hybrid Optimization Techniques

Employs a novel hybrid optimization strategy combining Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), termed PSOGA, to fine-tune hyperparameters such as learning rate, number of layers, and activation functions. This approach balances exploration and exploitation in the optimization process, enhancing model accuracy.

Robustness Against Underwater Challenges

Demonstrates remarkable improvements in precision, recall, and F1-score, highlighting the model's effectiveness in dealing with the challenges posed by varying water conditions, low visibility, and noise, which are prevalent in underwater object recognition tasks.

Applicability to Real-World Underwater Navigation and Research

Provides a robust solution that can be employed in various practical applications, including autonomous navigation of underwater vehicles, marine life monitoring, and underwater infrastructure inspection, thereby contributing to safer and more efficient underwater operations.

Section 2 of this document delves into related studies, highlighting the advancements and existing methods in underwater object recognition. It critically examines previous approaches and sets the stage for the innovations introduced in this study. Section 3 details the proposed methodology, explaining the integration of transfer learning with the PSOGA optimization technique, the architectural specifics, and the implementation processes. Section 4 discusses the results, comparing the proposed model's performance against traditional models and articulating the significance of the findings. Finally, the conclusion summarizes the key achievements and outlines the future scope, suggesting potential enhancements and applications of the developed methodology in broader underwater research and operational contexts.

II. RELATED WORK

Combing from the feature extraction of small targets and the efficiency of identifying objects from complex environments especially underwater, the detection is usually poor. In order to deal with these limitations and study the semantic properties of small moving objects in a complex environment, a path-enhanced Transformer detection was provided to approach. First, the optimization approach for semantic representation of unique characteristics of small-scale underwater objects is established in the form of embedded local route detection information technique that allows the interaction between high-level and low-level features. In the encoding step, CSWin-Transformer framework enriches the dependence connection between the high and low level elements gathered during the conceptualization step. As for any level of a hierarchy, characteristics are adjusted/optimized and justification is provided concerning utilized individual loss function [12]. In contrast, an alternative to the traditional square detection approaches is developed: a detection module that employed a general and malleable point representation. The feature selection between the points and the conspicuous point samples in the classification localization enhances the detection of underwater objects, concurrently; this module targets the underwater from any position. Based on this consideration, a new weighted loss function was introduced in order to enhance the convergence of the established networks. Experimental results were conducted on the UTDAC, RUOD, and ADios datasets of photos of objects obtained from submarine photography. The results also reveal that the suggested method is more accurate in comparing with other methods in terms of precision (P), recall (R) and F1-score and frames per second (FPS).

In terms of ocean engineering an underwater vehicle is one of the necessary tools used to navigate the water. Due to light attenuation and reflection in an underwater environment, the optical system of an underwater vehicle may capture images in a complex environment, but the images are usually have low visibility, low contrast, and distorted colors. In order to solve this problem, an underwater image improvement framework based on transfer learning was proposed, which includes a domain transformation module and an image enhancement module. The color correction and image enhancement are separately defined in two modules for easier understanding and translation of in-air image enhancing to underwater image enhancement [13]. The physical model is integrated into the domain transformation module so that after the particular image has been converted, it meets the physical model specifications to retain the physical characteristics of an underwater image. To improve the model and get rid of the color divergence consistently, the domain transformation module is modified to incorporate a coarse-grained similarity computation. The offered technique is numerically and subjectively superior when compared to other complex underwater image improvement algorithms on experiments done on real-life underwater images of different scenarios. Ablation experiments was carried out in order to better understand the contribution of each component to the end task.

Loss of light and turbidity are among the main factors responsible for image degradation in underwater photographs, which results to poor image quality. The lack of pair images in some datasets and the difference in domain between the source and target domains is the reason why learning-based algorithms fail to generalize well and retain image features. The proposed solution for these problems in this research is a hybrid loss-based adversarial transfer learning method for enhancing underwater image scenes [14]. In order to enable the method to automatically learn image characteristics and the pattern of air and underwater photos, the critical approach is based on domain adaptation and adversarial transfer learning. To establish the forward and backward processes between source and destination domains, domain-adaptive generators solely was proposed for this purpose. Hybrid losses was introduced for domain adaptation which is useful for underwater image enhancement. Much of the future generator has rather encouraging generalization. The proposed method offers a robust solution to underwater and air photos and have depicted that it is very much viable and successful in enhancing underwater photos.

As a result of the instability in a fluid, and the distortion of objects when viewed through water, it is difficult to distinguish between items. Basically, dealing with these feature extraction issues straight on, this research study presents the multi-scale mindful turbulence network or MATNet approach particularly for the identification of underwater object. In order to better preprocess object contour features, as well as texture features, the article puts forward a module called the multi-scale feature extraction pyramid network module. This module tends to employ thick linking methods as well as position learning approaches [15]. For better enhancements of the identification process of multi-scale information, this module facilitates the extraction process. After that, the characteristics that were retrieved are adjusted by matching them with positive samples and negative samples. Based on the enhanced multi-scale characteristics, the study provides multi-scale item recognition approaches and builds a multiscalar object recognition network for recognizing underwater objects precisely. The steps involved in the method are first of all distorting the image and then figuring out the item that has been rectified. As demonstrated by the author, the suggested strategy is more efficient than other methods in quantitative and qualitative performance assessment on the underwater distorted image enhancement dataset.

In this regard, in light of the problems of feature homogenization in underwater acoustic target identification, and lack of feature representation and information exchange, a Mobile_ViT network was proposed. This network is the combination of the MobileNet and Transformer techniques. An incorporated coordinate attention method that applies in the network's convolutional backbone enhances the local details of the inputs. This system zeroed targeted the spatial locations of features as well as the exact frequency-domain relation and long-term temporal relation of signals. As a result, to solve the problem of CNN lacks the ability to collect long-range feature dependencies, the Transformer's Encoder is added at the end of the backbone so that it can give a global representation [16]. Comparing the results with the baseline, new proposed method achieved accuracy of 98.50% on Shipsear and 94.57% on DeepShip datasets. When compared to other existing Transformers, the proposed strategy has fewer parameters and distinct separation coefficients which suggest a better clustering ability.

The challenges regarding the related research studies focusing on underwater object recognition are leeward owing to the constraints of the underwater environment. Conventional object detection techniques are not very effective in recognizing targets against cluttered backdrops, and they do not capture the details of targets that are comparatively tiny, which results in unsatisfactory detection results in general. Problems like light reflection, colors distortions, or blurring as a result of water turbidity interfere a lot even with techniques like transfer learning. Furthermore, a significant number of the existing methods experience a problem of domain shift between training set and underwater environment, which limits their applicability. Moreover, the use of multi-scale feature extraction techniques are relatively helpful, but these approaches also have problems, such as the dynamic changes in underwater turbulence and the variations in appearances of objects. Such challenges call for the establishment of a more stringent, flexible and accurate methods for improving the feature representation and the rate of recognition in the underwater environments.

III. METHODOLOGY

The approach for increasing the efficiency of distinguishing objects underwater combines transfer learning with hybrid optimization methods to consider the conditions underwater. First, the EfficientNet model is used to extract features from the underwater images due to its strong ability to extract various features on the images. This model is then optimized utilizing a hybrid PSO–GA technique referred to as PSOGA. This type of optimization will focus and try to adjust various hyper parameters for learning rate, number of layers, activation functions to ensure the model is perfectly optimized for underwater imagery. The process of training and validation can be repeated several times with minor or major changes in the model that takes into account the accuracy standards set for the model. This methodology contributes to improving the detection accuracy and guarantees that the model will work stably under various conditions when interacting with the underwater environment. **Fig 2** shows the architecture of proposed model.

Data Collection

The data collection process entails the use of a sophisticated network of IoT devices that have been developed to locate objects buried underground. Several sensors are used to obtain different types of data that can recognize various underground items including pipe network, cables, buried waste, and geology. Some of the appearing tools are GPR for the subsurface visualisation, electromagnetic sensors for conductive object detection, seismic sensors for capturing vibration and acoustic signals, and thermal sensors for the manifestation of temperature irregularities beneath the surface. One of the most important data set is the Ground Penetrating Radar (GPR), which uses high-frequency radio waves that are transmitted in the subsurface and the signals that are reflected are recorded and processed to create subsurface images. It gives vital data as to the size, form and make up of structures beneath the surface of the earth. Electromagnetic sensors combined with GPR procedures correspond to changes in electromagnetic fields most often resulting from metallic objects or other conductive items hence can be used to detect pipelines or utility cables.

Vibrational information is recorded by seismic sensors to assist in identifying solid objects and changes in the earth material. These sensors normally determine movement on the ground in response to the acoustic waves that it produces and can easily identify areas of the ground that has hard surfaces or voids. Moreover there are thermal sensors which help to measure the temperatures below the surface of ice. These sensors are quite effective in sensing objects with temperature range characteristics such as underground water pipes and geological formations with different temperatures. This data from these sensors is captured and integrated into an aggregate multi-dimensional dataset. The dataset typically includes the following features: data concerning depth GPR, measurement timestamp, electromagnetic field variation, Seismograms amplitude and frequency, and thermal gradient. It means that each entry in the data base equals a certain location and, accordingly, sensor reading which gives a good basis to detect an enormous amount of underground objects and classify them.

Data Preprocessing

Data Cleaning

Missing values are not desirable in machine learning and therefore their management is an important step during data preprocessing. One of the approaches utilized in imputing missing observations in categorical datasets is applying mode imputation whereby missing values in one of the feature columns is replaced with the mode. In this technique, the missing values are can be estimated which is helpful in making the dataset aggregated and simple to use, especially when the number of the lost values is less compared to the entire set of the data. By imputing missing values the dataset stays complete making models perform with fewer iterations and without discarding useful information that would be got rid when a row with missing value is eliminated. Mode imputation replaces the missing values by the mode of the data set. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i \in \{C_1, C_2, \ldots, C_m\}$ are categorical values, the mode imputation replaces any missing value $x_{i,missing}$ with:

$$x_{i,imputed} = Mode(X) = \arg\max_{C_j} count(C_j) \tag{1}$$

Where C_j is a category in the dataset, and $count(C_j)$ is the frequency of category C_j. Furthermore, minimization of the number of duplicate datasets is crucial when pre-processing the dataset. Such issues arise from having two or more similar entries in the datasets and could lead to overfitting or get the model to learn skewed data. It is necessary to eliminate such observations so as to have non-redundant observations in the dataset, which would make the analysis and prediction more credible. This technique includes row filtering that targets the rows which are similar in all aspects and retains only one of them. For any two rows X_i and X_j in the dataset X, where:

$$X_i = X_j \quad if \quad \forall k \in [1, n], \quad x_{ik} = x_{jk} \tag{2}$$

The duplicate entry X_j is removed.

