



## Edge Cloud IoT Model Based Marine Life Analysis Using Machine Learning Algorithms

Gagan Kumar Koduru<sup>1,\*</sup>, S. Kalaimagal<sup>2</sup>, M. Srilakshmi Preethi<sup>3</sup>, G. L. Narasamba Vanguri<sup>4</sup>,  
Shivanadhuni Spandana<sup>5</sup>, M. Syed Rabiya<sup>6</sup>, M. Rajesh<sup>7</sup>

<sup>1</sup>Associate Professor, Department of Computer Science and Engineering, MLR Institute of Technology, Hyderabad, India

<sup>2</sup>Professor, Department of AI & DS, Panimalar Engineering College, Chennai, Tamil Nadu, India

<sup>3</sup>Assistant Professor, CSE-Cyber Security, Madanapalle Institute of Technology & Science, Madanapalle, Andhra Pradesh, India

<sup>4</sup>Assistant Professor, Department of Information Technology, Aditya University, Surampalem, Andhra Pradesh, India

<sup>5</sup>Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad-500043, Telangana, India

<sup>6</sup>Assistant Professor, Department of CSE, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India

<sup>7</sup>Department of Computer Science and Engineering, Aarupadai Veedu Institute of Technology, Vinayaka Mission's Research Foundation (DU), Tamilnadu, India

Emails: [gagan.koduru@gmail.com](mailto:gagan.koduru@gmail.com); [drsivamunikalaimagal@gmail.com](mailto:drsivamunikalaimagal@gmail.com); [preethinaveen22@gmail.com](mailto:preethinaveen22@gmail.com); [gayatrijeedigunta05@gmail.com](mailto:gayatrijeedigunta05@gmail.com); [s.spandana@klh.edu.in](mailto:s.spandana@klh.edu.in); [drsyedrabiya@veltech.edu.in](mailto:drsyedrabiya@veltech.edu.in); [rajesmano@gmail.com](mailto:rajesmano@gmail.com)

### Abstract

The amount of marine data is such that it is pointless, and at times infeasible, to attempt training deep learning models on personal workstations. In this work, we present the advantages of cloud based distributed learning in training of deep learning (DL) model and management of big data. Moreover, large volumes of marine big data are classically through wire networks, which are costly, if at all deployable, to maintain. This research propose novel technique in marine life analysis based on remote sensing image using edge cloud IoT model and machine learning algorithms. Here the edge cloud IoT model has been used for collecting remote sensing image in marine life analysis. This remote sensing image has been processed for noise removal as well as normalization. Then this image is feature extracted as well as classified utilizing principal Gaussian convolutional fuzzy encoder with Bayesian reinforcement Markova algorithm. Experimental analysis has been carried out in terms of classification accuracy, average precision, recall, F1 score, AUC for various marine life dataset. proposed technique obtained 97% Classification accuracy, 95% Average precision, 93% Recall, 88% AUC, 94% F1 SCORE.

Received: March 30, 2025 Revised: June 07, 2025 Accepted: July 17, 2025

**Keywords:** Marine life analysis; Remote sensing image; Edge cloud IoT; Machine learning algorithms; Fuzzy encoder

### 1. Introduction

High image content databases provide an essential means for the ecological examination of deep-sea organisms since they aid in the detailed, prolonged investigation of underwater species. Nevertheless, it may also take an unreasonable

amount of effort in analyzing a collection of images to yield relevant data such as the size or existence of species. This is exacerbated by the fact that contrary to rigid upright and stationary aquatic cameras, complex images obtained from mobile cameras mounted on rovers or drones (in this case underwater) does not offer clear quality as lighting and focus is arbitrary. For this reason, such approaches have gained popularity, often in the contexts of computer and marine biology, targeting specific image datasets [1]. Developing and deploying machine learning models can be quite a challenge, which might discourage the lay users, as they may not be proficient in scientific programming languages. Because of this model-sharing barrier, some image analysis work is becoming obsolete. To some extent, these concerns can be addressed through ready-made, user-friendly, and generalized machine learning technologies. Thanks to this, it is now possible for non-coders, who do not know how to work with machine learning deeply, to create models - by teaching the so-called “base” neural network. The Earth’s surface has vast turbulent oceans that cover majority of the surface, yet many volumes and the predicted ocean floor remain a mystery. This is primarily due to the distinct characteristics of the ocean setting, which includes factors such as, extreme pressure, frigid temperatures, and absence of light that render it oppressive to humans as well as form concrete barriers to exploration [2]. However, thanks to advancements in robotic platforms and sensor technology, it has now become possible to penetrate virtually any zone of deep marine biosphere and correlate obtained information on numerous ecosystems and the organisms inhabiting them. The input and output of many resources contained in oceans can be exhausted due to over-exploitation of some of these resources. Hence in order to aid in the implementation of effective and scientifically based management regimes, more efforts are required to be oriented towards the baseline understanding of marine ecosystem integrity in terms of species and habitat composition. This data acquisition is up to date with framework of UNESCO Ocean Decade Initiative [3]. Progress in High Definition (HD) optics, low light technology aided by acoustic multi beam cameras that makes vision in water even in darkness, have significantly increased the underwater photography technology in the 20 years. In addition, together with cable system observatories and their mobile delivery vehicles (crawlers), advanced sensor suites are being integrated that integrate cameras and oceanographic as well as geochemistry resources [4]. As a practical approach for evaluating their ecological niche, this new method that allows the simultaneous acquisition of images and multiparametric environmental data has begun to enable visual estimates of various animal species inhabiting a marine community to be correlated with changes in the quality of the respective habitat. This is accompanied by advancement in technology in the use of robotic vessels, like AUVs and ROVs. The training process of the models however may vary with the apparatus used; the interfaces are commonly designed to hide this intricacy. In discipline of AI, DL is a subdivision of ML that combines advanced techniques in data categorization using the activity of the human brain and makes prediction possible through its own training. This technology has been of great interest in the computing sector because of its impressive learning skills, and it has been applied in many areas text processing, image recognition, medicine, and protection of information, for example. In addition, the availability of accessibility by open source sharing facilitates optimizing and improving tools in no time, which overall enhances the speed of growth. Recognition technology is a typical part of AI systems that can be described as those that identify to an object by means of a photograph [5]. The total size of the global artificial intelligence market is expected to increase to \$36.8 billion by the year 2025 with image recognition and image tagging segment forecasted to be around \$8.1 billion. The growing intersection of AI-oriented technology is giving a lot of advantage in the progression of marine biotechnology with the bioeconomy, which seeks to solve the issues of the future, emerged. Accordingly, Jeju Island announced in June 2023 the "Jeju Bio-Industry Development Strategy" in order to enhance the global competitiveness of the marine bio-industry and to facilitate its growth into generating more value-added services. Jeju Island, which lies for about three fourths of the coastlines of South Korea and is blessed with plentiful diverse marine biological raw materials, is considerable to the bioindustry. However, the growth of the bio-industry lags behind that of other regions because of a lack of technology and skilled labor [6].

