



Intelligent Fish Behavior Classification Using Modified Invasive Weed Optimization with Ensemble Fusion Model

B. Keerthi Samhitha* and R. Subhashini

School of Computing, Sathyabama Institute of Science and Technology, Chennai, 600 119, India

*Corresponding Author: B. Keerthi Samhitha. Email: samhithab711@gmail.com

Received: 26 March 2023; Accepted: 17 May 2023; Published: 11 September 2023

Abstract: Accurate and rapid detection of fish behaviors is critical to perceive health and welfare by allowing farmers to make informed management decisions about recirculating the aquaculture system while decreasing labor. The classic detection approach involves placing sensors on the skin or body of the fish, which may interfere with typical behavior and welfare. The progress of deep learning and computer vision technologies opens up new opportunities to understand the biological basis of this behavior and precisely quantify behaviors that contribute to achieving accurate management in precision farming and higher production efficacy. This study develops an intelligent fish behavior classification using modified invasive weed optimization with an ensemble fusion (IFBC-MIWOF) model. The presented IFBC-MIWOF model focuses on identifying the distinct kinds of fish behavior classification. To accomplish this, the IFBC-MIWOF model designs an ensemble of Deep Learning (DL) based fusion models such as VGG-19, DenseNet, and EfficientNet models for fish behavior classification. In addition, the hyperparameter tuning of the DL models is carried out using the MIWO algorithm, which is derived from the concepts of oppositional-based learning (OBL) and the IWO algorithm. Finally, the softmax (SM) layer at the end of the DL model categorizes the input into distinct fish behavior classes. The experimental validation of the IFBC-MIWOF model is tested using fish videos, and the results are examined under distinct aspects. An Extensive comparative study pointed out the improved outcomes of the IFBC-MIWOF model over recent approaches.

Keywords: Fish behavior; aquaculture; computer vision; deep learning; invasive weed optimization; fusion model

1 Introduction

The demand for aquatic products has significantly increased over the past 20 years due to the world population's fast growth [1]. Many countries' principal source of fishery products has shifted to fish farming due to the rapidly increasing demand for aquatic goods. However, the lack of efficient fish health and management practices and welfare monitoring is causing this industry to suffer tremendous



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

losses on a global scale [2]. Animals frequently engage in complex interactions with one another and with their environment. Tools that measure behavior automatically and simultaneously provide in-depth knowledge of the relationship between the external environment and behavior are required since it is possible to distinguish behavioral activities from video files [3]. In fish farming research, how the yield and efficiency can be improved turns out to be a significant problem, and this becomes a severe issue that should be sorted out that limits the growth of the aquaculture sector [4]. The premise for solving this issue is obtaining the fish's behavior analysis via the behavior trajectory, identifying the irregularities and rules to guide the breeding and thwart the occurrence of abnormal conditions [5].

To proceed with the fish behavior analysis, the location and recognition become essential for the fish after examining the fish's behavior. Currently, the prevailing techniques for analysis and recognition are categorized into quantitative and qualitative methods [6]. Qualitative techniques have three types. Firstly, the complete manual approaches for analysis and recognition, like the manual marking technique or manual observation approach. Secondly, the fluorescent-related credit and manual analysis approach employed fluorescent markers to locate the fish and physically analyze the behavior [7]. Thirdly the video-related award and manual analysis approach. Thirdly, a feature was put into the fish's body, and the video was utilized for tracking the marker in the fish's body. This approach so far requires physical behavior analysis for fish [8]. The following two types of techniques were all invasive; thus, such techniques results in the stress behavior of fish, and the behavior analysis only is performed manually. These methods were laborious and time-consuming, and the behavior analysis outcomes were vulnerable to subjective elements. In addition, only the qualitative effects are acquired instead of the quantitative results, and the behavior trajectory may not be plotted.

To make a solution to the issue, the subjective elements can influence the manual analysis, and the quantitative outcomes may not be attained; the quantitative approaches were suggested for analyzing the fish behavior. The quantitative methods have two kinds. One is related to appearance features for recognizing the objects and obtaining the behavior trajectory, like infrared reflection and shape matching [9]. Another is related to image processing for identifying objects and obtaining the behavior trajectory, like the optical flow method and inter-frame difference technique. Introducing the Computer Vision (CV) approach in aquaculture has created new options for accurately and efficiently observing fish behavior. Conventional CV methods to identify fish behavior depend on deriving significant features, namely trajectory, motion patterns, and clustering index [10]. Deep learning (DL) techniques have revolutionized the ability to mechanically examine videos utilized for live fish identification, water quality monitoring, species classification, behavior analysis, and biomass estimation. The benefit of DL is it can mechanically learn to extract image features and exhibits remarkable performance in identifying sequential activities.

