International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
***Article Received**: 10 June 2022 **Revised**: 20 August 2022 **Accepted**: 25 August 2022 **Publication**: 15 September 2022*

Underwater Aerial Vehicle Networks Based Image Analysis By Deep Learning Architecture Integrated With 5G System

Aakansha Vyas
Associate Consultant
Performio, Melbourne
aakansha.vyas@performio.co

Dr. Anand Sharma
Asst. Prof., Department of CSE,
Mody University of Science and Technology,
Lakshmangarh
https://orcid.org/0000-0002-9995-6226
anand_glee@yahoo.co.in

Abstract:
With its astonishing ability to learn representation from data, deep neural networks (DNNs) have made efficient advances in the processing of pictures, time series, spoken language, audio, video, and many other types of data.In an effort to compile the volume of information generated in remote sensing field's subfields, surveys and literature revisions explicitly concerning DNNs methods applications are carried out Aerial sensing research has recently been dominated by applications based on Unmanned Aerial Vehicles (UAVs).There hasn't yet been a literature review that integrates the "deep learning" and "UAV remote sensing" thematics.This research propose novel technique in underwater aerial vehicle networks based image analysis by feature extraction and classification utilizing DL methods. here UAV based images through on 5G module is collected and this image has been processed for noise removal, smoothening and normalization. The processed image features has been extracted using multilayer extreme learning based convolutional neural networks. Then extracted deep features has been classified utilizingrecursive elimination based radial basis function networks. The experimental analysis is carried out based on numerous UAV image dataset in terms of accuracy, precision, recall, F-measure, RMSE and MAP.Proposed method attained accuracy of 96%, precision of 94%, recall of 85%, F- measure of 72%, RMSE of 48%, MAP of 41%.

Keywords: underwater aerial vehicle, feature extraction, classification, deep learning, 5G module.

1 Introduction:

Computer vision methods are utilized for a various processing tasks for studies utilizing remote sensing picture data.Applications utilizing statistical and Machine Learning (ML) methods are utilized mostly in classification as well as regression issues over last 10 years.With introduction of UAV, aerial imagery is popular technique for gathering data.They can also be called drones (multi-rotor, fixed-wing, hybrid, etc.) or remotely piloted aircraft (RPA) [1].Due of their comparatively inexpensive price as well as excellent operating competence to collect photographs swiftly and simply, these devices have enhanced in market availability.Large and intricate datasets might be handled because to the great spatial resolution of UAV-based images as well as its capability for multiple visits.In comparison to orbital as well as other aerial sensing techniques of acquisition, surface mapping with UAV platforms has some advantages [2].Systems for unmanned vehicles have gained popularity because of their capacity to adapt to challenging settings and carry out autonomous duties.Environments beneath the water are difficult, hazardous, and mostly unknown. The safest method for carrying out underwater operations is using AUVs as well as ROVs[3].Due to environment, autonomous underwater navigation continues to be a difficult and open subject.Cameras are restricted by water visibility as well as lighting conditions because water attenuates electromagnetic waves that would otherwise limit wireless communication as well as light-based perception.These restrictions do not apply to underwater sonar, but the acoustic images are loud and less useful.Authors have created hybrid vehicles that can navigate in both the air as well as water since there are activities that cannot be completed by vehicles that can only move in one environment [4].These vehicles are capable of carrying out inspection activities on partially submerged or fully submerged structures, such ship hulls or risers.Additionally, when diving, hybrid vehicles can capture high-resolution acoustic photos of the ocean floor as well as overhead images.

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
***Article Received**: 10 June 2022 **Revised**: 20 August 2022 **Accepted**: 25 August 2022 **Publication**: 15 September 2022*

In addition to this effect, it's important to note how different underwater picture processing techniques, such sea snowfall, enhance amount of dispersed light[5].To alter the existing colour schemes as well asimprove image for future processing, a pre-processing phase is therefore necessary.It might be approached from two different angles.The goal of image restoration is to recover a wavelet coefficient from an obtained image and a degradation model.The favoured technique, image improvement, makes advantage of contextually relevant individual features to enhance the aesthetics of an image.The main distinction between the two approaches is that, while picture restoration yields more realistic results, it also necessitates the estimation or measurement of a number of factors [6].

Contribution of this research is as follows:

1. To propose novel technique in underwater aerial vehicle networks based image analysis by feature extraction and classification using deep learning techniques
2. To collect the UAV based images through on 5G module and process for noise removal, smoothening.
3. To extract the features using multilayer extreme learning based convolutional neural networks
4. To classify extracted features using recursive elimination based radial basis function networks

2 Related works:

UAVs have gained popularity in recent years as an efficient, low-cost image-capture tool that may be used to accurately monitor aquatic habitats [7].Deep CNN activities like automatic categorization, object detection, and semantic segmentation have been quite popular recently [8].CNN methods like AlexNet, VGGNet, ResNet, DenseNet, and Inception, utilized for image classification or for semantic segmentation in combination with Fully Convolutional Network (FCN)[9].However, they completely ignore the top-layer spatial data, leading to a lack of precise positioning and definition of class boundaries.Every pixel in the image is intended to receive a set of predefined class labels through semantic segmentation.The patch-based CNN method [10], in which images are first separated into patches as well as then input into CNN networks, was employed in early studies on deep semantic segmentation.The surrounding picture patches are used by the network to forecast the label for the core pixel.Each pixel goes through this process once, which results in a high computational cost, especially in overlapping regions.[11] advocated using a Fully Convolutional Network (FCN) to address this issue. Encoder-decoder structure dilated convolutions and spatial pyramid pooling, two types of FCN-based semantic segmentation models, are explained below. Semantic segmentation frequently employs the encoder-decoder structure [12].First, the encoder uses convolutions, pooling, and an activation layer to create feature maps with high semantic level but low resolution. The decoder then performs an upsample operation on the low-resolution encoder feature maps to retrieve location data and produce fine-scaled segmentation output. Typical architectures having encoder-decoder topologies include SegNet [13] and U-Net. Recently, [14] employed SVM to categorise these interest points into vehicle as well as no-car classes based on SIFT descriptors after utilising SIFT to identify interest points of autos. They combined SIFT points for the same automobile at the end. The same authors suggested an approach in Reference [15] that used a sliding-window search and involved extracting HOG features through filtering operations in both the horizontal as well as vertical directions. After computing a similarity measure with a catalogue of cars as reference, cars were discovered. Following that, they presented an object detection approach in Reference [16] and used it to address issues of car detection as well as detection of solar panels in an urban setting. Higher-order gradients as well as GP regression are the method's foundations. A combination of Viola-Jones + SVM method and a HOG + SVM method utilizing a detector switching method based on various descending trends of detection speed of both methods was recently proposed by the authors in Reference [17] in order to improve detection efficiency. The authors worked on car detection in UAV street videos. The classification system is mostly constructed using handcrafted features in the aforementioned superficial ways. However, more recent DL based approaches have greatly outperformed handcrafted features [18].DL sometimes referred to as feature learning, relies on input data to automatically acquire effective feature representation. CNNs [21], SAE [20], and DBNs [19] are examples of common deep learning architectures. These techniques are currently thought to be the most successful ones for classifying images.