Main contribution for this research is as follows,

To suggest a new method for analysing marine life from remote sensing images by utilising machine learning techniques and edge cloud IoT models. Here, remote sensing images for the investigation of marine life have been collected using the edge cloud IoT architecture. This remote sensing image has undergone normalization and noise reduction processing. The principal Gaussian convolutional fuzzy encoder with the Bayesian reinforcement Markov method was then used to extract features and classify this image.

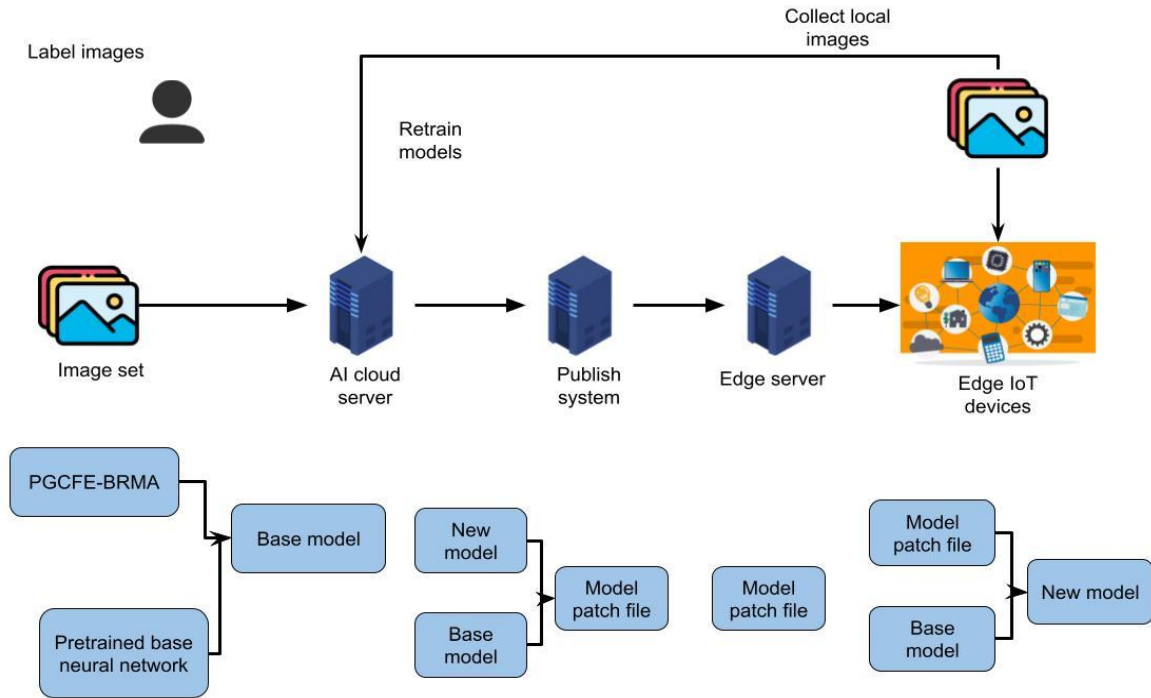
## **2. Related works**

The environment that is below water level is not a good one for photography. The scattering, absorption and backscattering of the light rays directed to the camera are caused by the particles dissolved in seawater, resulting in poorly contrasted, blurred and pale pictures. In addition, different light spectra are absorbed in different powers by seawater due to the frequency dependent power absorption characteristic of the seawater. Other than these and many