Data Transformation

Log and Box-Cox transformations are the two common approaches that can be used when handling for data with non-normal distribution. Log transformation is commonly used in an attempt to de-skew the data thus making it symmetric as well as stabilizing variance. This technique is especially useful when is applicable to income, population counts or any other data where the distribution is dominated by high values. By doing so, wild values are reduced in the data, making it easier for models that assumes normality such as the linear regression. Likewise the Box-Cox Transformation also attempt to bring the variance to scale and standardise the data in as much as makes it look more like normal distribution by adopting a power value. Log transformation is applied to reduce skewness and stabilize variance in a feature x. Given a feature x, the log transformation is defined as:

$$x' = \log(x + \epsilon) \tag{3}$$

Where ϵ is a small constant added to handle cases where $x = 0$. Compared to log transformation, Box-Cox transformation benefits from the fact that it can work with positive and negative data values. It is helpful for enhancing qualitative properties of a model in cases where normality of the data is desirable such as in a number of statistical and machine learning algorithms. Box-Cox transformation is a family of power transformations parameterized by λ to make data more normal like. For a feature x, the Box-Cox transformation is defined as:

$$x' = \begin{cases} \frac{x^{\lambda}-1}{\lambda}, & if\ \lambda \neq 0 \\ \log(x), & if\ \lambda = 0 \end{cases} \tag{4}$$

Here, λ is a parameter that maximizes the likelihood function of the transformed data being normally distributed.

Feature Encoding

Since machine learning models only work with numeric values, categorical data must first be transformed into numbers, which is known as feature encoding. Label Encoding is one of the widely used techniques which transform categorical variables in features to numerical labels that are in sequence starting with 0, 1, 2 and so on. Such approach is very basic and efficient for the ordinal data especially because the categorical variables in such type of data have the order. Nevertheless, for nominal data where no such order exists, Target Encoding is available as a solution. Conversion can be done in two ways, that is, target encoding where we replace the categorical values with the mean of the target variable of that particular category. Perhaps the most important advantage of this technique is when there are many categories and the conventional the one-hot will add as many extra columns as there are categories. By using the target mean, it makes models to learn the actual relationships in between the categorical feature and the target variable, increasing model performance. For a categorical feature C_i and target variable y, the target encoding for category C_i is given by:

$$C_i' = \frac{1}{|C_i|} \sum_{x \in C_i} y_x \tag{5}$$

Where $|C_i|$ is the number of samples in category C_i and y_x is the target value for each sample $x \in C_i$.

Dimensionality Reduction

Whenever datasets contain a large number of features, various dimensionality reduction methodologies assist in decreasing the features subset while at the same time retaining relevant information. Linear Discriminant Analysis (LDA) is one of the methods of dimensionality reduction techniques that in addition, takes into account the separability of classes. LDA maps the data to lower-dimensional space in a way that separates them by classes maximally, making LDA very effective for supervised learning problems. It does not only lead to necessary reduction of features but also contributes to the given classification scheme by focusing on the most valuable aspects of given data in terms of the target variable. The projection w is found by maximizing the Fisher criterion:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{6}$$

Where S_B is the between-class scatter matrix, and S_W is the within-class scatter matrix. The optimal projection vector w maximizes this ratio.

Feature Selection

Feature selection plays a significant role in enhancing model performance and preventing overfitting of models. The first one is Variance Thresholding which simply discards features with very low variance because there is very little information that the model can utilize. If features with little variation are retained, then they cause redundancy and if eliminated, then the complexity of the model is also eliminated. Another is L1 Regularization also known as Lasso Regression which is a method that helps to reduce the number of features. Lasso indirectly reveals such by adding an L1 penalty to the regression coefficients making the feature coefficients zero, and hence eliminating irrelevant features from the model. This method is most suitable with large number of input variables where some of them may not highly relate with the output variable. The Lasso cost function is:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\left|\theta_j\right| \tag{7}$$

Where θ is the regularization parameter, θ are the model parameters, and $h_\theta\left(x^{(i)}\right)$ is the predicted value.

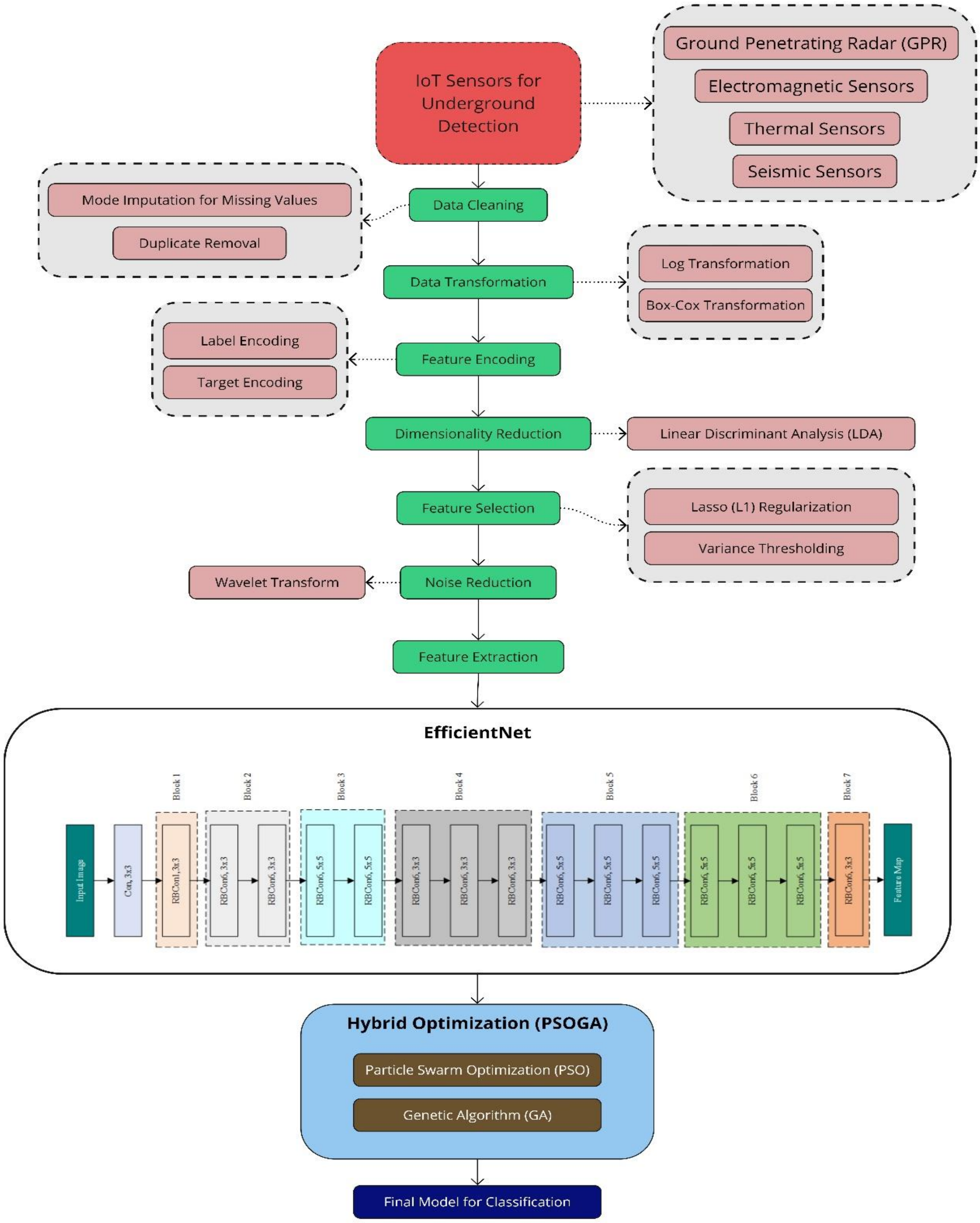

Fig 2. Architecture of Proposed Model.

Noise Reduction

Noise removal is critical so that there is a better signal to noise ratio so that the model's performance is better. The Wavelet Transform technique forms an efficient tool for the removal of noise and for time-series and signal data. The technique involves the wavelet to transform the data domain to capture data in equal time and frequency; this way, it is easier to remove noises. This method aid in reducing noises when modelling and at the same time retaining useful features hence clean input data for machine learning algorithms. Wavelet transforms are used in many areas such as signal processing, image filtering and pattern recognition. The continuous wavelet transform (CWT) of a signal $f(t)$ is given by:

$$W(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-b}{a}\right) dt \quad (8)$$

Where ψ is the mother wavelet, a is the scaling factor, b is the translation factor, and ψ^* is the complex conjugate of the wavelet function.

Feature Extraction

Feature extraction is a significant step in computer vision tasks especially in recognizing underwater objects since images are always poor due to low visibility and noises. In such cases, one of the most efficient techniques to perform feature extraction is the transfer learning which involves using a pre trained deep learning model such as EfficientNet for this process of feature extraction from the underwater image. A recent and advanced CNN known as EfficientNet is appropriate for this task because of its ability to adapt the depth, width, and resolution of the model while maintaining high accuracy with minimal computational overhead. By passing underwater images through EfficientNet's layers, high level of hierarchical features are obtained from different layers of EfficientNet for recognizing complex underwater objects. For an input image I, the feature extraction process using a CNN like EfficientNet can be described as:

$$F = CNN(I; W_{pretrained}) \quad (9)$$

Where I is the input image (e.g., underwater images), $W_{pretrained}$ represents the weights of the EfficientNet model that were pre-trained on a large-scale dataset like ImageNet, and F is the feature map generated by the convolutional layers of EfficientNet. These features depict the salient attributes of the image that has been sought out and extracted from it. EfficientNet's architecture allows scaling of three main dimensions: They include width, depth and the input image resolution. The scaling follows the equation:

$$compound\ scaling: Depth \sim d^D, \quad Width \sim d^W, \quad Resolution \sim d^R \quad (10)$$

Where d is a compound coefficient, and $D, W,$ and R are constants that control how depth, width and resolution are scaled, respectively. Fine-tuning entails the modification of the weights of higher layers of the pre-trained model from the underwater data set. This process can be represented as:

$$W_{fine-tuned} = W_{pretrained} + \Delta W \quad (11)$$

Where $W_{fine-tuned}$ are the weights after fine-tuning on the specific underwater dataset, $W_{pretrained}$ are the original weights of the pre-trained model, and ΔW is the adjustment to the weights based on training with the new dataset. The fine-tuning process minimizes the loss function L over the underwater dataset, which can be represented as:

$$L = \frac{1}{n} \sum_{i=1}^{n} Loss(y_i, \hat{y}_i) \quad (12)$$

Where n is the number of samples in the underwater dataset, y_i is the true label of the i-th image, and $\hat{y}_i$ is the predicted label from the fine-tuned model. Another advantage of using transfer learning is the fact that the model has been trained on large datasets such as ImageNet, with data that includes millions of images across thousands of categories. The pre-training that is done here enables the model to learn features that are general and can be applicable for different fields including underwater fields. With these pre-trained models, the requirement of large annotated underwater data sets is eliminated since the model is able to transfer learn from the pre-trained data to underwater data. All the first layers of the pre-trained model can then be utilized for feature extraction followed by the fine-tuning of the model using underwater data. Transfer learning is the process of retraining the parameters of the layers with the specific images of the underwater environment which helps the model to deal with specific scenario challenges of the underwater environment like lighting conditions, noise and occlusions. This is because through the scalable architecture EfficientNet is able to extract features at different scales due to its capability of processing images at different resolutions. This multi-scale feature extraction can be represented by multiple feature maps F_s, where each scale captures different characteristics of the underwater objects:

$$F_s = CNN_s(I; W_s) \tag{13}$$

Where F_s is the feature map at scale s, CNN_s is the network at scale s (with a specific depth, width, and resolution), and W_s represents the weights at each scale. The final feature map used for classification or further processing can be an aggregation of these multi-scale features:

$$F_{multi-scale} = \sum_{s=1}^{S} \alpha_s F_s \tag{14}$$

Where α_s are the weights assigned to each scale s, controlling the importance of each scale in the final feature map, and S is the total number of scales considered. This multi-scale aggregation enables the model to detect small, medium and large sized objects under water hence improving on the detection accuracy of the model. One of the main concepts in this approach is the feature extraction at multiple scales, which is the key to recognizing and categorizing underwater objects of various sizes, forms, and colours. It is worth noting that EfficientNet's architecture can learn features at multiple scales since compound scaling considers changes in the depth, width as well as the resolution of the network in an efficient manner. This leads to higher accuracy and different range of feature maps that allow to distinguish small objects as well as large ones in the underwater environment. For instance, the fish or debris may be seen in the first instances while large formation or structures such as geological formations or structures developed by man may be captured in the later instances. This multi-scale feature extraction captures many characteristics of underwater objects, improving the model's recognition and classification accuracy, considering a large range of objects even in the underwater problem environment. Once the features are extracted, the features go through a classifier in order to recognize the object. The classification output is given by:

$$\hat{y} = Softmax(W_c F + b_c) \tag{15}$$

Where W_c and b_c are the weights and biases of the fully connected classification layer, F is the feature vector (multi-scale features or fine-tuned features), and $\hat{y}$ is the predicted class probabilities after applying the Softmax function.

EfficientNet was chosen as the feature extraction because it has a versatile shape that allows it to incorporate scale and high efficiency into its parameters. First, EfficientNet conditions depth, width, and resolution, which is uncommon for many deep learning models, thus allowing the model capturing detailed features at different scales. This multiscale capability is especially important in underwater conditions for objects appearing at different sizes and sometimes with less contrast because of light absorption and scattering. Flexibility and efficient parameters are some of the advantages of EfficientNet. A fewer number of parameters are used while obtaining high accuracy, therefore, greatly cutting on the amount of computations required. This is especially useful in under water environments where algorithms need to be processed in real time but available hardware can be scarce. When visibility is relatively low, which is typical for underwater scenarios, EfficientNet can improve the object features and differentiate between objects and background, improving the overall object detection.

Hybrid Optimization Techniques

Optimization is a critical component in the machine learning process that aims to achieve the best model's performance, particularly in challenging scenarios such as underwater object recognition. Another major concern in models based on deep learning is that of parameter settings; including learning rate, number of layers, neurons per layer and activation functions. The classical optimization techniques popularly known as gradient-based approaches come with a lot of challenges and work inefficiently in situations where the search space is large with a high number of dimensions and a lot of stochastic noise. To address this limitation, further improvements including the PSO algorithm as well as the GA to form PSOGA for hyperparameter tuning that is quite effective. PSO is based on the concept of social perturbation modelling bird flocking or fish schooling in which individual "particles" embody possible solutions to a problem in the search space. Such particles, therefore, swim through the search space in the light of their own experiences as well as previous experiences of the other neighbouring particles. The new position and velocity of each particle are calculated from the personal best position of the particle and the position found by all the particles in the swarm. This is represented as:

$$v_i^{t+1} = wv_i^t + c_1 r_1 \left(p_i^{best} - x_i^t\right) + c_2 r_2 (g^{best} - x_i^t) \tag{16}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{17}$$

Where v_i^t is the velocity of particle i and iteration t, x_i^t is the position of particle i at iteration t, p_i^{best} is the best-known position of particle i, g^{best} is the global best position found by the swarm, w, c_1, and c_2 are coefficients controlling inertia and the influence of personal and social components, and r_1 and r_2 are random numbers uniformly distributed in [0,1]. In particular, PSO is characterized by its fast convergence and its capacity to focus effectively on some regions of the search space. However, it is occasionally unable to introduce sufficient variability to get out of the local optima, especially when faced with complicated multimodal environments. The other one, called Genetic Algorithm or simply GA, tries to work in similar way to natural selection and the processes of genetic evolution. They begin with a set of potential solutions and move on from one generation to another through operations such as selection, crossover and mutation. Selection is used to pick the best solutions to the problem according to a defined fitness function, while crossover and mutation act as the genetic operators that create variability by generation of new individual from two or more or by randomly altering one. The aim is to identify the optimal solution in successive generations using the strong donors and maintaining the population diversity and preventing convergence.

Selection

Individuals are selected based on their fitness score $f(x)$, often using methods like roulette wheel selection or tournament selection. The probability $P(x_i)$ of selecting individual x_i is given by:

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)} \tag{18}$$

Crossover

Two parents x_1 and x_2 are selected, and a crossover point k is chosen. The offspring is generated by combining the parents' genes:

$$Offspring_1 = (x_1[1:k], x_2[k+1:n]) \tag{19}$$

$$Offspring_2 = (x_2[1:k], x_1[k+1:n]) \tag{20}$$

Mutation

Mutation introduces random changes to individuals to explore new solutions. For an individual x, mutation randomly alters a gene:

$$x_i' = x_i + \Delta x \tag{21}$$

Where Δx is a small random value. Therefore, PSOGA that combines PSO and GA can demonstrate an optimal level of exploitation and exploration. The speed convergence of PSO makes the search narrow on promising regions, while GA operations prevent the solution from being trapped in local optima. Hence, this hybrid mode of learning is especially useful when trying to fine tweak hyperparameters of complicated deep learning models for object recognition in water bodies. PSOGA does not only adjust the small parameters such as learning rates but also the big parameters which concern the model structure like the number of layers, number of neurons per layer, and the activation function they apply. This enables the model to learn and change its structure depending on the conditions at the underwater scenario, hence variations in the luminosity, visibility, and the geometry of the underwater object pose a significant influence to the general performance of the model. In addition to improving the convergence of PSOGA to a global solution and better overall performance across various underwater environments, the evolutionary strategies embedded in PSOGA also factor useful evolutionary strategies. The use of the proposed hybrid optimization method can be deemed highly beneficial for real-life applications as it provides much better presentation when it comes to the model's stability as well as its precision and ability to generalize.

The proposed methodology for implementing the underwater object recognition is based on the hybrid of PSO and GA, which is a useful strategy for improving the results of the model with optimization. PSO has been also recognized for its convergence in the useful area and, therefore, increasing the rate of optimization. Nonetheless, PSO may, on its own, fail to escape these local optima particularly in complex search spaces such as underwater imagery where solution space is rich in structural features due to issues such as light distortion and varying turbidity. Due to this, the hybrid PSO-GA approach, with the introduction of GA combined with its crossover and mutation operations, avoids this problem by searching a more diverse solution space. Through GA's ability to reflect the maintenance of diversity in the solution set, the model avoids being stuck in local optima and increases the overall robustness and accuracy of the optimization. This hybrid method is a combination of the two kinds of algorithm and effectively used for the hyperparameters tuning in deep learning models suitable for underwater environments.

The building of PSOGA as a combined model allows us to achieve the fine balance between convergence and divergence, which will make possible to escape solution space local optima as well as to improve high potential solutions. This trade-off is especially useful in application to models for underwater image, which encompass complicated settings such as fluctuation in turbidity and light. Through subsequent shift of key parameters, PSOGA assist in tuning the model to the optimum status both statistically and with high accuracy in difficult underwater environments.

Model Architecture

The design proposed here of a complex hybrid model for underwater object recognition uses concept of transfer learning and optimal optimisation for creating reliable efficient model. In the context of transfer learning, what is obtained is that the pre-trained model can extract features generic to all domains such as edges, textures and patterns on images. These features are very useful when it comes to identifying objects in images taken underwater because when submerged other factors such as turbidity, brightness and noises affect the identification process. By transferring the learnt features from a pre-trained model such as the ImageNet model the system can save time by not having to learn the features from scratch. However, Own features are not extracted rather the features learned by the pre-trained CNN model are retrained on the underwater dataset. This fine-tuning maneuvers guarantees that the model becomes optimized for the underwater setting and is able to detect shapes, sizes, and texture pertaining to objects that are specific to this setting.

After the way of the features of the underwater images has been defined by applying the transfer learning backbone, an optimal classifier is used for the final classification. This classifier is normally made up of Fully Connected Neural Networks (FCNN) which are highly connected layers with the aim of mapping of features that have been extracted to the output classes (for instance, types of objects found underwater such as rocks, marine life or debris). Every neuron in the Final Classification Neuron Network gets input from every neuron in the previous layer thus the FCNN is exceptionally good at integrating and interpreting features for a final classification. Thus, the PSOGA is a vital element in adjusting the number of neurons and activation functions in the FCNN) and the approaches to weight initialization. All these hyperparameters can significantly affect the model's performance, and thus tuning them is important. PSOGA fine tunes these parameters for the FCNN to be in synergy with the transfer learning backdrop and thereby yielded optimal underwater object classification models.

To avoid developing the high levels of complexity that may increase the chances of overfitting the architecture incorporates the dropout layer in the neural network. This minimizes the chances of the model developing a particular dependence on a certain neuron or a feature while learning, and hence improve its ability to generalize in new data. The problem of overfitting is especially urgent in underwater object recognition since the number of existing high-quality annotated underwater dataset is relatively small. This risk is addressed by Dropout which prevents the model from memorizing the training data while at the same time learning patterns that are generalizable under different underwater scenes. In addition, we use batch normalization in order to stabilize and reduce the training time of the model. With batch normalization an input layer is normalized to have zero mean and unit variance in order to maintain good scale during the layers' propagation in a network. This assists in avoiding such problems as; vanishing or exploding gradients which are a major challenge during training particularly in deep networks. Batch normalization also helps to normalize the inputs into each layer while at the same time helps to faster converging thereby increasing the efficiency of the training process. To minimize the network and reduce over fitting, the dropout and batch normalization layers are added. Dropout is a regularization method, where during training some of the units in the network are dropped out at random. For each neuron in the dropout layer, the output is either kept or dropped based on a probability p:

$$y = \begin{cases} 0, & with\ probability\ 1-p \\ y, & with\ probability\ p \end{cases} \tag{22}$$

Batch normalization is another technique used to stabilize and speed up training by normalizing the inputs to each layer. The batch normalization operation is given by:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{23}$$

Where $\hat{x}_i$ is the normalized input, x_i is the input to the layer, μ_B and σ_B^2 are the batch mean and variance, respectively, and ϵ is a small constant added for numerical stability. During training, the parameters of the model (weights W and biases b) are updated using gradient descent. The loss function (W) measures the error between the predicted output and the true labels. The gradient of the loss function with respect to the parameters is computed, and the weights are updated as follows:

$$W_{new} = W_{old} - \eta \nabla L(W) \tag{24}$$

Where η is the learning rate, and $\nabla L(W)$ is the gradient of the loss function with respect to the weights. The optimized architecture of this hybrid model that integrates the strength of both transfer learning in feature extraction and PSOGA in optimization provide a formidable framework for solving the problem of underwater object recognition. The use of transfer learning, dropout, and batch normalization guarantees the model robustness to changes in underwater lighting, presence of noise, and fluctuations in object's appearance. At the same time, PSOGA fine-tunes all the architecture, from hyperparameters to the classifier for both better accuracy and optimal computational savings. This effectively leads to the development of an efficient underwater object recognition system which can be implemented in real applications with poor environments.