3 System model:

This section discuss novel technique in underwater aerial vehicle networks based image analysis by feature extraction and classification using deep learning techniques. here the UAV based images through on 5G module has been collected and this image has been processed for noise removal, smoothening and normalization. The processed

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
Article Received: 10 June 2022 Revised: 20 August 2022 Accepted: 25 August 2022 Publication: 15 September 2022

image features has been extracted using multilayer extreme learning based convolutional neural networks. Then extracted deep features has been classified using recursive elimination based radial basis function networks. The overall proposed architecture is shown in figure-1.

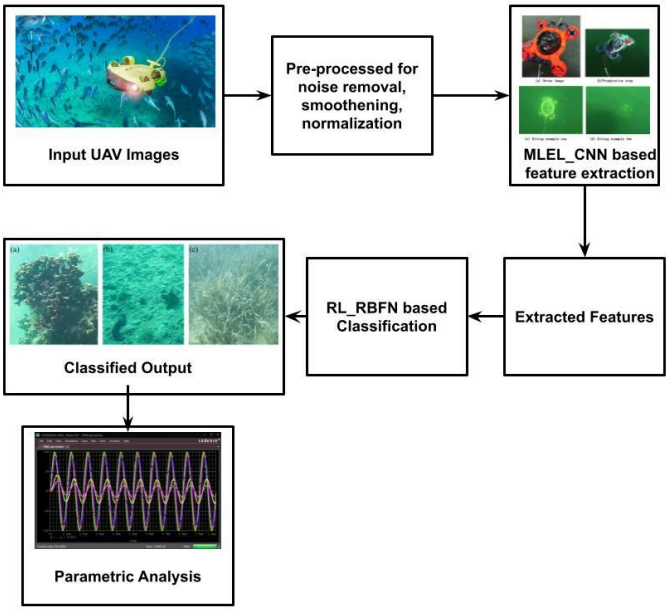


Figure 1. Overall proposed architecture

Aerial Image Processing:

Finding stationary markers in the environment is necessary for navigation in dynamic situations. In this work, recognising partially submerged buildings like piers, piles, and shorelines is of interest. Prior to robot navigation, a satellite provides us colour aerial view. Boats at harbours or marinas can be summed up as movable objects, as depicted in Fig. 2.The segments that are crucial for the matching procedure are extracted from aerial photos using semantic segmentation. To do semantic segmentation on the aerial photos, we employed a CNN.U-Net is well recognised for its skip connections that send leftover data.Many different applications, including image categorization, reconstruction, and translation, have made advantage of its architecture. The model takes as input a 256 × 256 × 3 colour aerial image, and outputs a 256 × 256 × 3 segmented image using one-hot encoding.5encode layers and 4 decode layers make up architecture, which keeps original padding to prevent cropping. Our final activation function employs a softmax layer. In order to conduct multi-categorical segmentation, we apply softmax cross entropy. The architecture was developed in the Tensor Flow framework and trained on an NVIDIA GTX Titan X with Adam optimizer as well as batch size of 16.In figure 2, UASN application scenario is depicted.

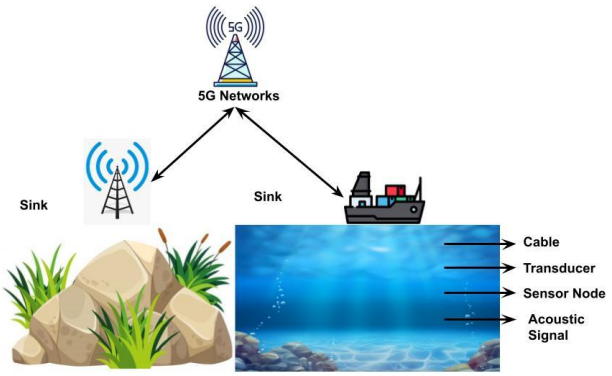


Figure-2 application scenario of UASN

Multilayer extreme learning based CNN based feature extraction:

The feature extraction issue can be expressed as eq. (1): Given N training samples X (x1 ; x2 ;... ; xN) $\in RN\times d$and associated labels Y = (y1 ; y2 ;... ; yN) $\in RN\times M$.

$$X = X^* + \psi \qquad (1)$$

where $\beta* = (\beta 1 * , \beta 2 * , \dots , \beta d*) \in Rd\times d$. From Eq. (6), it is clear that our goal is to discover the expression X $*$ = Xβ $*$ in order to extract features from X. Based on ELM, an optimization issue may be created to reduce classification error and redundancy feature for better classification, as specified by equation (2):

$$\min_{\beta,\beta^*} \|X - X\beta^*\|_F^2 + \|Y - H^{*T}\beta\|_F^2$$
$$H^* = [H^*(x_1^*), H^*(x_2^*), \dots, H^*(x_N^*)] =$$
$$x = x^* + \psi \begin{bmatrix} h^*(w_1, b_1, x_1^*) & \cdots & h^*(w_1, b_1, x_N^*) \\ \vdots & \ddots & \vdots \\ h^*(w_L, b_L, x_1^*) & \cdots & h^*(w_L, b_L, x_N^*) \end{bmatrix} \qquad (2)$$