imaging challenges, the underwater medium presents spatially varying imaging properties [7]. The above-discussed nonuniform image degradation processes make underwater picture segmentation challenging. This problem is often addressed with the help of statistical models by some researchers. For example, in reference [8], the authors have reported recognition of objects in underwater images using optimally weighted Gaussian mixture modelling and genetic algorithm. It is now commonly seen that deep learning algorithms far advanced existing techniques in realization of underwater images but the classical inner distance form matching techniques have been found to work effectively in only a few scenarios. In the work by [9], they argue that the dust particles from tumbling tires are an important source of pollution and emphasize the fact that vehicle tire wear contributes to a large number of microplastics and fine dust emissions. The study observed that the pollution of microplastics is an emerging crisis that is linked to the services meant to control air pollution, justifying a call for more sophisticated monitoring equipment. In a related context, the author examines in more detail the methods allowing increasing the efficiency of such large-scale environmental systems as, for example, watersheds. This research contributes to the understanding of advanced management practices in mitigating environmental degradation, along with its sustainability against pollution over time, which is directly related to the use of edge computing in the monitoring and controlling of emissions into the environment. Some of the marine Internet of Things technologies that are focused on linking maritime robotic agents are also presented in Works [11]. Their study investigates use of IoT sensors in real time as well as looks at the IoT prospects in the maritime setting. It is of utmost importance in the case of marine IoT systems, among others as well, where edge computing can come in handy since data can be processed in real-time from the devices, increasing the efficiency of the systems. As an illustration of an IoT smart water quality monitoring system, the author [12] explains how it is possible to measure water quality efficiently by the use of IoT technologies. This study, which focuses on such an interesting application of IoT, tends to be in line with the works that we are doing on monitoring the environment in real-time using edge computing. The compute continuum solution we are explaining can be classified under mobile Cloud computing as classified under [13]. The edge device reduces the bandwidth demand and emphasis on the detection of parts of photographs underwater for monitoring purposes (e.g., to send out alarms earlier) by sending to the cloud only the relevant images for further processes. Reference [14] proposes a four-stage classifier that consists of two convolutional layers, followed by a feature-size shrinking pooling layer, then involves Spatial Pyramid Pooling SPP structures which get rid of the effect of large angles in information. The last phase of the framework is classified using SVM with over 90% classification performance. [15] created a framework for fish detection using Fast R-CNN as CNNs yielded high-quality results on general-purpose photos in image datasets like ImageNet and VOC. The mean Average Precision (mAP) of 30,000 photos taken from Fish4Knowledge, taking into account 12 species, was used to evaluate the experimental findings. [16] offers an additional option that makes use of regions such as CNN. The three-stage pipeline they propose uses an underwater image, its Gaussian Mixture Model, its Optical Flow (OF) as inputs. At time of publishing, the authors claimed that their accuracy of 87.44% was among best on LCF2015.

### **3. Edge cloud IoT model in remote sensing image analysis**

To enhance the implementation of NN methods with transfer learning practices, Cloud-Edge Smart IoT Infrastructure, method-publishing system is presented in Figure 1. The patterned cloud AI server downloads the weights and layers of previously trained NN method, and adjusts local weights of recently trained NN method using the technique known as transfer learning. A means to help with the communication of neural network models is afforded by a mechanism to publish models. After training, cloud AI server makes use of proposed method publish system as well as encloses feature weight updates into a NN method 'patch' file. The received method patch file is then distributed to edge servers. After obtaining a new method patch file and caching it, edge server notifies its surrounding smart edge IoT devices of need for NN method upgrade. IoT devices transform into model update requests after the nearby smart edge devices have sent notifications, and request to download updated model patch file, to change their local ANN methods, if they exist. In this method, there is no need for the cloud server to send large-volume NN methods files to the smart edge IoT devices because it is only required to send small-sized NN method patch files with help of edge servers. The weights can be updated in conjunction with the edge IoT device's current models once the device receives the patch file. Neural network model deployment speed can be increased and network bandwidth transmission needs are significantly decreased with aid of suggested collaborative edge computing platform.



**Figure 1.** Proposed edge cloud IoT model in remote sensing image analysis

The cloud AI servers in the suggested system architecture use same basic NN. With aid of the suggested model publish system, the preprocessed and trained feature weights are then processed through a difference comparison (diff) and stored in a NN method patch file following the training of new NN model. Using edge servers, method publish system is able to update each edge IoT device by sending NN method patch file for current neural network method in accordance with its weights. To ensure that an image can be traced to a specific source, each node has a jump table that has two parts, one for keys and the other for the home addresses for their corresponding values. The key is given as the hashcode of the features of an RS image, which are typically its coordinates, capture time, satellite type and so forth. Once a node issues a request, fastChord will make use of the well-known Chord protocol and forward the key to the node at which value is supposed to be present. The desired outcome is returned if it is present. If not, the image's home address is found by searching the node's jump table. After processing, the image is returned to the node that made the request. The jump table allows RSpCloud to process RS pictures without requiring the migration of many data, in addition to providing the computing function.

In contrast to sending user data directly to the main cloud, there may be a comparatively lower latency while transferring the user data to the edge. At the same time, the processing resources of the edge cloud are very limited, equal to a small data centre, meaning that the edge cloud in question consists only of around 20-50 servers. Every server possesses physical resources, like CPU, RAM, and ingress bandwidth that are finite. Because each server is equipped with its own hard drive, consider that there exists no centralised storage in edge cloud. Once the edge fails to satisfy QoS need for competitive process, the task is sent to the core, which contains sufficient computing power. The core is described as a federation of data centres that are oftentimes hundreds of miles away from the consumers. Although latency is greater in comparison to edge servers, core servers perform just as well if not better than edge solutions. In terms of energy expended, the time taken for processing in the core as compared to that at the edge is far less. Still, a lot of information has to enter the core for processing, which means that the transmission of data from the user to the core through Internet cannot be disregarded. A job is a request that a vehicle submits and that needs to be processed using computational resources. It is positioned to edge and core at any time. After the request is approved, a Virtual Machine (VM) situated on edge or core where analysis is needed completes request. The fundamental unit of resource allocation is a virtual machine (VM). Every virtual machine is built with unique CPU and RAM specifications. The virtual machine is destroyed and its reserved resources are returned to server when car leaves this section of road.