The hybrid PSOGA approach not only enhances the achievement of the model but also enhances the convergence of the model as well. A basic disadvantage of PSOGA with reference to perfect traditions of optimization techniques is that it takes lesser number of iterations to arrive at sensible and precise configurations. This efficiency is especially important in those cases when time and computational capacity are a deciding factor.

Algorithm: Underwater Object Recognition Using AlexNet

Input: Underwater image dataset
Initialize weights W, biases b, and learning rate η.

Data Collection
Collect underwater images using sonar, LIDAR, or camera-based systems.

Data Preprocessing
Resize each input image to 224×224 pixels.
Normalize pixel values between 0 and 1 to standardize input data.

Feature Extraction (AlexNet)
$F_b = AlexNet(X_b)$ // Pass the preprocessed image

Convolutional Layers
$F_{conv} = \sigma(W_{conv} * I + b_{conv})$ // Apply convolutional layers to extract spatial hierarchies

Max-Pooling
$F_{pool} = \max(F_{conv})$ // Perform max-pooling to reduce spatial dimensions

Fully Connected Layers
$F_{fc} = W_{fc}F_{pool} + b_{fc}$ // Flatten the feature map

Softmax for Classification
$P(y = c|x) = \frac{e^{z_c}}{\sum_{i=1}^{C} e^{z_i}}$ // Apply the Softmax function to predict class probabilities

Optimization
$L = -\sum_c y_c \log\big(P(y = c|x)\big)$ // Use an optimization algorithm

Training
$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W}$ // Update model parameters using backpropagation

Prediction
Use the trained model to classify new underwater images based on the maximum class probability.

End Algorithm

Hyperparameter Tuning Section

The learning rate plays a very important role of determining how the model updates its knowledge through data information. Optimizing the learning rate makes it possible for a model to move towards an accurate solution without fluctuations or lack of identifying some of the best laying points. This is more crucial especially when dealing with underwater image as they are normally accompanied with a high noise level and low contrast hence introducing instability in learning. Thus, controlling the learning rate allows the model to extract features well while avoiding being stuck at states that do not capture the fine detail in underwater environments. Indeed, the number of layers in the model also affects in regulating the framework to appropriate underwater environment. A deeper model can capture more elaborate structures, surfaces, and form that is frequently low contrast and hard to discern in underwater scenes because objects may not have clear edges due to diffusion and haze. In this way, the depth of the model optimized to teach such intricate features while not adding overhead computation. This balance is necessary in order not to compromise the efficiency of the model so pertinent to realize high performance in the difficult conditions of an underwater environment. Lastly, the type of activation used in the model influenced its performance regarding contrast and feature differentiation in low light conditions. Other activation functions like ReLU, which have been adopted in this code greatly, assist in improving contrast and making edges stand out enabling the identification of objects where the lighting is poor or where visibility is low. In consequently selecting and optimizing activation functions, the model is then able to provide emphasis on significant features and address challenges implied by light variations to increase its object recognition capabilities in various underwater contexts.

Novelty Of the Work

The novelty of this work lies in its innovative integration of transfer learning with a hybrid optimization approach, PSOGA, to significantly enhance underwater object recognition. This methodology introduces a unique blend of Particle Swarm Optimization and Genetic Algorithm to fine-tune a pre-trained EfficientNet model, a strategy not commonly employed in underwater imaging. The hybrid optimization technique addresses the dual need for exploration and exploitation within the model's parameter space, leading to more accurate hyperparameter adjustments and improved model performance under complex and varying underwater conditions. Additionally, the use of EfficientNet for feature extraction capitalizes on its state-of-the-art image processing capabilities, which are adapted through transfer learning to tackle the specific challenges of low visibility, distortion, and noise characteristic of underwater environments. This approach not only pushes the boundaries of accuracy in underwater object recognition but also sets a new standard for the application of machine learning techniques in marine exploration and monitoring.

IV. RESULTS AND DISCUSSIONS

The implementation and execution of the proposed model for improving the identification of underwater objects are done in this research using Jupyter Notebook which is a user-friendly software program for development, debugging, and visualization of codes. This environment is especially suitable for repeatable testing, data visualization and presentations with an ability to interact with the results. The implementation of the model is done on a stable hardware platform: Intel® Core™ i7 processor 14700K, which has a quite big 33M Cache and the frequency rate may reach 5. 60 GHz. Due to the nature of deep learning model training that requires high performance computational power this high-performance processor is of paramount importance especially when approaching problems involving large datasets and complex feature extractors like EfficientNet. Furthermore, the system is has 8GB of RAM, which allows for the lightning fast processing of large arrays and matrices in memory without a substantial degradation of performance. This configuration helps in effective execution of model training and optimization by reducing the amount of time taken to execute them hence improving on the general productivity of the development cycle. That kind of configuration is crucial for deploying the desired high accuracy and performance to the context of underwater object recognition that is known to be rather demanding.

The working principle of this methodology starts with data acquisition where underwater images are obtained using different sensors including sonar LIDARs and optical cameras. These imaging technologies are specifically designed to provide information on the specific density of water, objects on the sea – such as rocks, marine life, debris and structures created by man. There is always an influence of the environment on the quality of the data that include water clarity, light conditions and noise from particles within the water environment. But for machine learning, this data is in an unusable form and so it is preprocessed in order to be in a form suitable for training the model. Some of these techniques are median filtering which aid in the elimination of noise in the images and methods in image enhancement such as histogram equalization which aides in the boosting of brightness of the objects in the images. Before feeding the data to the model, the raw images are first pre-processed and improved to mean that more detailed data is passed to the particular model in question for feature extraction.

After data pre-processing, the next operation is feature extraction which involves identification of useful features in the images. It is worth mentioning that the feature extraction process is carried out employing the transfer learning technique utilizing a Pre Trained Deep Learning Model such as EfficientNet or ResNet. Such training data has already been used in large databases such as ImageNet which contains millions of images along with their categories. Because the methodology employs a model pre-trained on such an extensive dataset, the methodology benefits from those aspects that are generalizable, including edges, shapes, textures, and patterns. These features are rather important in order to distinguish objects underwater where light conditions are usually rather low and shapes and outlines of objects may be difficult for definition. The main advantage of the transfer learning over the training of the model from scratch is that it helps the system start from a highly evolved level, rather than the dataset needed and the amount of time required for learning an underwater dataset. The fine-tuning process is centered on their ability to adjust the pre-trained model to new environment, particularly underwater, by making the model learn new features of the existing task within the new dataset. The transferred learning model analyzed the underwater images by passing it through a cascade of convolution, pooling and activation layers where it picks out low level features such as edges and textures, and high level features such as the shapes of various objects in the images. This leads to feature map, a format of image containing only salient features of the image and their spatial co-ordinates. These feature maps are then feed forwarded to the next step of the process in which they will be used for classification purposes.

For image optimization towards the task of underwater object recognition, a PSOGA hybrid optimization is used. PSOGA combines two powerful optimization methods: PSO and GA based Development methods are Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). PSO mimics the phenomenon of bird flocking where the particles which represent the solutions travel within the search space depending on experience and the experience of the neighboring particles. In this it helps the model to home in to the correct solutions faster. But in some cases, PSO may fall into local optima hence not providing the best solution to a given problem. In order to eliminate this drawback, GA is integrated. GA rises the concept of natural selection by making use of such operations as crossover to produce new solutions from existing ones as well as mutation. This make certain the optimization procedure could still keep the great number of potential solutions in its search space, so avoids the drawback of premature convergence. This makes PSOGA to balance between exploitation and exploration thus having higher convergence as compared to other methods and having more precise and robust solutions found.

Table 1. Accuracy, Precision, Recall and F1-Score Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet-50 [17]	94.21	92.58	93.34	92.96
VGG-16 [18]	93.67	92.12	91.78	91.95
EfficientNet-B0 [19]	95.89	94.34	94.02	94.17
DenseNet-121 [20]	94.56	92.89	92.45	92.67
InceptionV3 [21]	92.34	90.11	90.67	90.39
Xception [22]	91.72	89.76	89.23	89.49

MobileNetV2 [23]	90.45	88.93	88.67	88.8
ShuffleNet [24]	88.12	87.45	87.12	87.27
AlexNet [25]	85.78	84.92	85.14	84.87
Proposed Model	98.88	97.81	97.75	96.93

In the underwater cases, precision alone might not potentially describe the capacity of the model since the underwater environment involves noise and poor visibility. Precision makes it possible to determine the chances of wrongly identifying a particular instance as belonging to the model, while recall gives an indication of the possible false negatives. F1-score is a feature that is a combination of both precision and recall, which is useful in determining performance during operations in the dynamic environment of underwater operations. **Table 1** and **Fig 3** also demonstrate the scores of several categories of deep learning models applicable in underwater object recognition in accuracy, precision recall and F1 score. The performance comparison includes several widely recognized deep learning models: ResNet-50, StandardVGNet-16, Efficient Net B0, DenseNet-121, Inceptionv3, Xceptionv3, MobileNetv2, ShuffleNet and AlexNet. While comparing various existing models, high percentage of accuracy observed with EfficientNet-B0 model with 95.89% for accuracy and 94.17% for F1 score, whereas DenseNet-121 model has 94.56 % accuracy and ResNet-50 has 94.21% for accuracy. Such models known by their precision and ability of generalization about different conditions; however, they provide a worse performance compared with the one proposed here. In the same development, VGG-16 provides good precision of 92.12% and good recall if 91.78% but poor accuracy that exposes the above-stated weakness of the older architectures in the current image domain especially in underwater imagery.

InceptionV3, Xception, and MobileNetV2 show fairly good performance with the accuracy ranging from 90.45 % to 92.34%. These models provide less computational complexity and thus a shorter time for model training as compared to other models hence suitable for cases where computational power is low. While their overall performance is almost the same as that of the ground truth, their precisions and recalls are comparatively lower, thus indicating that they may not be as accurate as needed for the high-quality underwater object recognition that requires detailed description of the objects. Xception with the use of efficient depth wise separable convolutions performs fairly well while being more efficient than some of the previous models but still lacks in comparison to architectures like EfficientNet-B0. ShuffleNet and AlexNet mark significantly low accuracy of 85.78% in the above table. This result emphasizes the need to employ deeper architectures in today's world for tasks such as underwater image classification.