T is the matrix transpose, and $\beta = (\beta 1 , \beta 2 , \dots , \beta M) \in RL\times M$. Our goal is to build a MELM to extract useful features as well as address ELM's poorly presented challenge, which could result in only moderately accurate classification.Bartlett's generalisation theory states that a lesser weight will lead to a training model with less inaccuracy. In order to do this, the MLEL optimization model is modified as eq. (2):

$$\min_{\beta^*\beta^*)} \frac{1}{2}\|\beta^*\|_F^2 + C/2\|\psi_i\|_2^2 + \frac{1}{2}\|Y - H^{*T}\beta\|_F^2 + \lambda \parallel \beta \parallel_1$$
$$\text{s.t. } x_i - x_i\beta^* = \psi_i; \text{i} = 1,2, \dots \text{N} \qquad (2)$$

where $\psi = (\psi 1 ; \psi 2 ; \dots ; \psi N) \in RN\times d$. As Eq. (9), variable splitting concept is used to impose sparse representation on

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
***Article Received**: 10 June 2022 **Revised**: 20 August 2022 **Accepted**: 25 August 2022 **Publication**: 15 September 2022*

ELM and establish new variables. The model in Equation (3) equals

$$\min_{(\beta,\beta^*,v)} \frac{1}{2}\|\beta^*\|_F^2 + C/2\|\psi_i\|_2^2 + \frac{1}{2}\|Y - H^{*T}\beta\|_F^2 + \lambda \| v \|_1 \quad (3)$$

$$\text{s.t. } x_i^* - x_i\beta^* = \psi_i; v = \beta; \text{i} = 1,2, \dots \text{N}$$

Aforementioned MSELM method is solved by ADMM methods as follows by applying augmented Lagrangian to Eq. (4).

$$\beta^* = \arg\min_{\beta^*}\{ \frac{1}{2}\|\beta^*\|_F^2 + C/2\sum_{i=1}^N \|\psi_i\|_2^2 + \sum_{i=1}^N\sum_{m=1}^d \gamma_{i,m}(x_i - x_i\beta^* - \psi_i)\}$$

$$\beta^{t+1} = \arg\min_\beta \left\{\frac{1}{2}\|Y - H^*\beta\|_F^2 + \frac{\lambda^*}{2}\|\beta - v^t - d^t\|_F^2\right\} \quad (4)$$

$$v^{t+1} = \arg\min_v \left\{\lambda \| v \|_1 + \frac{\lambda^*}{2}\|\beta^{t+1} - v - d^t\|_F^2\right\}$$

$$d^{t+1} = d^t - (\beta^{t+1} - v^{t+1})$$

where the Lagrange multiplies are represented by γ = [γ1 ; γ2 ; … ; γN] $\in RN\times d$. The Karush-Kuhn-Tucker (KKT) theory, or equation (5), $\beta*$can be used to determine.

$$\beta^* = X^T\left(\frac{l_2}{c} + XX^T\right)^{-1}X \quad (5)$$

βt+1 is solved by 1st order derivation eq. (6):

$$\beta^{t+1} = \left(H^*H^{*T} + \lambda^* I\right)^{-1}\left(H^*Y + \lambda^*(v^t + d^t)\right) \quad (6)$$

$$v^{t+1} = \text{soft}\left(\beta^{t+1} - d^t, \frac{\lambda}{\lambda^*}\right) \quad (7)$$

where I is an identity matrix, whose size is equivalent to dimension of H $*$H $*T$, λ and $\lambda*$t is index of iterations. For simplicity in implementation and parameter adjustment, we set $\lambda* = 10\lambda$. Let X = ($x\hat{}$1 , $x\hat{}$2 , … , $x\hat{}n$) $\in Rn\times d$. Equation (8) can be used to implement the test procedure for the proposed MLEL.

$$f(\hat{(}_i) = H^*(\hat{\imath}_i^*)\beta = H^*\left(\hat{\imath}_i^*\right)^T(H^*H^{,T} + \lambda^* I)^{-1}$$
$$(H^*Y + \lambda^*(v + d)) \quad i = l, 2, \dots, n \quad (8)$$

where $H*$ ($x\hat{\imath}*$) = [$h*$ (w1 , b1 , $x\hat{\imath}\beta*$), … , $h*$ (wL, bL, $x\hat{\imath}\beta*$)] T and $x\hat{\imath}*$ = $x\hat{\imath}\beta*$. Let $X*$ = (x1 $*$; x2 $*$; … ; $xN*$) $\in RN\times d$ be N training samples from $X*$ and Y = (y1 ; y2 ; … ; yN) $\in RN\times M$, where p represents number of pixels utilized in every training pixel's vicinity. As a result, eq. (9) may be used to define the MELM training model:

$$\beta_{SS}^{t+1} = (H_{SS}^* H_{SS}^{*\ T} + \lambda^* I)^{-1}\left(H_{SS}^* Y^* + \lambda^*(v_{SS}^t + d_{SS}^t)\right)$$

$$v_{SS}^{t+1} = \text{soft}\left(\beta_{SS}^{t+1} - d_{SS}^t, \frac{\lambda}{\lambda^*}\right)$$

$$d_{SS}^{t+1} = d_{SS}^t - (\beta_{SS}^{t+1} - v_{SS}^{t+1}) \quad (9)$$

where I is an identity matrix as well as its dimension based on dimension of $HSS*HSS*T$, $Y*$ = (Y, Y, . . , Y) $\in R(p+1)N\times M$, and $HSS*\in RL\times(p+1)N$ is represented by eq. (10)

$$H_{SS}^* = \begin{bmatrix} h^*(w_1, b_1, x_1^*) & \cdots & h^*(w_1, b_1, x_{Np}^*) \\ \vdots & \ddots & \vdots \\ h^*(w_L, b_L, x_1^*) & \cdots & h^*(w_L, b_L, x_{Np}^*) \end{bmatrix} \quad (10)$$

Equation (11) provides the testing procedure for the proposed MELM.

$$f(\hat{x}_i) = H_{SS}^*\left(\hat{\imath}_i^*\right)^T\beta_{SS} = H_{SS}^*(\hat{x}_i^*)^T(H_{SS}^* H_{SS}^* + \lambda^* I)^{-1}$$
$$\left(H_{SS}^* Y^* + \lambda^*(v_{SS} + d_{SS})\right), i = l, 2, \dots, n, \quad (11)$$

where $H_{SS}^*(\hat{\imath}_i^*) = \left[h^*(w_1, b_1, \mathring{A}_i^*), \dots, h^*(w_L, b_L, \Re_i^*)\right]^\top$

Algorithm of MLEL:

Input: Training sample pairs $X \equiv (x_1; x_2; \dots; x_N) \in R^{N\times a}$ and $Y = (y_1; y_2; \dots; y_N) \in R^{N\times M}$, where N is number of training samples; specifications$\lambda, L, C, d = 0$.