Deep learning algorithms have produced accurate target detection and tracking. However, it is still necessary to create strategies for recognizing fish behavior to consider the complex underwater environment and swift target movement. In reaction to environmental stimuli, fish typically alter their swimming technique. Due to the chance that two dissimilar kinds of activities in a video sequence display comparable activity, the success probability of recognition is lowered. For instance, both eating exercise and threatening behavior involve fish moving quickly after being aroused. An extensive range of various activity states is also included in one category of behavior. For instance, hypoxic fish are sinking on the tank's side and actively congregating there. Therefore, tiny alterations among dissimilar behavior categories and frequent samples of the similar action pose the biggest obstacles to implementing deep learning in fish behavior identification.

This study proposes an IFBC-MIWOEF model that uses temporal and spatial feature information to read a school of fish's behavior states from video data to overcome the abovementioned challenges. Our work has made the following main contributions: (1) An innovative method for detecting fish activity based on deep learning. (2) The proposed framework combines the RGB and optical flow image elements to identify the five fish behaviors. (3) The suggested framework exhibits good performance and acceptable outcomes. By automatically identifying fish behavior in real-time from video clips using our suggested method, we lay the framework for aquaculture to develop into a more intelligent industry.

This study develops an intelligent fish behavior classification using modified invasive weed optimization with an ensemble fusion (IFBC-MIWOEF) model. The IFBC-MIWOEF model's recognition and classification of various types of fish behavior is one of its main objectives. Primarily, the input videos are collected, and the frame conversion process takes place. The VGG-19, DenseNet, and EfficientNet models also employ a fusion-based feature extraction method. Additionally, the IWO algorithm and OBL concept are combined to create the MIWO algorithm, which is used to tune the hyperparameters of the DL models. Further, the input is divided into various fish behavior classes by the softmax (SM) layer after the DL model. Several simulations utilizing various measures were run to show the effectiveness of the IFBC-MIWOEF model.

2 Related Works

A thorough analysis of the models for categorizing fish behavior is provided in this section. Iqbal et al. [11] provided an automatic mechanism for classifying and detecting fish. It is helpful for marine biologists to get a better understanding of their habitats and fish species. The presented method depends on Deep Convolutional Neural Networks (DCNN). It employs a minimized version of the AlexNet method and has two fully connected (FC) layers and four convolutional layers. A comparison was made with the other DL methodologies, namely AlexNet and VGGNet. Jalal et al. [12] projected a hybrid method combining Gaussian mixture methods and optical flow with YOLO Deep Neural Networks (DNN) to locate and categorize fish in unrestricted underwater footage. The YOLO-related object detection mechanism was initially used to capture just the static and visible fish. And removes this restriction of YOLO, enabling it to detect freely moving fish, invisible in the background, utilizing temporary data obtained through optical flow and Gaussian mixture methods.

For automatic fish species categorization, Tamou et al. [13] used transfer learning (TL) in conjunction with CNN AlexNet. The pre-trained AlexNet, either with or without finetuning, generates features from foreground fish photos of the existing underwater data set. Chhabra et al. [14] suggested a fish classifier system that functions in a natural underwater atmosphere to detect fishes with nutritional or medicinal significance. The article used a hybrid DL method, whereas the Stacking ensemble method was employed for detecting and classifying fishes from images, and a pre-trained VGG16 technique was utilized for Feature Extraction. According to [15], machine vision is offered to assess fish appetite, and CNN automatically scores fish feeding intensity. Following were the specific implementation steps. First, photos taken throughout the feeding process have been collected, and a data set has been expanded using noise-invariant data expansion and rotation, scale, and translation (RST) augmentation methods. Afterward, a CNN was trained over the trained data set, and the fish appetite level could be graded using the trainable CNN method.