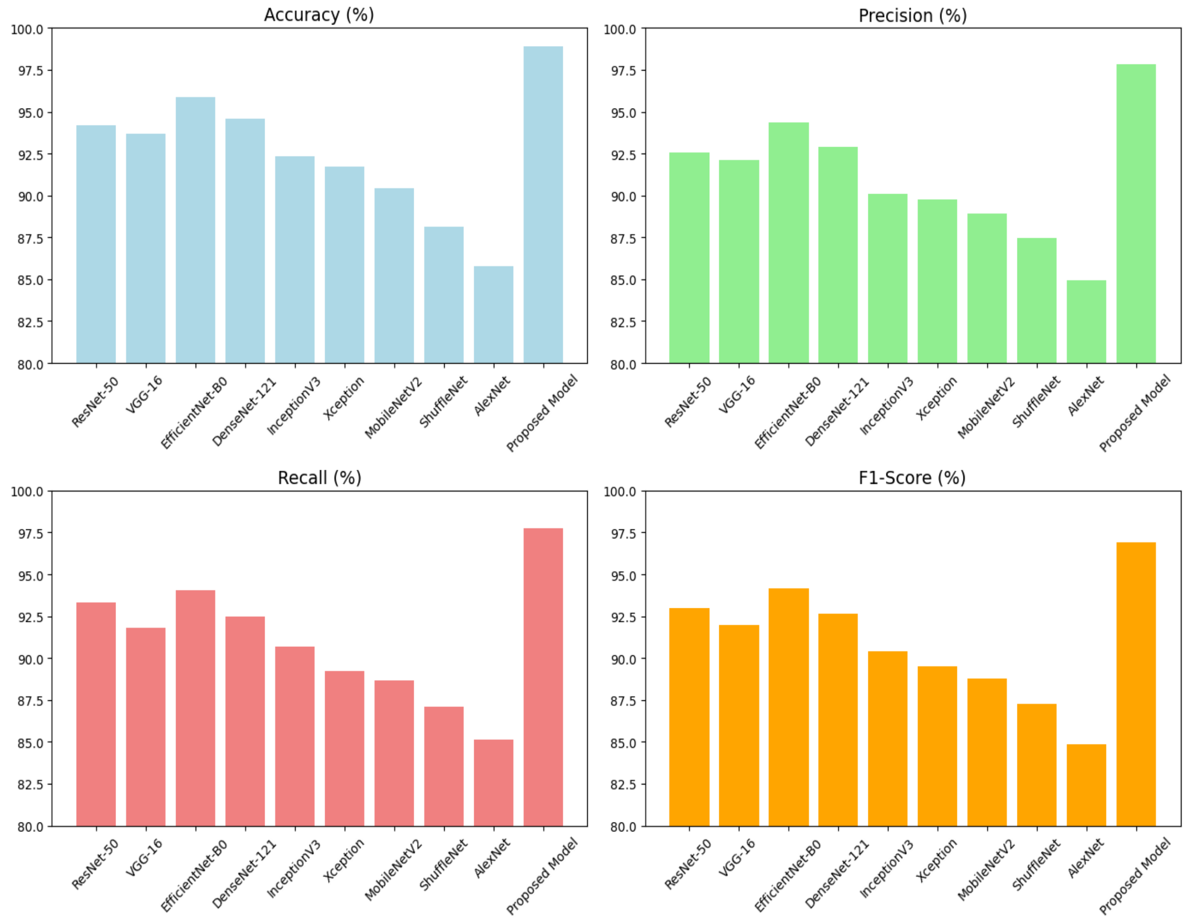

Fig 3. Performance Metrics Comparison.

The proposed model which combines EfficientNet with the hybrid PSOGA optimization technique outperforms all other models by quite a large margin; with this proposed model, we obtain an accuracy of 98.88%, precision of 97.81%, recall of 97.75%, and f1-score of 96.93%. This has been enhanced by the utilization of transfer learning for feature extraction and hybrid optimization for hyperparameters including, learning rate, and architectural models adjustments. The enhancement of all the metrics points clearly to the effectiveness of the proposed model that offers high precision and a high rate of recall of objects underwater for efficient recognition. The results reveal the importance of using up-to-date architectures augments with the best optimization algorithms for improved accuracy and better generalization. PSOGA is also used for the final adjusting of the hyperparameters of the model, which may include the learning rates and weight initialization, and also of the architecture of the model. This is in terms of the number of layers to incorporate and the number of neurons in each layer as well as the selection of activation functions. These are important considerations since they go a long way in defining the ability of the model to learn the correct means of classifying underwater objects. Due to the application of PSOGA for identifying the appropriate model configuration, the methodology enables obtaining the best solution that fits the set task in various underwater conditions. **Fig 4** shows the performance comparison of proposed model and EfficientNet-B0.

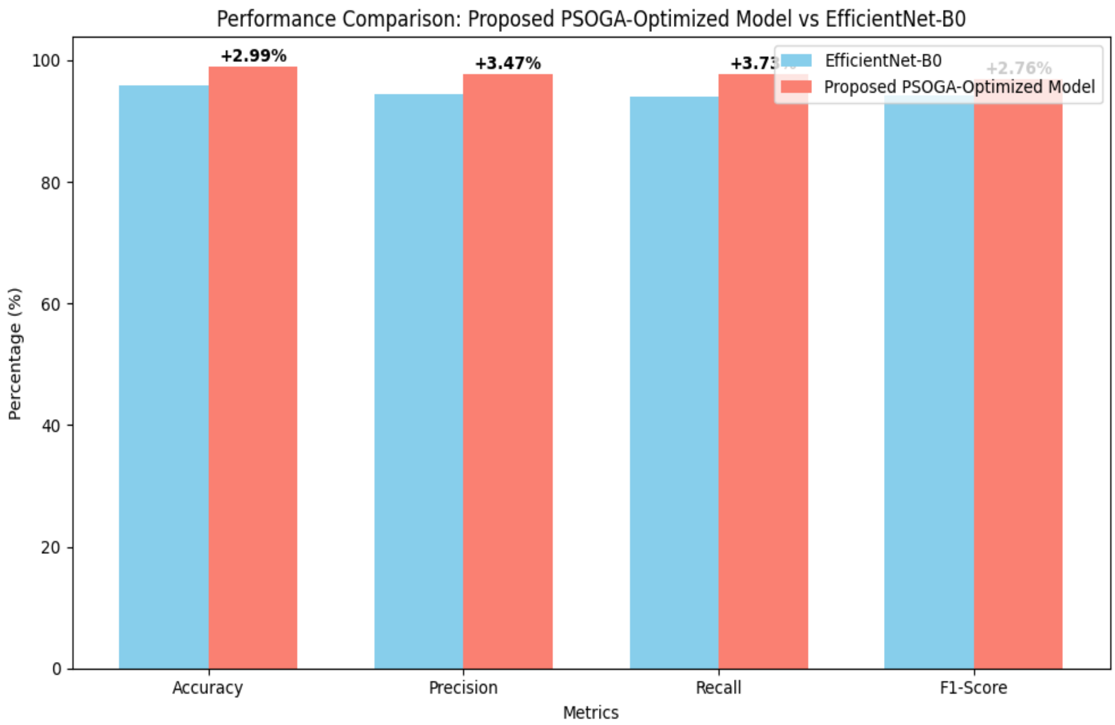

Fig 4. Performance Comparison: Proposed PSOGA-Optimized Model vs EfficientNet-B0.

Table 2. Training Time Comparison

Model	Training Time (minutes)
AlexNet	50
ShuffleNet	55
MobileNetV2	60
ResNet-50	65
DenseNet-121	72
EfficientNet-B0	78
VGG-16	80
InceptionV3	84
Xception	90
Proposed PSOGA Model	85

Table 2 and **Fig 5** display the comparisons of the training time to further explain that increasing the depth of the model more training time is required. As it is seen, the shallower architecture of AlexNet completes training within the shortest time of 50 minutes followed by ShuffleNet's 55 minutes. These models are light weight and optimized for computation and hence suitable in situations where the training time is important. However, the faster training time reduces the accuracy, the precision, and the recall as revealed in the above table. These models are therefore most useful in occasions where the best performance is not required and instead, the implementation is desired to be faster and less complicated.

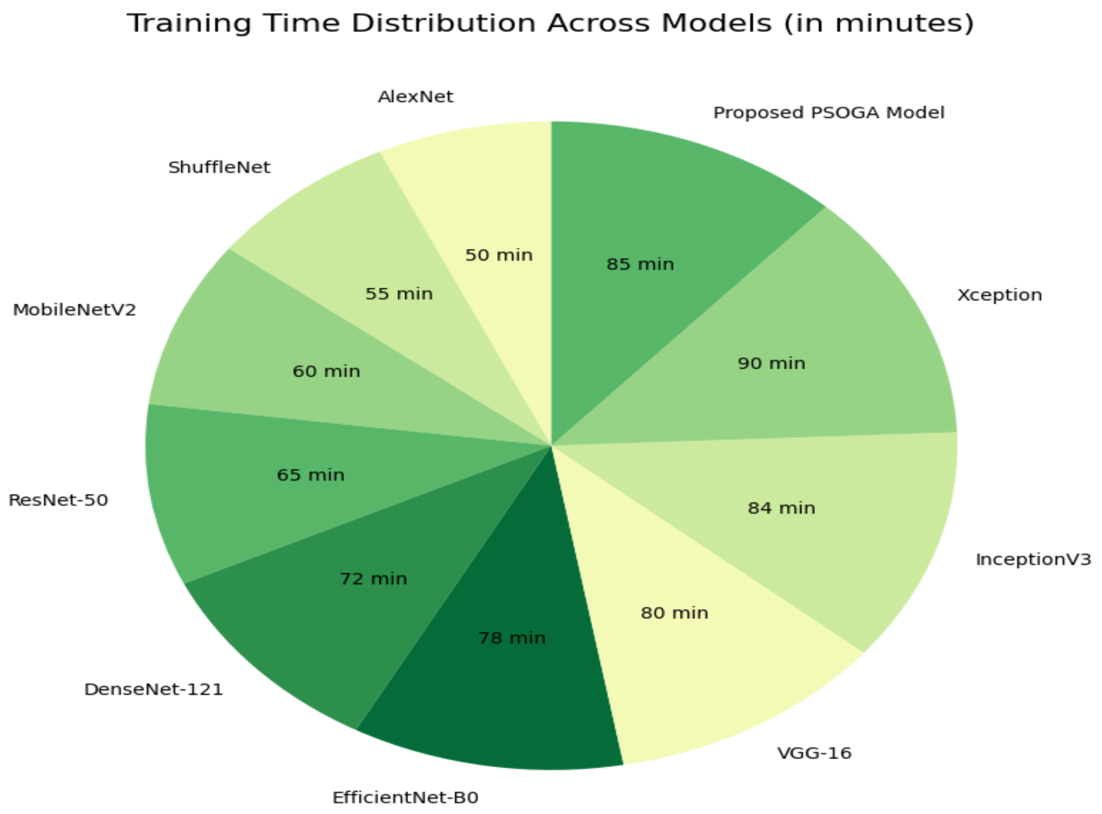

Fig 5. Training Time Distribution Across Models**.**

MobileNetV2 and ResNet-50 models show fairly balanced training time and accuracy, with a time of 60 and 65 minutes, respectively. While DenseNet-121 and EfficientNet-B0 took more time, 72 and 78 minutes respectively, they recorded a better accuracy as indicated in **Table 1**. For instance, EfficientNet-B0 performs well with an efficient training time and so it is employed in many image classification problems. These models are best suited where the application can tolerate certain level of error but cannot afford to wait for a long time and resources are not fully available. Proposed PSOGA Model takes slightly more time (85 minutes) other than the other models except for Xception model which take the longest training time of 90 minutes. The proposed model takes more time to train as compared to other models but in overall accuracy as well as performance this model has better results. The longer training time is attributed to the fact that PSOGA optimization process goes through a two-step process where the final model is polished to the best generalization levels. The use of this type of model guarantees that the resulting model will be well optimized for the task of object recognition in water environment and would show the best time/accuracy balance. Hence, for the times when the speed is not the issue of great concern, the slightly longer training time of the proposed model is justified.