Training stage:

$H(\cdot)$: Sigmoid function.

β : Output weight from 3rd to output layer.

1: Solve optimization issue to get feature extraction specification:

$$\beta^* = \arg\min_{\beta^*}\left\{\frac{1}{2}\|\beta^*\|_F^2 + C/2\sum_{i=1}^N \|\psi_i\|_2^2 + \sum_{i=1}^N\sum_{m=1}^d \gamma_{im}\right.$$

$$\Rightarrow \beta^* \leftarrow X$$

2: Obtain significant feature: $X^* \leftarrow X\beta^*$;

3: Randomly produce input weights $\{w_1, \dots w_L\}$ and bias $\{b_1, \dots, b_L\}$, then evaluate3rd layer matrix

$$\text{H}(x_i^*) = [H_1(w_1 * x_i^* + b_1), \dots, H_L(w_L * x_i^* + b_L)]_{L\times 1}^T$$

4: Evaluate preliminary weight for β from 3rd layer to output layer: $\beta = (H^*)^\dagger Y^\top$

5.2β^{t+1} = arg $\min_\beta \left\{\frac{1}{2}\left\|Y - H^{*T}\beta\right\|_F^2 + \frac{\lambda^*}{2}\|\beta - v^t - d^t\|_F^2\right\}$

$\Rightarrow \beta^{t+1}$

$\leftarrow (H^*H^{-T} + \lambda^* I)^{-1}\left(H^*Y + \lambda^*(v^t + d^t)\right)5.2v^{t+1}$

$= \arg\min_v \left\{\lambda \| v \|_1 + \frac{\lambda^*}{2}\|\beta^{t+1} - v - d^t\|_F^2\right\}$ 5.3d^{t+1}

$\leftarrow d^t - (\beta^{t+1} - v^{t+1}) \quad \Rightarrow v^{t+1}$

$\leftarrow \text{sof } t\left(\beta^{t+1} - d^t, \frac{\lambda}{\lambda^*}\right)$

5.4 Enchancet to t + 1;

5.5 Quit method if stopping criterion is met go back to Step 5.2.

Prediction phase:

Input: $\hat{X} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) \in R^{n\times d}$

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
***Article Received**: 10 June 2022 **Revised**: 20 August 2022 **Accepted**: 25 August 2022 **Publication**: 15 September 2022*

1: Extract significant features: $\hat{x}_i^* = \hat{x}_i \beta^*$
2: Evaluate output layer matrix:
$$H^*(\hat{x}_i^*) = [h^*(w_1 \hat{x}_i^* + b_1), \dots, h^*(w_L \hat{x}_i^* + b_L)]_{l\times 1}^T; \mathrm{i} = 1, \dots, \mathrm{n}.$$
3: $f(\hat{x}_i) = H^*(\ell_i^*)\beta = H^*\left(\hat{(}_i^*\right)^T (H^* H^{*T} + \lambda^* I)^{-1}(H^* Y + \lambda^*(v + d)), \mathrm{i} = 1,2, \dots, \mathrm{n}$

FFNN that can process images is CNN.A convolutional layer, a pooling layer, and a fully connected layer make up the CNN.Multilayer Perceptron (MLP)-based CNNs were developed as a result of biological inspiration.Every layer functions differently and in a different way, and they have various sorts of levels.There are often many feature planes in the convolutional layer, each of which is made up of a number of neurons arranged in a rectangular pattern. Convolution kernels are the weights that are shared by neurons on same feature plane. The risk of overfitting can be decreased by reducing the number of connections between neurons. Down sampling, often referred to as pooling, can be thought of as a specific form of convolution that lowers processing requirements and enhances model generalisation .Method complexity is considerably reduced as well as method specifications are sampled and convolutionized. Forward as well as backward propagation are key components of CNN training method. Convolutional layer as well as down sampling layer alternately appear during training. Figure 3 depicts the convolutional network design.

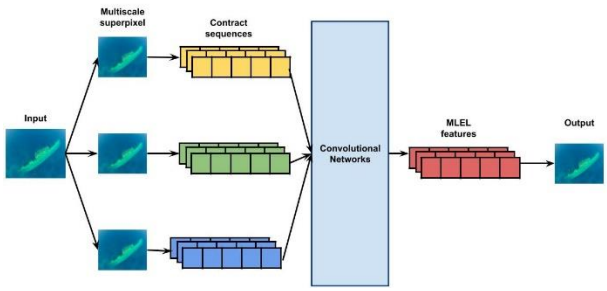


Figure 3. Convolutional neural network

Convolutional Layer: In order to build feature maps in response to various feature detectors, a convolutional layer extracts features of interest from the input and condenses the input by doing so. Neurons in first convolutional layer filter simple features like edges.In the subsequent convolutional layers, neurons learn to gather data to create a more comprehensive picture of image, leading to high-order feature detection.Every convolution kernel has the ability to extract features from entire input picture plane, The depth of stack is influenced by number of each type of neuron. Every type of neuron generates a slice of the stack using same weight and bias vector.