#### 4. Principal Gaussian convolutional fuzzy encoder with Bayesian reinforcement markov algorithm (PGCFE-BRMA)

The first principal component has the largest sample variance conceivable, meaning it maximally discriminates among the sample individuals. Because PCA was the default process in many software applications, it was once the most widely utilized technique for estimating pattern coefficients. The PCA equation (1) is:

$$pc_l = w_{l1}X_1 + w_{l2}X_2 + \dots \dots \dots w_{lp}X_p \quad (1)$$

where X are the variables that permit the individuals' PC1 score to vary as much as feasible, and w are the loadings. Due to their differences, it is crucial to go over the factor analysis equation and contrast it with the principle components equation. The variables are a linear combination in PC, and the factors are a linear combination in FA. Additionally, PCA does not distinguish error, whereas EFA does. Despite their mathematical differences, they are too frequently treated the same way due to their comparable appearance. It is interesting to note that component analysis is rarely applied completely. The following equation provides a mathematical model, where y is output value, I1, I2, ..., by eqn (2)

$$y = (\theta) \cdot \sum_{i=1}^n (W_i - \mu) \quad (2)$$

Contribution of Ith covariate to log-odds is generalized weight W, which is defined by following equation (3):

$$W = \frac{\partial \log\left(\frac{o(x)}{1-o(x)}\right)}{\partial x} \quad (3)$$

Network was trained until error measurement became close to (10<sup>-4</sup>), or for at least 1000 iterations. The training dataset was subjected to cross-validation procedure with LOOV technique in order to find number of neurons in hidden layer for this model.

The input images are pre-processed using a Gaussian filter. Sixteen layers are then included, starting with the input layer's pre-processed augmented images and continuing with feature selection, downsampling, and convolution operations. Overfitting is prevented by employing the dropout layer. Softmax and fully connected layers are later used for output prediction; classification layer is added for predicted-class classification. The illustration of each layer is as follows: the input layer is used for input size validation and data normalization. A 2D convolution is applied by moving the filters on the input BT images and by evaluating dot product of the inputs and weights. The input images undergo a stride, which is the sliding of filters or kernels, by adhering to the horizontal and vertical steps. The actual picture is padded before the kernel is loaded. These filters are used as feature-identifiers. While the advanced layers are used for complicated feature identification, the first layers filter the low-level highlights.

Acquisition functions, including Gaussian process hyperparameters, rely on prior observations; these dependencies are represented as a(x; {x<sub>n</sub>, y<sub>n</sub>}; P). There are numerous widely used acquisition functions, however they only use predictive mean function of a particular method, μ(x; {x<sub>n</sub>, y<sub>n</sub>}, P), in combination with predictive variance function, σ<sup>2</sup>(x; {x<sub>n</sub>, y<sub>n</sub>}; P), when using Gaussian process prior. Thus, {X<sub>b</sub> = argmax<sub>x<sub>n</sub></sub> f(x<sub>n</sub>), β(.) as the standard normal's cumulative distribution function, and γ(.) as its standard normal density function are the current best values. Maximizing the likelihood of improving the existing best result is a noteworthy intuitive strategy that is referred to as probability of improvement (PI). This is calculated analytically as follows (4):

$$a_{PI}(\cdot; \{x_n, y_n\}, P = \beta(\varphi(x)),$$

$$\varphi(x) = \frac{f(x_b) - \mu(x; \{x_n, y_n\}, P)}{\sigma(x; \{x_n, y_n\}; P)} \quad (4)$$

An alternative approach, which is closely associated with Gaussian process, was to increase anticipated improvement (EI) of present best result in this work by eqn (5):

$$a_{EI}(\cdot; \{x_n, y_n\}, P = \sigma(x; \{x_n, y_n\}; P)(\varphi(x)\gamma(\varphi(x)) + N(\varphi(x); 0,1)). \quad (5)$$

In the first layer, there is a max-pooling of (2 x 2) operations, 16 number of filters, 1-stride, and 9 kernel size. Second contains parameters that are similar to first, except a dropout layer with 0.15. Third layer has a max pooling of (2x2), 36 filters, 1-stride, and 9 kernel sizes, and a dropout layer of 0.15. Finally, we perform the neural architectural Search operations in the dynamic layer, which is the fourth and final layer. Adam was dyr commercialized while categorical cross entropy served as the loss function. A model optimal in every regard with the exception of its parameters was finalized for the training, prior to training, 30 epochs of training were scheduled with a batch size of 128, and a model with the parameters set at zero was prepared before the training. Eleven calls were made, and expected improvement was acquisition function used.