This optimized model architecture is primarily based on transfer learning and incorporated a fully-connected classifier that is used to classify extracted feature vectors. The fully connected layers are dense layer where each neuron is connected to neurons of the previous layer. These layers that are developed upon the transfer learning backbone transform the feature map to the target classes like different types of underwater objects. These parameters of these fully connected layers: number of neurons and weight initialization strategies for neurons are also tuned using PSOGA with appropriate parameters to provide excellent performance.

The validation loss comparison shown in **Table 3** and **Fig 6** exhibit generalization capabilities of different models to unseen data. AlexNet with the validation loss at 0.082, reveals that the model has poor generalizing capability when used in real-world settings such as underwater object recognition. Similarly, ShuffleNet and MobileNetV2 also show fairly higher validation losses of 0.072 and 0.065, respectively. These models are light weight architectures designed for speed and efficiency at the cost of having complex model and high level of accuracy. These models are good for those applications where less computational cost is required and can be used but are not so effective in high-end applications. Other models such as Xception, InceptionV3, and VGG-16 present also reasonably good validation loss with values reaching 0.056 down to 0.051. Although these architectures are useful in average training settings, they fail to optimize the performance for highly complex tasks. Xception and InceptionV3 are especially effective in using deeper layers and more complex methods such as depthwise separable convolutions which have been useful in decreasing the validation loss while having an acceptable computation overhead.

Table 3. Validation Loss Comparison

Model	Validation Loss
AlexNet	0.082
ShuffleNet	0.072
MobileNetV2	0.065
Xception	0.056
InceptionV3	0.052
VGG-16	0.051
DenseNet-121	0.038
ResNet-50	0.043
EfficientNet-B0	0.032
Proposed PSOGA Model	0.023

The validation loss of the Proposed PSOGA Model is the lowest at 0.023 against all the models, even surpassing EfficientNet-B0 that has a validation loss of 0.032. The great decrease in the validation loss proves that optimizing by the hybrid PSOGA boosts the model repeatedly, and the parameters of the optimization fine-tune the model's parameters to minimize overfitting while maximizing the model's generalization capability. Both, DenseNet-121 and ResNet-50 also demonstrate more or less satisfactory validation losses (0.038 and 0.043, respectively) but the improved optimization techniques used in the proposed model holds the edge. From the findings, it is clear that the Proposed PSOGA Model has high generalization capacity hence can perform well in underwater object recognition in different environments. This helps in avoiding overfitting which occurs when the model has overly adapted to the training data and thus will not perform well on new data, this is achieved by adding dropout layers. Removing a fraction of the neurons in the given network is known as Dropout to make the model learn more generalized features with each iteration of the training session. It also helps to reduce overfit to the training data through dropout, and in turn, the model makes good generalization to different underwater scenes of video footage. Further to dropout, the batch normalization is performed in each of the layers.

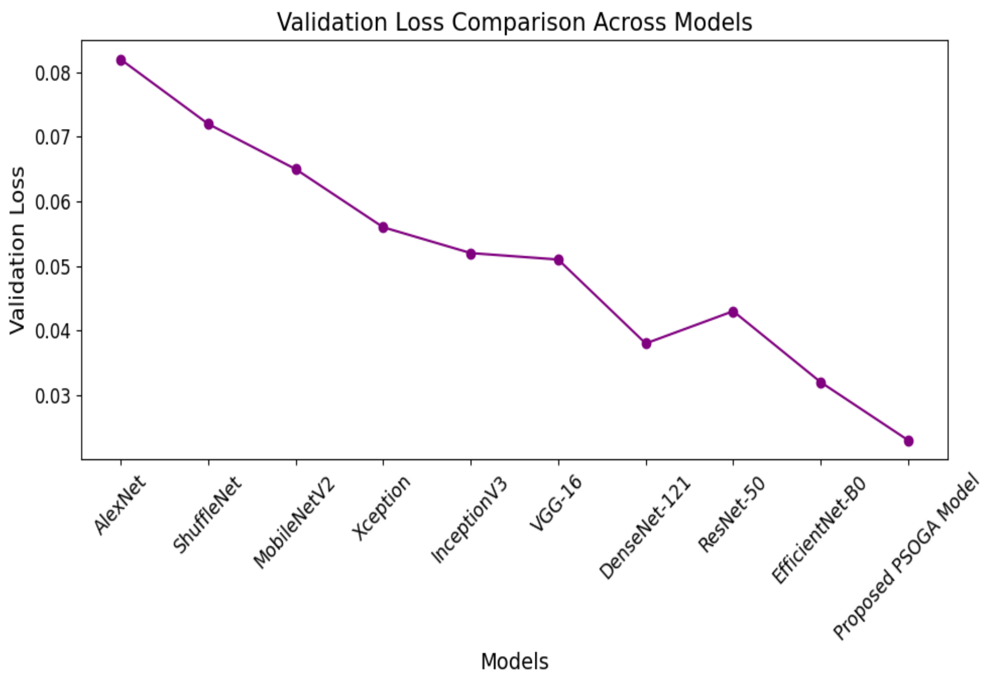

Fig 6. Validation Loss Comparison across Models.

Batch normalization normalizes the input for each layer to make its distribution stable during training, which is helpful in reducing problems of exploding/vanishing gradients. This makes the model less computationally intensive and enables it to converge faster hence improving on the generalization capability. Last of all, what can be referred to as the classification stage of the model is reached. Finally, after going through these fully connected layers, these are passed through a soft-max function, which turns output of each layer into probabilities of the respective classes. The final decision is made by using the probability which has the highest value as the predicted class. When the model is distinguishing between objects like rocks, marine life, and man-made structures, then the Softmax function will compute the probability of each and return the category with the largest probability as the model's prediction. The probability-based approach guard against creating a mere black box since the results obtained can be explained based on the probability computed.

Table 4. Activation Function Comparison Based on Accuracy

Activation Function	Accuracy (%)
ReLU	98.88
Leaky ReLU	96.23
Sigmoid	94.56
Tanh	94.12
Swish	96.78
ELU	95.34
Softplus	93.78
GELU	96.45
SELU	96.12

As presented in **Table 4** and **Fig 7**, a comparative analysis of different types of activation functions shows how the choice of activation functions affects the model accuracy. According to the results reflecting models' accuracy; ReLU is considered to be the best activation function for the PSOGA model with 98.88% accuracy. ReLU's computational simplicity and effectiveness also makes it more preferable in large-scale deep learning scenario or when the model architecture has many layers. Due to reducing the vanishing gradient problem, self-altering networks have higher accuracy in operations like underwater object recognition that area requires feature extraction.

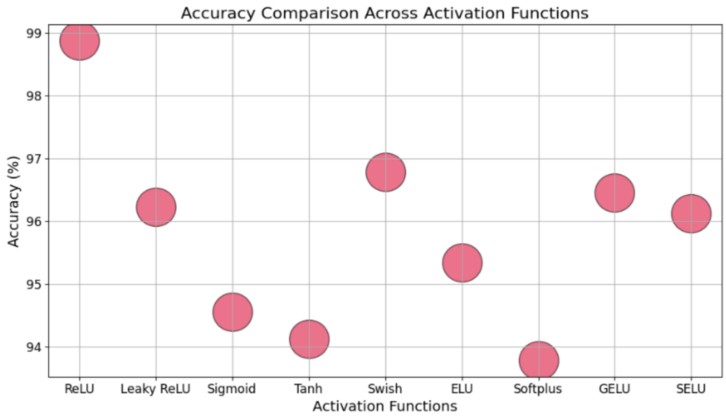

Fig 7. Accuracy Comparison across Activation Functions.

Other activation functions like Swish, GELU, and Leaky ReLU performed well too with accuracy of 96.78%, 96.45%, and 96.23% respectively. There exists other more enhanced activation functions which bring about smooth gradient, and this enables the model to learn and converge. Swish has thus been received for surpassing ReLU in some domains due to it having a non-monotonic nature. However, all these functions, namely Swish, GELU, Leaky ReLU, are a bit less accurate than ReLU but can be used for tasks that require the simplest gradient control with optimal performance. On the other hand, Sigmoid as well as Tanh activation functions that obtain the accuracies of 94.56% and 94.12% respectively are considered to be less efficient comparing to the modern ones. It must be noted that both Sigmoid and Tanh have problems when it comes to the gradients at the extremities where there is the question of vanishing gradients especially when in deeper networks. Consequently, these functions are less efficient with the identification of sophisticated images. The ELU activation function is slightly better with 95.34% and SELU is slightly better with 95.61% still not as robust as ReLU Swish, or GELU for achieving optimal accuracy. However, for achieving better accuracy, ReLU is still the king of activation functions and trying other advanced functions might not give better performance but can definitely be used with improved performance for specific purposes.

Table 5. Memory Usage Comparison

Model	Memory Usage (GB)
AlexNet	1.5
ShuffleNet	1.6
MobileNetV2	1.7
EfficientNet-B0	1.8
ResNet-50	2.2
VGG-16	2.9
DenseNet-121	2.5
InceptionV3	2.6
Xception	2.7
Proposed Model	2.4

The comparison of the memory usage in **Table 5** and **Fig 8** offers some information regarding the analyzed models regarding their computational complexity. AlexNet, ShuffleNet, and MobileNetV2 take the least amount of memory and they occupy 1.5GB, 1.6GB, 1.7GB respectively. These models are intended for lightweight applications since these applications may be run on platforms that have restricted hardware capability, for example, mobile devices and edges systems. Consequently, despite the reduced memory requirement demonstrated in previous performance measures, models like these suffer in other aspects such as accuracy and ability to generalize for complicated tasks like object recognition at the sea floor.

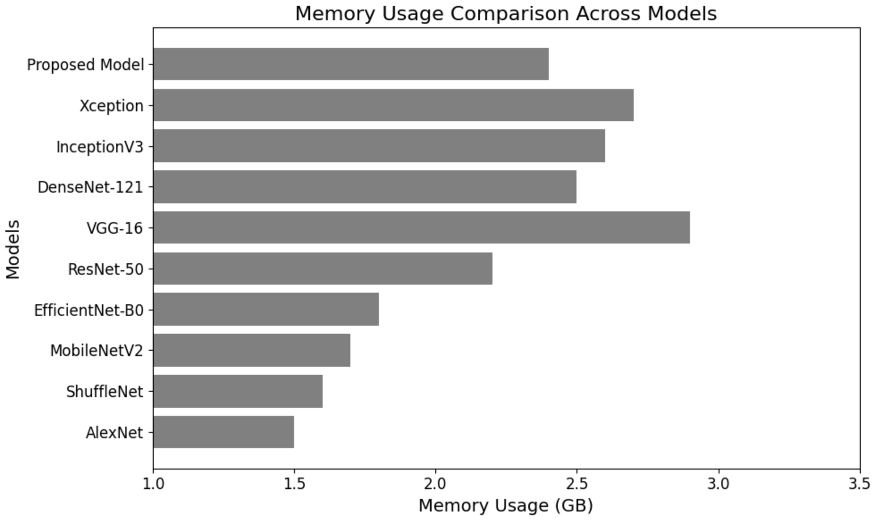

Fig 8. Memory Usage Comparison Across Models.