Convolution:
A number of specifications consists of input size, kernel size, depth of map stack, zero-padding, and stride, are used to define every convolutional layer.Equation (12) can be used to determine the output size.

$$\mathrm{M}_x = \frac{I_x - K_x}{S_x} \quad , \quad \mathrm{M}_y = \frac{I_y - K_y}{S_y} \qquad (12)$$

Activation:
There should still be an activation after weighted sum and the bias.LeCun mentioned using a sigmoid function to compress a pooling layer's output. Later, to boost performance, Jarrett et al. incorporate ReLUs into CNNs.ReLUs eventually gain acceptance as a common method for triggering convolutional outputs. ReLU is provided by equation (13):

$$f(x) = \max(0, x) \qquad (13)$$

Pooling Layer:
General-pooling, overlapping-pooling, and other sorts of operations are included in pooling, or subsampling.Typically, it acts as a bridge between various convolutional layers. For CNNs, pooling has numerous advantages. By focusing local data using a pooling window and lowering data dimensionality, pooling specifically tries to prevent overfitting. Reducing the dimensionality of data also speeds up calculations. Additionally, following adequate pooling, a few dislocations or scalings have little effect, resulting in invariance for translation, rotation, and scale.Non-overlapping max-pooling merely excites maximum result of every pooling region $(\mathrm{P}_x, \mathrm{P}_y)$over input picture and reduces its width and height by P_x and P_y, respectively.

Recursive elimination based radial basis function networks based classification:
The discriminant function in a linearly separable problem is ( )i i g x wx =·+ b, where w is a weight vector, and is training data i x xi ∈ n i =1,...,m.One may introduce slack variables i ξ which track a data point's departure from the ideal hyperplane, for the linearly non-separable situation. MLELCNN features are created by reducing equation (14)

$$\Phi(\mathbf{l}, \xi) = \frac{1}{2} \| \mathbf{w} \|^2 + C\sum_{i=1}^{n} \xi_i$$
$$\text{Subject to: } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \qquad (14)$$

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
***Article Received**: 10 June 2022 **Revised**: 20 August 2022 **Accepted**: 25 August 2022 **Publication**: 15 September 2022*

Wherey_i is corresponding to target, y_i $y_i = \{\pm 1\}, i = 1, \dots, m -$

Equation (15) resolves the optimization issue as a dual problem.

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j)$$

Subject to: 1) $0 \le \alpha_i \le C, i = 1, \dots, m$ (15)

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

where . α i are Lagrange coefficients by eq. (16)

$$\mathbf{w} = \sum_{k \in SY} y_k \alpha_k \mathbf{x}_k \quad (16)$$

$$w_i = \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K}(-i)\boldsymbol{\alpha}$$

Here, K(-i) denotes a kernel matrix that was constructed by removing the i-th feature from the input x.RFE retrains SVMs based on the remaining features by recursively removing features at every stage and reranking the remaining features.RFE will just remove a feature if it is a weak feature. A weak feature, however, could nonetheless be a crucial feature when combined with additional features.Therefore, just eliminating weak or duplicate characteristics may result in worse classification performance.We measure the classification performance after a putative weak feature has been deleted in terms of the corresponding value of w_i in order to determine the feature's significance.Even though this feature has the lowest value of w_i, we will preserve it if the classification performance suffers after it is eliminated.Additionally, the feature with the second-smallest value of will be taken into account.This method is repeated until a feature is discovered without which classification performance will not suffer. We refer to this technique as recursive feature elimination since features are first ranked according to their values of w_i (RE).The following describes this algorithm.

Algorithm

$F :=$ set of ranked feature $= \emptyset$
$R :=$ set of residual features $= \{1,2,3, \dots, n\}$
$S :=$ score of a criterion function
Number of remaining features, $k :=$ n,
Step 1: Train a SVM, evaluate score
$S_1 = S\{(x_i(R), y_i)\}$
Step 2: Evaluatew_i and rank features based on values of w_i, ₹ $= \{f_1, f_2, \cdots, f_k\}$.
Step 3: Set $k = k - 1$ and $j = 1$.-
Step 4: Build a new feature set R_l by removingfeature f_j from R.
Step 5: Train a SVM on samples with remaining features as well asevaluate score
$S_2 = S\{(x_i(R_1), y_i)\}$
Step 6: If $S_1 < S_2$,

$$\boldsymbol{F} = \boldsymbol{F} \cup \boldsymbol{f_j}, = 1$$

go to step 7
else if $j = k$,
remove first feature in ℑ

$$F = F \cup f_1, R = R - f_1$$

go to step 7
else

$$\mathbf{j} = \mathrm{j} + 1;$$

go to step 4.
Step 7: Repeat steps I - 6 until $R = \emptyset$

In Figure 4, a typical RBF is displayed.Every neuron in MISO-hidden RBFNN's layer is built as a Gaussian function because it is a k-m-1 network.Equation (17) gives a mathematical description of the RBFNN output.

$$\hat{y}(t) = \sum_{j=1}^{m} w_j(t)\theta_j(t) \quad (17)$$

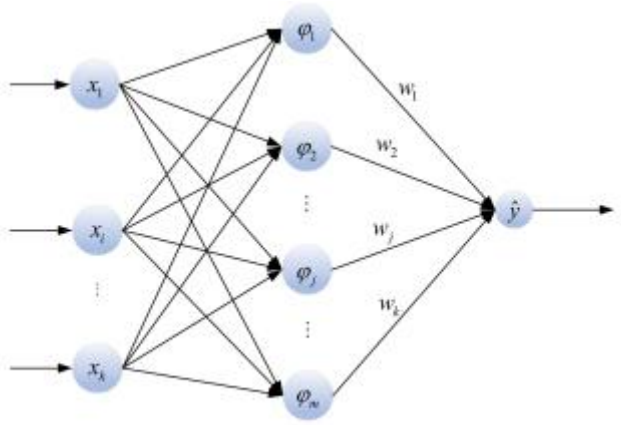


Figure 4. The structure of RBFNN.

where m is quantity of hidden neurons, wj (t) denotes weight between jth hidden and output layer at time t, and θj(t) denotes output of jth hidden layer neuron, as shown in equation (18):

$$\theta_j(t) = e^{-\frac{\mathrm{lx}(t) - \varepsilon_i(t)]^2}{2\sigma_j^2(t)}}$$

$$\mathbf{c}_j(t) = \left[c_{1,j}(t), c_{2,j}(t), \cdots, c_{n,j}(t)\right]^T \quad (18)$$

Equation (19) describes the generalisation error for the entire input space.

$$E_{g_{en}}(t) = \int_D [f(\mathbf{x}(t)) - F(\mathbf{x}(t))]^2 p(\mathbf{x}(t)) d\mathbf{x}(t) \quad (19)$$

Type 1: zero-order polynomial by eq. (20)

$$f_i(x_{k1}, \dots, x_{kl}, v_i) = a_{i0}. \quad (20)$$

Type 2: 1st order polynomial by eq. (21)