Farooq et al. [16] proposed a novel method to identify unusual crowd behavior in surveillance footage. The main goal of this study is to identify crowd divergence behaviors that may result in dangerous scenarios like stampedes. We categorize crowd activity using a convolution neural network

(CNN) trained on motion-shape images (MSIs), and we provide the potential for physically capturing motion in photographs. The finite-time Lyapunov exponent (FTLE) field is generated after the optical flow (OPF), which was initially determined, has been integrated. The coherent Lagrangian structure (LCS) represents crowd-dominant motion in the FTLE field. It is suggested to transform LCS-to-grayscale MSIs using ridge extraction. Not to mention that CNN has received supervised training to foresee divergence or typical behaviors for each unidentified image. Describe a deep-fusion model that utilizes the numerous convolutional layers of a deep neural network's hierarchical feature exits effectively and efficiently in [17]. We specifically recommend a network that combines multiscale data from the network's surface to deep layers to transform the input image into a density map. The final crowd size is calculated by adding the peaks on the density map. On the benchmark datasets UCF_CC_50, ShanghaiTech, and UCF-QNRF, experiments are conducted to evaluate the performance of the future deep network.

Alzahrani et al. [18] proposed a framework for a deep model that automatically classifies crowd behavior based on motion and appearance. This study first extracts dense trajectories from the input video segment and then projects those trajectories onto the image plane to generate the trajectory image. The trajectory image effectively depicts the scene's relative motion. We employ a stack of trajectory images to teach a deep convolutional network a compact and potent representation of scene motion. Reference [19] proposed a technique for initializing a dynamical system described by optical flow, a high-level measure of overall movement, by superimposing a grid of particles over a scene. The dynamical scheme's time integration generates brief particle trajectories (tracks), which reflect dense but transient motion designs in scene fragments. Then, trackless are expanded into longer routes, clustered using an unsupervised clustering technique, and compared for similarity using the longest common subsequence.

Knausgård et al. [20] published a two-step DL method to characterize temperate fishes without prefiltering. The primary stage is to identify every fish species, regardless of sex or species, that can be seen in the image. From now on, make use of the YOLO object detection technique. Each fish in the picture is classified in the second stage using a CNN without prefiltering and the Squeeze-and-Excitation (SE) structure. And uses Transfer Learning (TL) to progress the classifier's performance and get around the poorly trained samples of temperate fish. A new image-related technique for fish identification was suggested by Revanesh et al. [21] to drastically reduce the number of undersized fish caught during commercial trawling. The suggestion concerns the trawl's direct-located Deep Vision imaging mechanism's acquisition of stereo images and processing those data. The pre-processing of the photos correctly accounted for the camera responses' non-linearity. Afterward, each fish in the pictures may be localized and segmented using a Mask Recurrent Convolutional Neural Networks (RCNN) framework.

3 The Proposed Model

To identify and categorize various types of fish behavior, a novel IFBC-MIWOF model has been established in this work. Primarily, the input videos are collected, and the frame conversion process takes place. Then, the fusion-based feature extraction process is carried out through three DL models, namely VGG-19, DenseNet, and EfficientNet. The performance of the DL algorithms can be improvised using the MIWO algorithm-based hyperparameter tuning process. The SM layer assigns the correct class labels to the input video frames at the very end. Fig. 1 portrays the block diagram of the IFBC-MIWOF method.

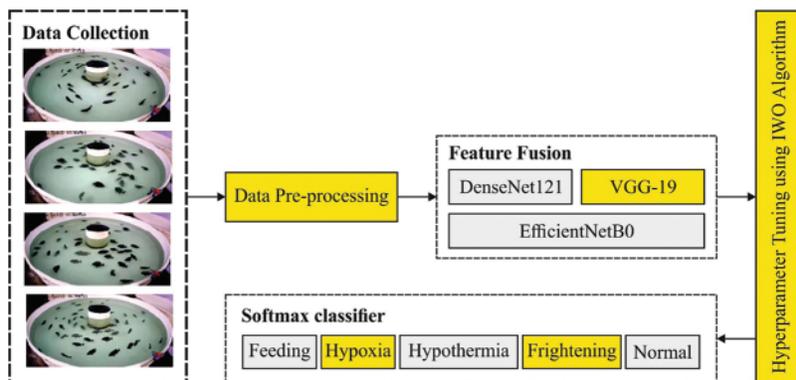


Figure 1: Block diagram of IFBC-MIWOEF method

3.1 Data Collection

A collection of behavior films of fish of various sizes, times, and quantities can be collected over three months to conduct the experiment in the best possible way and verify the proposed technique's effectiveness. Since fishes typically act in groups, every fish was photographed using the top camera [22]. The video is frequently downloaded into our computer from the memory card of network cameras and preserved there. In 24 h, several stages of fish behavior will be observed and recorded for the study. Other behaviors that are influenced by the environment, besides general behaviors like feeding, include fish startle stimulation, hypoxia, and hypothermia. They are categorized as hypothermia, eating, normal, hypoxic, and terrifying behaviors using expert experiences and talents.