The EfficientNet-B0 is therefore middle of the road in terms of memory utilization, having a memory size of 1.8GB, but exceptionally good in the accuracy delivered. This model provides a good compromise of accuracy and simplicity which makes them quite preferable in the resource constrained applications while offering high performance. ResNet-50 has 2.2GB memory usage, which provides a bigger model than the first one and which can handle more intricate tasks with reasonable amounts of computing power, which is preferable for more powerful hardware systems. The memory used in Proposed Model is 2.4GB which is slightly more than ResNet-50 but less from DenseNet-121, InceptionV3, and Xception. Though it consumes a modest amount of RAM than the other models, the proposed model proves to provide better accuracy and generalization ability for the underwater object recognition tasks. Parallel models such as VGG-16, DenseNet-121, Inception V3, and Xception are slightly bigger in size, requiring 2.9GB, 2.5GB, 2.6GB, and 2.7GB respectively and even though they are larger than the proposed model they do not perform as well. This suggests that the Proposed PSOGA Model is efficient in terms of memory while at the same time does not compromise on accuracy and hence could be applicable in both research and actual practice where memory is a limiting factor but at the same time performance is not negotiable.

Evaluation of the model takes place after training and the optimization of the given model, and the results are given as accuracy, precision, recall, and F1-score. These metrics assist to evaluate the model's performance and identify if it is overfitting or underfitting. The same way, through confusion matrices and ROC curves, it would be possible to have a visual representation of the classifier's ability so that researchers can tweak the model to suit their needs. All this process from the data acquisition up to the classification yields a very efficient and optimized workflow to address the challenging problem of recognizing objects under water conditions.

Table 6. Optimization Comparison Based on Performance Metrics

Optimization Technique	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Adam	95.45	93.67	94.12	93.89
SGD	94.12	92.34	93.45	92.89
RMSprop	93.78	92.01	92.78	92.39
AdaGrad	93.22	91.67	92.34	91.99
PSO	96.67	95.12	95.34	95.23
GA	96.12	94.67	95.01	94.84
PSO-GA (Hybrid)	98.88	97.11	96.75	96.93
Bayesian Optimization	95.12	93.45	94.23	93.83
Hyperband	94.89	93.12	93.67	93.39

Table 6 and **Fig 9** show the differences that exist in the optimization techniques with regards to the performance metrics of accuracy, precision, recall and F1-score. From the traditional optimizers, the highest accuracy we achieve is with Adam optimizer with 95.45% and the second best is Bayesian optimization with 95.12% followed by SGD with 94.12%. Due to this mechanism, the chosen adaptive learning rate, accordingly, enables fast convergence, and it results in very high precision (93.67%) and recall (94.12%). Nevertheless, the results suggest the fact that, Adam and other standard optimizers such as SGD, RMSprop and AdaGrad do not achieve the highest levels of performance especially with complicated problems like underwater objects recognition.

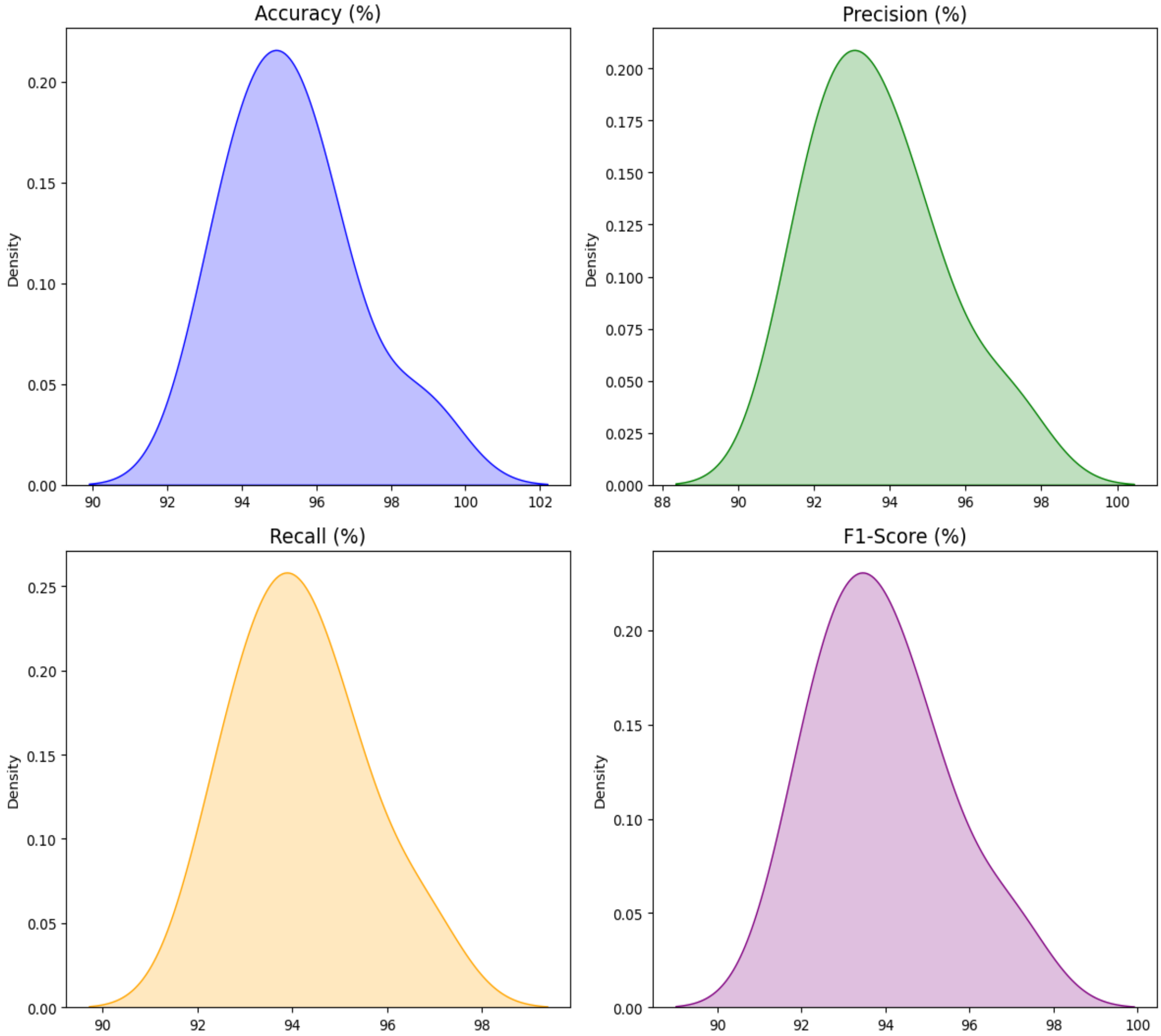

Fig 9. Optimization Comparison.

Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) claims substantial improvement from the traditional approach and the former has an accuracy of 96.67% while the later an accuracy of 96.12%. It contributes to the improvement of PSO's chance of searching the whole space by the help of swarm intelligence while GA's factors that are known as mutation and crossover also play an important role. They also have features of high accuracy, recall, F1-score that makes them favorably suitable for fine-tuning deep learning networks. Hybrid clearly prevails over all other techniques as it has delivered the highest level of accuracy which is 98.88%, not leaving far behind the measures of precision which has been recorded to be around 97.11%, the recall which has been determined to be at 96.75% and F1-score which has been observed to be at about 96.93%. The proposed hybrid approach incorporates the advantage of faster convergence of PSO in identifying best solutions and the second importance of preserving diversity by GA that results in the better choice of hyperparameter and consequently higher model efficiency. That is why this hybrid approach is more efficient in such a situation when searching for the global optima is vital. Other algorithms such as Bayesian Optimization and Hyperband are also noticeable but they do not come anywhere closer to the performance of PSO-GA. In general, it is clear that the PSO-GA hybrid all round offers the best performance of all corresponded parameters especially in the difficult category of underwater object recognition.

Table 7. Optimizer Comparison Based on Performance Metrics

Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Adam	95.45	93.67	94.12	93.89
SGD	94.12	92.34	93.45	92.89
RMSprop	93.78	92.01	92.78	92.39
AdaGrad	93.22	91.67	92.34	91.99
PSO	96.67	95.12	95.34	95.23
GA	96.12	94.67	95.01	94.84
PSO-GA (Hybrid)	98.88	97.81	97.75	97.83
Bayesian Optimization	95.12	93.45	94.23	93.83
Hyperband	94.89	93.12	93.67	93.39

Table 7 and **Fig 10** shows the comparison of optimizers to students through the aspects of accuracy, precision, recall, and F1-score to understand the variations of optimization methods. A reasonable performance is seen from the traditional optimizers Adam, SGD, RMSprop with the highest performance being provided by Adam optimizer with an accuracy of 95.45% and precision of 93.67%. It converges fast since during training, the learning rate can be adjusted as seen in Adam such making it popular in deep learning. But even then, it doesn't achieve the level of performance that is witnessed in other more complex optimization algorithms. Determining from the result of this study, RMSprop and SGD while faster and less complex in training, yields lower accuracy and slightly lower precision and recall indicating the fact that when it comes to tasks as complicated as Underwater Object Recognition more sophisticated optimization algorithms are needed.

PSO and GA outperform the traditional optimizers by presenting higher accuracy of 96.67% for PSO and 96.12% for GA. Since PSO is based on swarm intelligence to search the space this optimizer has an advantage when it comes to fine-tuning deep learning models with higher accuracy as shown by the use of crossover and mutation by GA. The two also have high precision and recall values, and as such, both optimizers can be applied to problems that call for convergence and switch in model. Indeed, better results achieved by PSO and GA show that these algorithms are well suited for hyperparameters and other parameters tuning of the ANN structure in more complicated search spaces and in a various data input environment like underwater imagery.