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
Article Received: 10 June 2022 Revised: 20 August 2022 Accepted: 25 August 2022 Publication: 15 September 2022

$$f_i(x_{k1,\dots,x_{kd}},\mathbf{v}_i)= \quad a_{i0} + a_{i1}(x_{k1} - v_{i1}) + a_{i2}(x_{k2} - v_{i2}) + \cdots + a_{i1}(x_{kl} - v_{i1}). \tag{21}$$

Type 3: second-order polynomial by eq. (22)

$$\begin{aligned} f_i(x_{k1},\dots,x_{k1},v_i) = \; & a_{i0} + a_{i1}(x_{k1} - v_{i1}) \\ & + a_2(x_{k2} - v_{i2}) + \cdots + a_{il}(x_{kl} - v_{i1}) \\ & + a_{i(l+1)}(x_{k1} - v_{i1})^2 + a_{i(l+2)}\left(x_{k2} - v_{f2}\right)^2 \\ & + \cdots + a_{i(2i)}(x_{kl} - v_{il})^2 \\ & + a_{i(2l+1)}(x_{k1} - v_{i1})(x_{k2} - v_{i2}) \\ & + \cdots + a_{i(((l+1)(1+2)/2)} \\ & \times \left(x_{k(l-1)} - v_{i(l-1)}\right)(x_{kl} - v_{il}). \end{aligned} \tag{22}$$

The network's empirical inaccuracy is given by equation (23):

$$E_{emp}(t) = \frac{1}{N}\sum_{a=1}^{N}[f(\mathbf{x}_a(t)) - F(\mathbf{x}_a(t))]^2 \tag{23}$$

where the approximation and actual mapping functions between ath input as well as output in input space, are represented by f(xa(t)) and F(xa(t)).Minimizing approximation error is the ultimate objective of increasing generalisation ability because it allows the network to directly predict previously unknown input.RBFNN is a local technique, for every sample xa(t) ∈ D, determine sample set:$\mathbf{X}_{S,\mathbf{x}_i}(t) = \{\mathbf{x}(t) \mid \mathbf{x}(t) = \mathbf{x}_a(t) + \Delta\mathbf{x}(t)\}$

We may derive description of local generalisation error bound as eq. (24) from Hoeffding's inequality

$$\begin{aligned} E_{\text{gen},\,S}(t) &= \sqrt{E_{emp}(t)} + \sqrt{E_{\mathbf{X}_{\varsigma,\Delta y^2}(t)}} \\ \Delta y(t) &= \hat{f}(\mathbf{x}_a(t)) - F(\mathbf{x}_a(t)) \\ E_{\mathbf{X}_{s,\Delta y^2}}(t) &= \frac{1}{N}\sum_{a=1}^{N}\int_{\mathbf{X}_s(\mathbf{x}_a(t))}(\Delta y^2(t))\frac{1}{(25)^1}dx(t) \end{aligned} \tag{24}$$

where the term EXS, Δy 2 (t) refers to the output of the network divided by the real value, and Δy(t) denotes the difference (t).

If inputs are independent and not uniformly distributed, every input feature will have its own expectation μxi and variance δ 2 xi, according to the equation (25)

$$\phi_j(t) = w_j^2(t)e^{\delta_{d_j}(t)/2\sigma_j^4(t) - \mu_{d_j}(t)/\sigma_j^2(t)} \tag{25}$$

With eq. (26):

$$\begin{cases} d_j(t) = \|\mathbf{x}(t) - \mathbf{c}_j(t)\|^2 \\ \delta_{d_j}(t) = \sum_{i=1}^{n}\mu\left[\left(x_i(t) - \mu_{x_i}(t)\right)^4\right] - \left(\delta_{x_i}^2(t)\right)^2 + 4\delta_{x_i}^2(t)\left(\mu_{x_i}(t) - c_{ji}(t)\right)^2 + \\ 4\mu\left[\left(x_i(t) - \mu_{x_i}(t)\right)^3\right]\left(\mu_{x_i}(t) - c_{ji}(t)\right) \end{cases} \tag{26}$$

We do not impose a strict data distribution constraint on the input variation as long as it is finite. In this case, we suppose that unseen samples in training samples' S-neighborhood follow uniform distribution, and as a result, we get δ 2 Δxi (t) = S 2/3. According to law of large numbers, RBFNN's sensitivity by eq (27),

$$\begin{aligned} & E_{\mathbf{X}_{s,\Delta y^2}}(t) \\ & = \frac{1}{N}\left\{\sum_{a=1}^{N}\int_{\mathbf{X}_{s\mathbf{x}_0}}[f(\mathbf{x}_a(t) + \Delta\mathbf{x}(t)) - f(\mathbf{x}_a(t))]^2 p(\Delta\mathbf{x}(t))d\Delta\mathbf{x}(t)\right\} \\ & \approx \sum_{j=1}^{m}\phi_j(t)\left\{\sum_{i=1}^{m}\left[\delta_{\Delta x_i}^2(t)\left(\delta_{x_j}^2(t) + \left(\mu_{x_j}(t) - c_{ji}(t)\right)^2 + 0.2\delta_{\Delta x_i}^2(t)\right)\right]/\sigma_j^4(t)\right\} \end{aligned} \tag{27}$$

$$\phi_j(t) = \sigma_j^4(t)\xi_j(t).$$

therefore, we can obtain by eq. (28):

$$E_{\mathbf{X}_{s,\Delta y^2}}(t) \approx \frac{1}{45}S^4 n\sum_{j=1}^{m}\xi_j(t) + \frac{1}{3}S^2\sum_{j=1}^{m}\sigma_j(t). \tag{28}$$

4 Performance analysis:

Tensorflow, an open-source, end-to-end machine learning framework, and the Keras high-level API were used in experiments carried out in the Python 3.7 programming environment.On an Intel i7 8700 (3.2GHz) computer with a CUDA-enabled NVIDIA GeForce RTX 2070 GPU and 8GB of parallel computing-ready memory, the training and inference operations were carried out.