Fuzzy clustering contains more data than clear clustering since membership degree characterizes uncertainty of data points that belong to cluster centers.  $X$  is a set of  $N$  samples, which corresponds to cluster  $c$  in Eq. (6), where  $d$  is the L2-norm.

$$d = \sqrt{\sum_{i=1}^N x_i^2} = \sqrt{x_1^2 + x_2^2 + \dots + x_N^2} = \sqrt{\sum_{i=1}^N x_i * x_i^T} \quad (6)$$

It is possible to determine the likelihood that each data point  $x_i$  is a member of cluster  $c$  by maximizing the objective function. That is, the membership degree  $u_{ic}$  of each  $x_i$  can be found by measuring the distance between each data point  $x_i$  and cluster center  $v_c$  by eqn (7)

$$\min_{u,v} \sum_{i=1}^N \sum_{c=1}^C u_{ic}^m \|x_i - v_c\|_2^2 \quad (7)$$

where  $\sum_{i=1}^N \sum_{c=1}^C u_{ic} = 1, u_{ic} \geq 0, 0 \leq i \leq N$ , and  $0 \leq c \leq C$ .  $C$  is pointed cluster number,  $v_c$  is  $c$ th cluster center,  $m$  is membership degree. Time-frequency localization and focused features are characteristics of the Wavelet transform. In order to address practical issues, it is necessary to combine wavelet transform with a regular auto-encoder. The "deep wavelet auto-encoder" model, a novel type of enhanced unsupervised neural network that can capture non-stationary vibration signals and represent complicated information, was proposed in this study. Wavelet auto-encoder method defined several resolutions by using wavelet function as activation function in a conventional state.

Deep auto-encoder's operation, and Equation (8) shows the stages of decoding.

$$X = \zeta(\hat{K}'Y + b') \quad (8)$$

where  $k$  is the kernel vector,  $b'$  is the bias value,  $X$  is the result of the reconstructed vector, and  $\epsilon$  is an error value introduced during backpropagation. Think of  $y$  as a dependent variable. Let a set of vectors  $X = \{x_1, \dots, x_K\}$  represent the ensemble of various  $K$  predictions, and let  $y^T$  be the training data of length  $T$ . In this study,  $K$  stands for the eight models, and  $y$  for ET ensembles. The BMA probabilistic forecast of  $y$ 's probability density function (PDF) can be obtained as eqn (9)

$$p(y | x_1, \dots, x_K) = \sum_{k=1}^K p(y | x_k) p(x_k | y^T) \quad (9)$$

where  $p(x_k | y^T)$  is the posterior probability of model predictions  $x_k$ , or the chance that the predictions of  $x_k$  will be accurate given training data  $y^T$ , and  $p(y | x_k)$  is the PDF of  $y$  obtained from  $x_k$  independently. Each regressor was given an equal prior by the uniform model, with a probability of  $p(y | x_k) = 1/K$ . Using the formula  $p(y | x_k) = \theta^i (1 - \theta)^{K-i}$ , the binomial model calculates a fixed prior to each regressor, which is the product of either inclusion and exclusion probability.

An average reward for an infinite horizon  $(S, A, r, p)$  can be used to describe the Markov Decision Process (MDP). Assumed to be finite sets,  $S$  and  $A$  have cardinalities of  $|S|$  and  $|A|$ , respectively. Starting from state  $s$ , average reward for each stage of a deterministic/stationary policy  $\pi: S \rightarrow A$  is described as eqn (10)

$$J^\pi(s) := \liminf_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{t=1}^T r(s_t, \pi(s_t)) \mid s_1 = s \right] \quad (10)$$

where  $p(\cdot | s_t, \pi(s_t))$  yields  $s_{t+1}$ . Assume that  $\max_{\pi \in \Pi} J^\pi(s) = J^*(s)$ . If a policy  $\pi^*$  satisfies  $J^{\pi^*}(s) = J^*(s)$  for all  $s \in S$ , it is considered optimal. distinct  $J^* \in [0, 1]$  such that the Bellman equation below holds and  $J^*(s) = J^*$  for all  $s \in S$  by eqn (11)

$$J^* + q^*(s, a) = r(s, a) + \mathbb{E}_{s' \sim p(\cdot | s, a)} [v^*(s')] \quad (11)$$

where  $\max_{a \in A} q^*(s, a) = v^*(s)$ . Following that,  $\pi^*(s) = \arg \max_a q^*(s, a)$  yields the optimal policy. We examine a learning issue in which the agent knows  $S, A$ , and the reward function  $r$ , but not the transition probability  $p$  (i.e., the Bellman equation cannot be solved directly). A random state  $s_1 \in S$  is where an agent begins. The agent observes state  $s_t \in S$  at every time step  $t = 1, 2, 3, \dots$ . Action at  $\in A$  is determined by history  $s_1, a_1, s_2, a_2, \dots, s_{t-1}, a_{t-1}, s_t$ . Next, environment draws  $s_{t+1}$  in accordance with  $p(\cdot | s_t, a_t)$  to decide next state. The concept of cumulative regret, which is difference between algorithm's total reward and optimal policy's, is used to assess a learning algorithm's performance by eqn (12)

$$R_T := \sum_{t=1}^T (J^* - r(s_t, a_t)) \quad (12)$$

In the worst-case scenario, the regret can increase linearly with  $T$  since  $r \in [0, 1]$ .  $RT / T$  goes to zero when a learning algorithm reaches sub-linear regret, meaning that the method's average reward converges to the best reward for each stage,  $J^*$ . Method based algorithm yielded the best known regret bound,  $Oe(\sqrt{DSAT})$ , which meets the bottom bound.

### 5. Results and discussion

Simulation setup- With a system specification of an Intel(R) Core(TM) i5-1005G1 CPU and 16 GB of RAM, development environment is built on top of Windows 10. VScode notebook for Jupyter. Proposed method is built utilizing Python as well as Visual Studio. For data preprocessing, NLTK Python model is used. Text is vectorized utilizing Keras and Pandas. Data analysis and manipulation are done with NumPy and pydata.org.