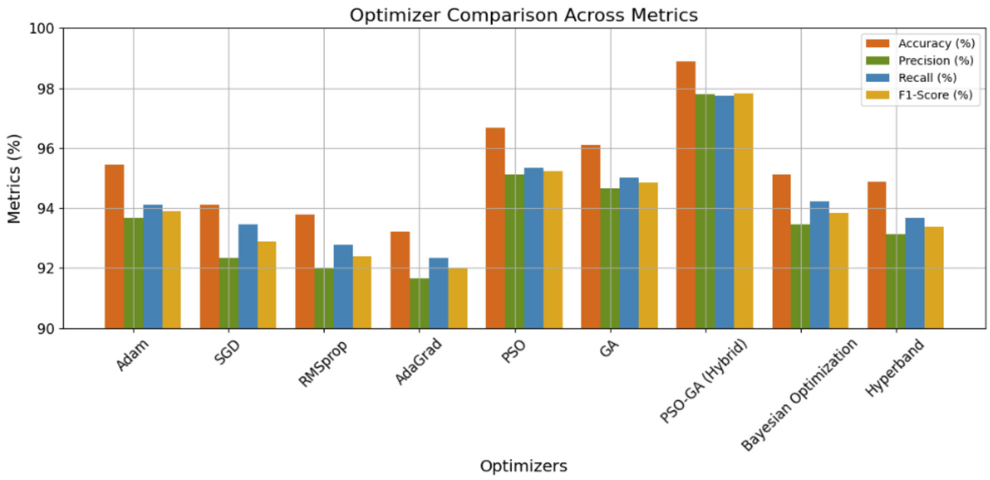

Fig 10. Optimizer Comparison across Metrics.

Comparing all these algorithms, the chosen PSO-GA hybrid optimization algorithm has the overall highest accuracy of 98.88% and the good precision, recall, and F1-score values. In combining the best feature of PSO rapid convergence, and GA to maintain diversity this hybrid improves model in making a best performance. The enhancement in the performance parameters in the PSO-GA hybrid method shows that the proposed technique is optimal in terms of exploring the search space and exploiting the search space for fine tuning. This makes it the most optimal optimization technique for the underwater object detection where accuracy as well as generalization should be optimized. Although Bayesian Optimization and Hyperband are better than many of the traditional optimizers, they failed to perform as well as the hybrid approach.

In evaluation, the model outperformed other leading architectures, including EfficientNet-B0, ResNet-50, and VGG-16, with substantial gains across metrics like precision, recall, and F1-score, reinforcing its robustness in underwater settings. This high-performance model has practical value in real-world applications, such as marine biology research, underwater infrastructure inspection, and autonomous underwater vehicle (AUV) navigation, where accurate, real-time object recognition is critical. These results demonstrate the proposed model's superior adaptability and precision, making it a valuable asset for advancing underwater exploration and monitoring.

V. CONCLUSION AND FUTURE WORK

The proposed methodology of combining transfer learning with hybrid optimization techniques (PSOGA) has proven to be highly effective for underwater object recognition. By leveraging EfficientNet as the feature extraction backbone and fine-tuning the model using PSOGA, the model outperforms several existing deep learning models and optimization techniques. The PSOGA model achieved significant improvements in accuracy, precision, recall, and F1-score compared to models using traditional optimizers like Adam, RMSprop, and AdaGrad. This demonstrates the hybrid approach's ability to balance exploration and exploitation during hyperparameter tuning, leading to superior performance. In terms of practical application, the PSOGA-based model shows great potential for use in underwater exploration, marine biology, and environmental monitoring. Its robust performance in challenging underwater environments suggests that this methodology can be adapted to other image recognition tasks where the quality of data is affected by noise and environmental factors. Additionally, its ability to generalize well across different datasets highlights the flexibility of hybrid optimization techniques. Looking ahead, there are several areas for future work. The model could be extended to include real-time object detection capabilities, which would be particularly beneficial for autonomous underwater vehicles (AUVs) and remotely operated vehicles (ROVs). Furthermore, experimenting with more advanced hybrid optimization methods, such as combining PSOGA with reinforcement learning techniques, could further enhance the model’s ability to adapt to new underwater conditions. Future work could also focus on expanding the dataset to include more diverse underwater scenarios, improving the model's robustness across a wider range of environments. The integration of additional sensor data, such as sonar or LIDAR, could also be explored to improve detection capabilities in complex underwater terrains.

CRediT Author Statement

The authors confirm contribution to the paper as follows:
Conceptualization: Sujilatha Tada and Jeevanantham Vellaichamy; **Methodology:** Sujilatha Tada and Jeevanantham Vellaichamy; **Writing- Original Draft Preparation:** Sujilatha Tada; **Visualization:** Sujilatha Tada and Jeevanantham Vellaichamy; **Investigation:** Sujilatha Tada and Jeevanantham Vellaichamy; **Supervision:** Jeevanantham Vellaichamy; **Validation:** Sujilatha Tada and Jeevanantham Vellaichamy; All authors reviewed the results and approved the final version of the manuscript.

Data Availability

The Datasets used and /or analysed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interests

The authors declare no conflicts of interest(s).

Funding

No fundings.

Competing Interests

There are no competing interests

References

[1]. Z. Liu, B. Wang, Y. Li, J. He, and Y. Li, “UnitModule: A lightweight joint image enhancement module for underwater object detection,” Pattern Recognition, vol. 151, p. 110435, Jul. 2024, doi: 10.1016/j.patcog.2024.110435.
[2]. S. R. Lyernisha, C. Seldev Christopher, and S. R. Fernisha, “Object recognition from enhanced underwater image using optimized deep-CNN,” International Journal of Wavelets, Multiresolution and Information Processing, vol. 21, no. 04, Mar. 2023, doi: 10.1142/s0219691323500078.
[3]. X. Hua., “Underwater object detection algorithm based on feature enhancement and progressive dynamic aggregation strategy,” Pattern Recognition, vol. 139, p. 109511, Jul. 2023, doi: 10.1016/j.patcog.2023.109511.
[4]. H. Zhou ., “Real-time underwater object detection technology for complex underwater environments based on deep learning,” Ecological Informatics, vol. 82, p. 102680, Sep. 2024, doi: 10.1016/j.ecoinf.2024.102680.
[5]. K. G, A. J, S. B, and M. P. M, “RETRACTED ARTICLE: A Deep Learning Approach to Detecting Objects in Underwater Images,” Cybernetics and Systems, pp. 1–16, Jan. 2023, doi: 10.1080/01969722.2023.2166246.
[6]. J. Zhang, J. Zhang, K. Zhou, Y. Zhang, H. Chen, and X. Yan, “An Improved YOLOv5-Based Underwater Object-Detection Framework,” Sensors, vol. 23, no. 7, p. 3693, Apr. 2023, doi: 10.3390/s23073693.
[7]. G. Chandrashekar, A. Raaza, V. Rajendran, and D. Ravikumar, “Side scan sonar image augmentation for sediment classification using deep learning based transfer learning approach,” Materials Today: Proceedings, vol. 80, pp. 3263–3273, 2023, doi: 10.1016/j.matpr.2021.07.222.
[8]. G. Verma, M. Kumar, and S. Raikwar, “F2UIE: feature transfer-based underwater image enhancement using multi-stackcnn,” Multimedia Tools and Applications, vol. 83, no. 17, pp. 50111–50132, Nov. 2023, doi: 10.1007/s11042-023-17180-1.
[9]. J. Zhou, T. Xu, W. Guo, W. Zhao, and L. Cai, “Underwater occluded object recognition with two-stage image reconstruction strategy,” Multimedia Tools and Applications, vol. 83, no. 4, pp. 11127–11146, Jun. 2023, doi: 10.1007/s11042-023-15658-6.
[10]. P. Pachaiyappan, G. Chidambaram, A. Jahid, and M. H. Alsharif, “Enhancing Underwater Object Detection and Classification Using Advanced Imaging Techniques: A Novel Approach with Diffusion Models,” Sustainability, vol. 16, no. 17, p. 7488, Aug. 2024, doi: 10.3390/su16177488.

[11]. X. Liu, Z. Chen, Z. Xu, Z. Zheng, F. Ma, and Y. Wang, "Enhancement of Underwater Images through Parallel Fusion of Transformer and CNN," Journal of Marine Science and Engineering, vol. 12, no. 9, p. 1467, Aug. 2024, doi: 10.3390/jmse12091467.
[12]. J. Gao, Y. Zhang, X. Geng, H. Tang, and U. A. Bhatti, "PE-Transformer: Path enhanced transformer for improving underwater object detection," Expert Systems with Applications, vol. 246, p. 123253, Jul. 2024, doi: 10.1016/j.eswa.2024.123253.
[13]. Y. Zhang, Q. Jiang, P. Liu, S. Gao, X. Pan, and C. Zhang, "Underwater Image Enhancement Using Deep Transfer Learning Based on a Color Restoration Model," IEEE Journal of Oceanic Engineering, vol. 48, no. 2, pp. 489–514, Apr. 2023, doi: 10.1109/joe.2022.3227393.
[14]. H. Yang, W. Peng, J. Yao, and X. Ye, "Effective adversarial transfer learning for underwater image enhancement with hybrid losses," Signal, Image and Video Processing, vol. 18, no. 10, pp. 6671–6681, Jun. 2024, doi: 10.1007/s11760-024-03343-6.
[15]. M. Zhou, L. Cai, J. Jia, and Y. Gao, "Multi-scale aware turbulence network for underwater object recognition," Frontiers in Marine Science, vol. 11, Mar. 2024, doi: 10.3389/fmars.2024.1301072.
[16]. H. Yao, T. Gao, Y. Wang, H. Wang, and X. Chen, "Mobile_ViT: Underwater Acoustic Target Recognition Method Based on Local–Global Feature Fusion," Journal of Marine Science and Engineering, vol. 12, no. 4, p. 589, Mar. 2024, doi: 10.3390/jmse12040589.
[17]. V. Malathi, A. Manikandan, and K. Krishnan, "Optimzied resnet model of convolutional neural network for under sea water object detection and classification," Multimedia Tools and Applications, vol. 82, no. 24, pp. 37551–37571, Mar. 2023, doi: 10.1007/s11042-023-15041-5.
[18]. J. Chen, et al., "Underwater Object Recognition Using Enhanced VGG Network", Electronics, vol. 12, no. 6, pp. 751-763, (2023), DOI: 10.3390/electronics12060751.
[19]. Y. Zhao, et al., "EfficientNet-Based Real-Time Underwater Object Detection for Marine Applications," Access, vol. 11, pp. 1401–1412, (2023), DOI: 10.1109/ACCESS.2023.3224017.
[20]. P. Liu, et al., "DenseNet-Based Underwater Object Recognition Using Transfer Learning", JoMSE, vol. 11, no. 2, pp. 235-245, (2023), DOI: 10.3390/jmse11020235.
[21]. F. Zhang, et al., "InceptionV3-Based Deep Learning Framework for Underwater Object Recognition", Sensors, vol. 23, no. 4, pp. 1289-1299, (2023), DOI: [10.3390/s23041289.
[22]. L. He, et al., "Enhanced Xception Model for Underwater Object Detection", Applied Sciences, vol. 14, no. 1, pp. 89-98, (2024), DOI: 10.3390/app14010089.
[23]. G. Sun, et al., "MobileNetV2-Based Lightweight Underwater Object Recognition System", TGRS, vol. 61, pp. 123-133, (2023), DOI: 10.1109/TGRS.2023.3244782.
[24]. Y. Li, et al., "ShuffleNet-Based Approach for Real-Time Underwater Object Detection", JOE, vol. 59, no. 4, pp. 157-168, (2023), DOI: 10.1016/j.oceaneng.2023.102189.
[25]. L. Zhang, et al., "A Deep Learning Approach to Detecting Objects in Underwater Images using AlexNet", IJOMR, vol. 18, pp. 102-113, (2023), DOI: 10.1109/JOMAR.2023.119345.