Dataset description:

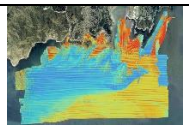
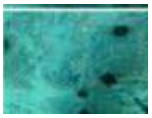
Real Word Dataset: the multiclass image classification and object recognition datasets now in use (Anti-UAV, ImageNet, MS-COCO, PASCAL VOC).Real-world film was used, converted into photos, and hand-labeled to produce a unique set of 56,821 images and 55,539 bounding boxes in order to increase the model's effectiveness.Compared with other UAV datasets, the Real Word dataset contains the most types of UAV and environments, and the image resolution is low, because all the data are obtained from YouTube videos, while other datasets are collected by researchers themselves. Due to the limitation of shooting perspective, most of the data in the Real World dataset is in flat view and elevation.

Det-Fly Dataset: a brand-new dataset called Det-Fly that includes more than 13,000 photos of a moving target UAV that were captured by a moving UAV.Det-Fly dataset is more extensive as compared to the current datasets because it includes various realistic scenarios with various background scenes, relative distances, flying altitudes, and lighting conditions.Det-Fly dataset overcomes the shortcomings of UAV data from a single perspective. The camera collects the target UAV directly in the air, including a variety of UAV postures under elevation, pitch, and horizon. However, the data set contains only one type of UAV, so that the model cannot be used for other types of

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
***Article Received**: 10 June 2022 **Revised**: 20 August 2022 **Accepted**: 25 August 2022 **Publication**: 15 September 2022*

UAV detection. Regarding the other image-based UAV dataset, the size of UAV in the MIDGARD dataset is relatively large, and the appearance and outline of UAV are very clear. The camera and UAV are very close, but it is impossible to study the detection problem in long distance. The images in the USC-Drone dataset are taken by people standing on the ground with hand-held cameras, and the UAV's perspective is too single. In addition, there are many unmarked images in this dataset that cannot be used directly. These datasets also contain only one type of UAV and a richer environment, but also has the disadvantage of a single viewing angle, so cannot be used for MOT tasks.

Table-1 UAV image analysis based on proposed technique

Dataset	Input image	Pre-processed image	Feature extracted image	Classified image
Real Word Dataset				
Det-Fly Dataset				

The above table-1 shows UAV image analysis based on proposed technique for various dataset. Here the image analysis has been carried out for real world dataset and Det-fly dataset. As shown above the input image has been processed, then the extracted features has been obtained and then classified image is obtained for extracted features.

Table-2 Comparative analysis for real world dataset between proposed and existing technique

Parameters	SIFT_SVM	HOG_SVM	UAV_IA_DL_5G
Accuracy	88	91	95
Precision	71	73	75
Recall	66	69	73
F_Measure	55	61	65
RMSE	65	63	55
MAP	61	54	52

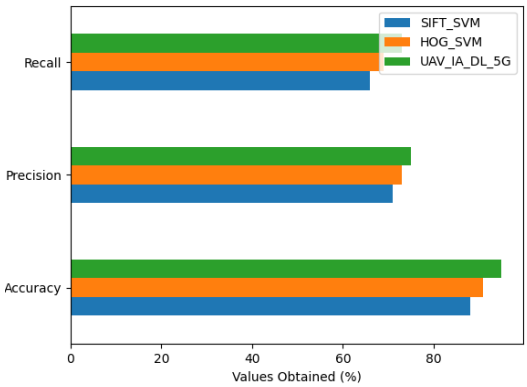


Figure-5Comparative analysis of real world dataset in terms of accuracy, precision, recall

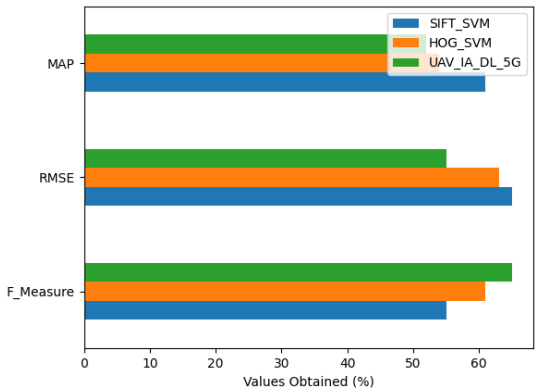


Figure-6 Comparative analysis of real world dataset in terms of MAP, RMSE, F-measure

The above table-2 represents comparative analysis between proposed and existing technique. here parametric analysis is carried out in terms of accuracy, precision, recall, F-measure, RMSE, MAP. Proposed method attained accuracy of 95%, precision of 75%, recall of 73%, F- measure of 65%, RMSE of 55%, MAP of 52%. Existing technique

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
***Article Received**: 10 June 2022 **Revised**: 20 August 2022 **Accepted**: 25 August 2022 **Publication**: 15 September 2022*

SIFT_SVM attained accuracy of 88%, precision of 71%, recall of 66%, F-measure of 55%, RMSE of 65%, MAP of 61%; HOG_SVM attained accuracy of 91%, precision of 73%, recall of 69%, F-measure of 61%, RMSE of 63% and MAP of 54% as shown in figure 5 and 6.

Table- 3Comparative analysis of between proposed and existing technique for Det-Fly dataset

Parameters	SIFT_SVM	HOG_SVM	UAV_IA_DL_5G
Accuracy	86	92	96
Precision	88	92	94
Recall	75	79	85
F_Measure	65	68	72
RMSE	54	51	48
MAP	51	45	41

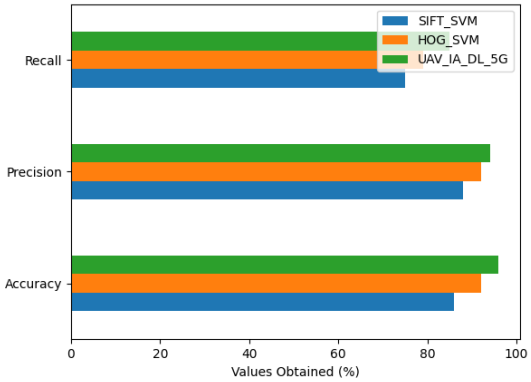


Figure-7 Comparative analysis of Det-Fly dataset in terms of accuracy, precision, recall