Dataset description: Planktonic communities are largely represented by large, publicly accessible tagged image databases for marine themes. These datasets, which include WHOI-Plankton27 and EcoTaxa11 among others, necessitate regulated lighting and imaging conditions that make use of darkfield, brightfield, or holographic illumination26. Larger organisms and other marine animals present in midwater and benthic settings are necessarily excluded from these massive datasets, which are primarily designed for classification tasks, despite the fact that the plankton imaging community has made great strides in developing picture data portals. These plankton databases are comparable in size to CoralNet28, a portal and platform for dealing with coral pictures, but it is also limited to a certain kind of organism. In the CV and AI fields, image sets of higher trophic-level animals29 are usually dispersed in separate archives that might be challenging non-subject matter experts to locate; there are no comparable datasets for macroscopic ocean species. Therefore, a tagged image dataset that is representative of various biological communities in the ocean across spatial scales and is easily accessible in a single, publicly accessible online repository is obviously needed. Images and videos of marine life can be found in the SeaCLEF 2017 multimedia dataset. Identifying species or persons from pictures or videos is the aim in every situation.

About 150 marine animal species from throughout the globe will be included in SeaCLEF's tackled species.

In particular, SeaCLEF datasets contains:

- 1) A collection of movies taken on Taiwanese coral reefs for automated fish identification and monitoring;
- 2) Recordings made close to a water turbine for automated identification of infrequent salmon-related events;
- 3) A collection of poorly labeled photos of 148 sea animal species.
- 4) A collection of whale caudal picture pairs.

**Table 1:** Processing of input marine life image utilizing proposed technique



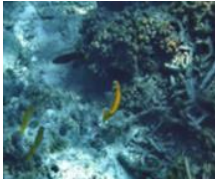
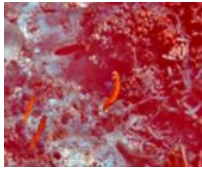








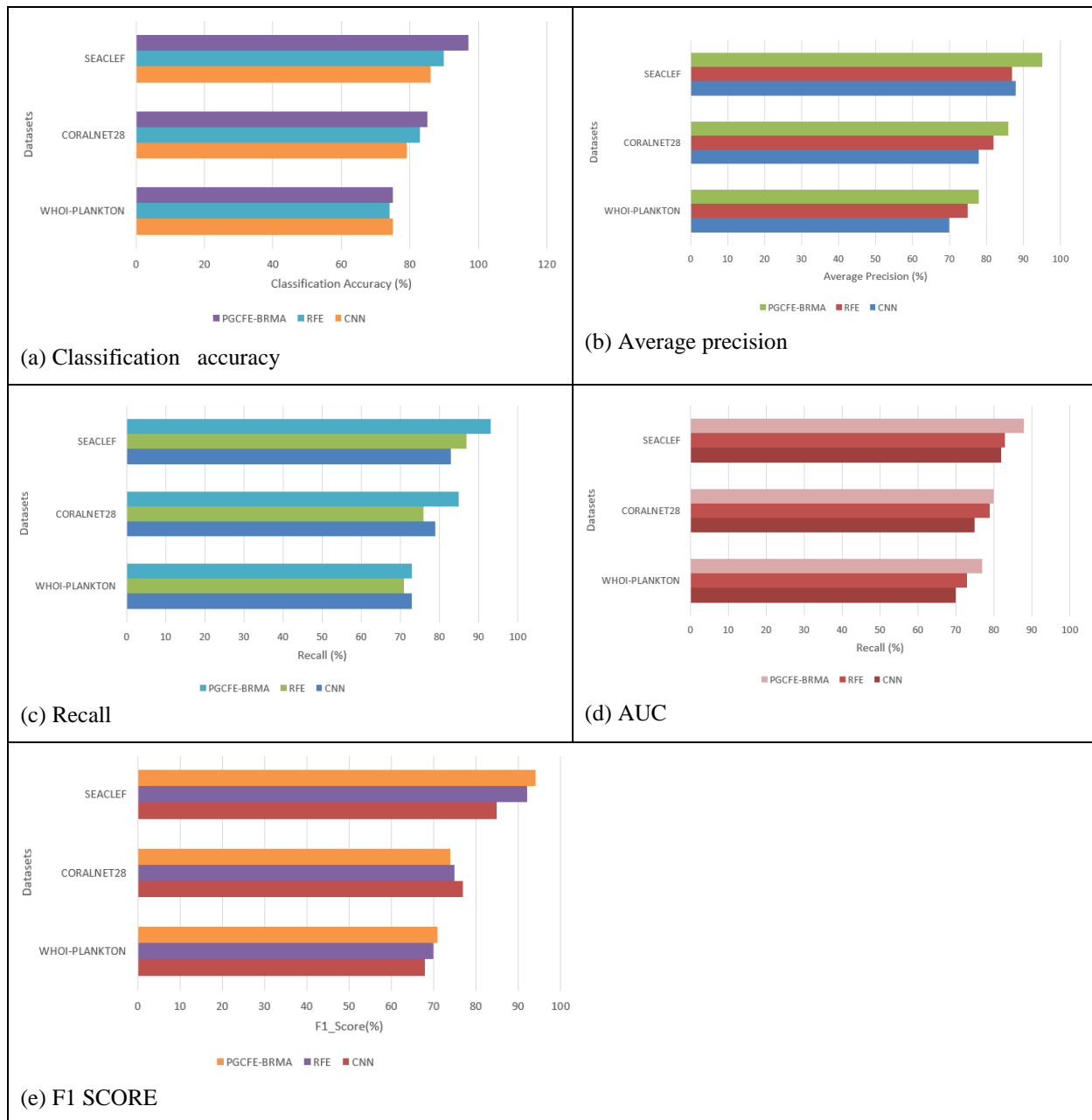
Input marine life image	Noise removed image	marine life image feature extraction	marine life image classification
			