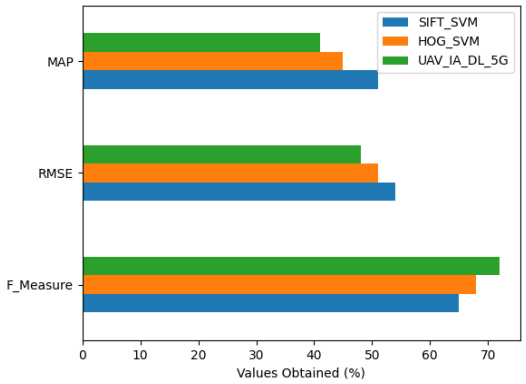


Figure-8 Comparative analysis of Det-Fly dataset in terms of MAP, RMSE, F-measure

The above table-3 shows comparative analysis between proposed and existing technique. The proposed method attained accuracy of 96%, precision of 94%, recall of 85%, F- measure of 72%, RMSE of 48%, MAP of 41%. Existing technique SIFT_SVM attained accuracy of 86%, precision of 88%, recall of 75%, F-measure of 65%, RMSE of 54%, MAP of 51%; HOG_SVM attained accuracy of 92%, precision of 92%, recall of 79%, F-measure of 68%, RMSE of 51% and MAP of 45% as shown in figure 7 and 8.By adding more training samples, DL can achieve higher target recognition accuracy. Development of cluster system has allowed for the use of several AUVs in cooperative work to circumvent the limitations of information collection from a single perspective and collect target data from various angles. Utilizing a wide range of marine data effectively helps mitigate or lessen the effects of the ocean's unique undersea environment. To further enhance underwater target recognition will be an essential research direction. The accuracy of target recognition cannot be assured solely by training dataset due to the emergence of diverse shapes for underwater dangerous targets and shapes of unknown opponent dangerous targets.

5 Conclusion:

In this research the proposed framework has been designed for underwater image analysis based on feature extraction and classification. The processed input features has been extracted usingmultilayer extreme learning based convolutional neural networks. Then extracted deep features has been classified using recursive elimination based radial basis function networks. On order to classify individual photos better than other image enhancement, CNN has been trained in image restoration techniques.With just one image as an input, the suggested method can generate images with high image restoration quality.Images from various places and attributes are used to validate NNs generalisation ability.A real-time deep learning approach for picture categorization is put forth in this work and contrasted with other cutting-edge options.Other restoration techniques that call for many inputs that are challenging to evaluate at time of intervention are used to train the system.However, after training, the system can correctly classify images in real time using only a still raw image as input.Future scope of this research can be carried out for image recognition with classification based metaheuristic techniques.

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 3
DOI: https://doi.org/10.17762/ijfrcsce.v8i3.2092
***Article Received**: 10 June 2022 **Revised**: 20 August 2022 **Accepted**: 25 August 2022 **Publication**: 15 September 2022*

Reference:

[1]. Cao, F., Yang, Z., Ren, J., Chen, W., Han, G., & Shen, Y. (2019). Local block multilayer sparse extreme learning machine for effective feature extraction and classification of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(8), 5580-5594.

[2]. Ding, S., Zhang, N., Xu, X., Guo, L., & Zhang, J. (2015). Deep extreme learning machine and its application in EEG classification. *Mathematical Problems in Engineering*, *2015*.

[3]. Yang, A., Yang, X., Wu, W., Liu, H., & Zhuansun, Y. (2019). Research on feature extraction of tumor image based on convolutional neural network. *IEEE access*, *7*, 24204-24213.

[4]. Chen, X. W., & Jeong, J. C. (2007, December). Enhanced recursive feature elimination. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)* (pp. 429-435). IEEE.

[5]. Huang, W., & Wang, J. (2013). Design of polynomial fuzzy radial basis function neural networks based on nonsymmetric fuzzy clustering and parallel optimization. *Mathematical Problems in Engineering*, *2013*.

[6]. Yang, Y., Wang, P., & Gao, X. (2022). A novel radial basis function neural network with high generalization performance for nonlinear process modelling. *Processes*, *10*(1), 140.

[7]. Ramkumar, G., Ayyadurai, M., & Senthilkumar, C. (2021, May). An effectual underwater image enhancement using deep learning algorithm. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1507-1511). IEEE.

[8]. Xu, S., Zhang, J., Qin, X., Xiao, Y., Qian, J., Bo, L., ... & Zhong, Z. (2022). Deep retinex decomposition network for underwater image enhancement. *Computers and Electrical Engineering*, *100*, 107822.

[9]. Sarma, K., & Vigneshwaran, P. (2021, May). Underwater Image Enhancement Using Deep Learning. In *International Conference on Image Processing and Capsule Networks* (pp. 431-445). Springer, Cham.

[10]. Estrada, D. C., Dalgleish, F. R., Den Ouden, C. J., Ramos, B., Li, Y., & Ouyang, B. (2022). Underwater LiDAR image enhancement using a GAN based machine learning technique. *IEEE Sensors Journal*, *22*(5), 4438-4451.

[11]. Zheng, Y., & Yang, Y. (2022). Underwater Image Processing and Target Detection Algorithm Based on Deep Learning. In *International Conference on Multi-modal Information Analytics* (pp. 368-374). Springer, Cham.

[12]. Maniyath, S. R., Vijayakumar, K., Singh, L., Sharma, S. K., & Olabiyisi, T. (2021). Learning-based approach to underwater image dehazing using CycleGAN. *Arabian Journal of Geosciences*, *14*(18), 1-11.

[13]. Wang, N. (2022, April). Underwater color restoration and dehazing based on deep neural network. In *Journal of Physics: Conference Series* (Vol. 2234, No. 1, p. 012016). IOP Publishing.

[14]. Kharazi, B. A., & Behzadan, A. H. (2021). Flood depth mapping in street photos with image processing and deep neural networks. *Computers, Environment and Urban Systems*, *88*, 101628.

[15]. Li, F., Lu, D., Lu, C., & Jiang, Q. (2022). Underwater Imaging Formation Model-Embedded Multiscale Deep Neural Network for Underwater Image Enhancement. *Mathematical Problems in Engineering*, *2022*.

[16]. Politikos, D. V., Fakiris, E., Davvetas, A., Klampanos, I. A., & Papatheodorou, G. (2021). Automatic detection of seafloor marine litter using towed camera images and deep learning. *Marine Pollution Bulletin*, *164*, 111974.