			
			

Table 1 above illustrates how various input images from datasets are processed. As part of model-learning phase, preprocessing of image data was used to retrain the pretrained image classification model, which was originally intended to classify objects in general, to classify specifically marine-life images. In view of these properties, we implemented an additional dense layer in the upper parts of each model so that the lower layers could be adjusted in their weights. After the model was trained, 20 classification layers were incorporated by transfer learning and fine-tuning techniques. In other terms, the model that had gone through transfer learning and which had been trained on various marine images was optimized for the marine images only in order to improve data efficiency and learning loss. Each model was trained ten times, and the accuracies were evaluated to pick the best model for marine life recognition. The model that achieved the best results was selected for further training. We calculated categorical cross-entropy and accuracy, monitoring the learning process by comparing actual values to predicted ones, and subsequently to validation data. After the apply step, the trained model was scored on validation data in order to assess its generalization potential and the degree of overfitting and accuracy along with cross entropy was used in the curing process. When we completed the model re-training phase, we tested the model performance and checked accuracy of method with training and testing sets. For re-training of the picture classification model, 81% of the dataset was used as training data. Nona of a sample was held for the validation of method in order to test generalizability of the method designed. Finally, the test data utilized to calculate performance of image classification model consisted of 10% images for each marine species through hierarchical sampling. At this time, data augmentation was not applied to the test data; therefore, testing was performed with 93 original images, corresponding to 10% of the total dataset.

**Table 2:** Comparative for marine life image dataset

<b>Dataset</b>	<b>Techniques</b>	<b>Classification accuracy</b>	<b>Average precision</b>	<b>F1 score</b>	<b>Recall</b>	<b>AUC</b>
<b>WHOI-PLANKTON</b>	<b>CNN</b>	75	70	68	73	70
	<b>RFE</b>	79	78	77	79	75
	<b>PGCFE-BRMA</b>	86	88	85	83	82
<b>CORALNET28</b>	<b>CNN</b>	74	75	70	71	73
	<b>RFE</b>	83	82	75	76	79
	<b>PGCFE-BRMA</b>	90	87	92	87	83
<b>SEACLEF</b>	<b>CNN</b>	75	78	71	73	77
	<b>RFE</b>	85	86	74	85	80
	<b>PGCFE-BRMA</b>	97	95	94	93	88

Table-2 shows comparative for various marine life image dataset. Proposed technique analysed for WHOI-PLANKTON, CORALNET28, SEACLEF multi-modal watermarked image dataset. Parametric analysis has been carried out in terms of Classification accuracy, Average precision, recall, AUC, F1 SCORE.



**Figure 2.** Comparative for marine life image dataset in terms of (a) Classification accuracy, (b) Average precision, (c) Recall, (d) AUC, (e) F1 SCORE

Figure 2 (a)- (f) shows comparative for marine life image dataset. Here proposed technique Classification accuracy 86%, Average precision 88%, Recall 83%, AUC 82%, F1 SCORE 85%. While existing CNN attained Classification accuracy 75%, Average precision 70%, Recall of 73%, AUC of 70%, F1 SCORE of 68%; RFE attained Classification accuracy of 79%, Average precision 78%, Recall 79%, AUC 75%, F1 SCORE 77% for WHOI-PLANKTON dataset. Proposed technique obtained 90% Classification accuracy, 87% Average precision, 87% Recall, 83% AUC, 92% F1 SCORE. existing CNN attained Classification accuracy 74%, Average precision 75%, Recall 71%, AUC 73%, F1 SCORE of 70%; RFE attained Classification accuracy of 83%, Average precision of 82%, Recall 76%, AUC 79%, F1 SCORE 75% for CORALNET28 dataset. For SEACLEF dataset proposed technique obtained 97% of Classification accuracy, 95% of Average precision, 93% Recall, 88% AUC, 94% F1 SCORE. existing CNN attained

Classification accuracy of 75%, Average precision 78%, Recall 73%, AUC 77%, F1 SCORE 71%; RFE attained Classification accuracy 85%, Average precision 86%, Recall 85%, AUC 80%, F1 SCORE 74%.

Under typical usage conditions, this model operates favorably from the temporal aspect. However, with respect to the growing data size, the performance needs to be evaluated since other frameworks have larger datasets. Evaluation of the use cases of different frameworks demonstrates that most of the devices do not have in-built frameworks for loading computations; switches for manual loading are absent. There is need to design a smart middleware gateway that is able to integrate with the operating system of smart devices for optimal loading. This is mainly because the existing ones are either rules-based or logic based, hence the challenge is in coming up with middleware architecture that self-learns from sensory data, battery status, and context inferences through and after ML of such data. Semantic attention is more suitable for IoT applications where data typically originates from a variety of sources for various purposes: 1) effectively integrating external commonsense knowledge graphs with image object attention allows for a deeper understanding of the interactions between objects in an image, enhancing the attention given to visual elements. This leads to captions that more accurately reflect the content and context of the scene compared to earlier methods, which lack semantic context or rely on scene graphs built solely from training data. The use of a semantic attention method facilitates this deeper understanding, enabling captions that more precisely capture both the content and context of the scene. This approach holds significant potential for a wide range of IoT applications, where data often comes from diverse sources, semantic attention mechanisms can more effectively capture the meaningful relationships between these different data types, improving the model's ability to interpret complex, multi-modal information.

## 6. Conclusion

This research proposes a novel approach to study marine animals from geospatial images with the help of modern methods such as AI and edge cloud IoT architecture. In this case, edge cloud IoT framework has been used for acquisition of remote images for marine life analysis. This remote sensing image has gone through normalization as well as noise reduction processing. The features were extracted and classification of this image done using a primary Gaussian convolution fuzzy encoding with a Bayesian reinforcement Markov approach. The findings confirm that it is possible to use edge computing for processing data in real time in order to quickly react to changes in the environment and their effects. This research is important in light of the increasing amount of literature on the effective deployment of AI and IoT for environmental monitoring along with what might be construed as a shift towards integrated application development. By presenting an extensive analysis of the effectiveness of various algorithms and the possibilities of their implementation in the environmental context, our research adds to the existing knowledge in this area and helps both researchers and practitioners think ahead about the ongoing evolution of this domain. In broad terms, the application of these technologies enhances the accuracy and efficiency of pollution detection as well as fosters the design of measures aimed at preventing environmental threats. Aiming at improving the environmental quality and protecting the existing aquatic fauna from further destruction, this paper rather underlines the constant need for evolution in such analytical processing schemes and edge computing in use.

## References

- [1] M. Jahanbakht, W. Xiang, N. J. Waltham, and M. R. Azghadi, "Distributed deep learning and energy-efficient real-time image processing at the edge for fish segmentation in underwater videos," *IEEE Access*, vol. 10, pp. 117796–117807, 2022.
- [2] S. J. Im, C. Yun, S. J. Lee, and K. R. Park, "Artificial Intelligence-Based Low-light Marine Image Enhancement for Semantic Segmentation in Edge Intelligence Empowered Internet of Things Environment," *IEEE Internet Things J.*, 2024.
- [3] L. M. A. El-Magd, A. Darwish, and A. E. Hassnien, "Fish Recognition Using MobileNet-V2 and MAR-Based Metaverse for an Educative Marine Life System," in *Artificial Intelligence for Environmental Sustainability and Green Initiatives*, Cham: Springer, 2024, pp. 127–145.
- [4] N. N. Ahamed and A. Ayesha, "Marine Resources: Identification, Restoring, and Monitoring of Fisheries Food Resources Using Deep Learning and Image Processing," in *Artificial Intelligence and Edge Computing for Sustainable Ocean Health*, Cham: Springer, 2024, pp. 101–121.

- [5] A. Algarni, T. Acarer, and Z. Ahmad, "An Edge Computing-Based Preventive Framework With Machine Learning-Integration for Anomaly Detection and Risk Management in Maritime Wireless Communications," *IEEE Access*, 2024.
- [6] H. S. Demir, J. B. Christen, and S. Ozev, "Energy-efficient image recognition system for marine life," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3458–3466, 2020.
- [7] M. Jahanbakht, W. Xiang, L. Hanzo, and M. R. Azghadi, "Internet of underwater things and big marine data analytics—a comprehensive survey," *IEEE Commun. Surveys Tuts*, vol. 23, no. 2, pp. 904–956, 2021.
- [8] A. S. Li *et al.*, "NASA NeMO-net's convolutional neural network: mapping marine habitats with spectrally heterogeneous remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5115–5133, 2020.
- [9] R. Manoharan, "Improving Security and Performance in Chaotic Optical Communication via Real-Time Pilot Signal Processing Techniques," *IETE J. Res.*, pp. 1–9, 2025.
- [10] M. Rajesh, S. Ramachandran, K. Vengatesan, S. S. Dhanabalan, and S. K. Nataraj, "Federated Learning for Personalized Recommendation in Securing Power Traces in Smart Grid Systems," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 88–95, Feb. 2024.
- [11] C. Prathima *et al.*, "Detecting and Recognizing Marine Animals Using Advanced Deep Learning Models," in *Proc. Int. Conf. Expert Clouds Appl. (ICOECA)*, 2024, pp. 950–955.
- [12] Á. F. Gambín, E. Angelats, J. S. González, M. Miozzo, and P. Dini, "Sustainable marine ecosystems: Deep learning for water quality assessment and forecasting," *IEEE Access*, vol. 9, pp. 121344–121365, 2021.
- [13] D. Tiwari, B. S. Bhati, B. Nagpal, S. Sankhwar, and F. Al-Turjman, "An enhanced intelligent model: To protect marine IoT sensor environment using ensemble machine learning approach," *Ocean Eng.*, vol. 242, p. 110180, 2021.
- [14] J. Yang, M. Xi, J. Wen, Y. Li, and H. H. Song, "A digital twins enabled underwater intelligent internet vehicle path planning system via reinforcement learning and edge computing," *Digit. Commun. Netw*, 2022.
- [15] L. Gao *et al.*, "AlgaeNet: A deep-learning framework to detect floating green algae from optical and SAR imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 2782–2796, 2022.
- [16] Y. Wang *et al.*, "Deep learning for genomic selection of aquatic animals," *Mar. Life Sci. Technol.*, pp. 1–20, 2024.
- [17] A. D. Mate *et al.*, "Marine Life Analysis Based on Ocean Water Level Rise and Climate Change Using Underwater Imaging Techniques," *Remote Sens. Earth Syst. Sci.*, pp. 1–13, 2024.
- [18] J. Li *et al.*, "Deep learning for visual recognition and detection of aquatic animals: A review," *Rev. Aquacult.*, vol. 15, no. 2, pp. 409–433, 2023.