3.2 Fusion Based Feature Extraction

The DL mechanisms are combined in the weighted voting-based ensemble approach, and the highest results are carefully selected using this method. Each vector is utilized for training the voting mechanism, and the fitness function (FF) is then used to determine the tenfold cross-validation accuracy. Consider the D base classification models and n number of classes for voting, the predictive class c_k of weighted voting for each sample, k is defined by

$$c_k = \arg \max_j \sum_{i=1}^D (\Delta_{ji} \times w_i), \quad (1)$$

Now, Δ_{ji} indicates the binary variable. After the i -th base classification technique categorizes the k sample into the j -th classes, $\Delta_{ji} = 1$; then, $\Delta_{ji} = 0$. w_i signifies the weight of i -th based organization perfect.

$$Acc = \frac{\sum_k \{1|c_k \text{ is the true class of instance } k\}}{\text{Size of test instances}} \times 100\%. \quad (2)$$

3.2.1 VGG-19 Model

Recently, many standard deep learning modules, like VGGNet, AlexNet, Inception, ResNet, Xception, DenseNet, and so on, have been utilized to detect plant disease. VGG network comprises multiple convolutional and pooling layers with various filter counts [23]. VGGNet consists of VGG16 and VGG19 modules. VGG16 has sixteen convolutional and pooling layers with the FC layer. VGG19 comprises nineteen convolutional and pooling layers with FC. The parameter count produced in the VGGNet is 140 million. The VGGNet is pre-trained on a larger dataset (ImageNet) with 1000 classes.

In this study, we employed shallow VGG module that takes nine layers of VGG19 that consist of 2 max-pooling and 7 convolution layers. Because there were fewer parameters, VGG stacked smaller filters together rather than using bigger ones. Instead of using massive filters, VGG uses smaller, deeper filters. It now has the same effective receptive field as if there were just one 7×7 convolutional layer. The input size applied in the presented model is $256 \times 256 \times 3$ and afterward implementing the pooling and convolution operations the output size is $64 \times 64 \times 256$. We used a global average pooling layer instead of an FC layer, which lowers the parameter count and dropout layer. The dropout layer is essential in reducing the network's over-fitting issues.

3.2.2 DenseNet Model

DenseNet is a network with many connections. Dense Blocks and Transition Layers are the two components that make up this structure. All levels in this network are directly connected in a feed-forward fashion. Prior layers' feature maps are inputs for each layer. ReLU activation, 3 3 Convolution, and Batch Normalisation comprise a Dense Block. Between every two Dense Blocks are transition layers. These include Average Pooling, Batch Normalisation, and 1–1 Convolution. Each dense block's feature maps are the same size for concatenation.

DenseNet is CNN, where every layer in the network is interconnected with each successive layer in a feedforward manner [24]. The input layer is a concatenation of feature map of each preceding layer. Specifically, for a network with L layer. For l^{th} layers in the network, it encompasses l input that comprises of concatenation of each preceding layer feature map and passes the resulting feature maps to the succeeding $L - 1$ layer. Therefore, the network has an overall of $\frac{L \times (L + 1)}{2}$ links among the layers. As a result, the network needs fewer limitations than traditional CNN because duplicate feature maps do not need to be re-learned. Densely connected patterns also minimize the effects of gradient vanishing issues in training deep architecture since every layer has direct access to the rise from the original input signal and the loss function, which enhances the flow of gradient and data over the network.

A DenseNet comprises L layers in which every implement a nonlinear conversion H_l whereby l indicates the index of layer, H_l represent a compound purpose of rectified linear unit (ReLU), a batch normalization (BN), and a convolution (Conv). The densely connected input of layer l is formulated as follows.

$$x_l = H_l([x_0 x_1, \dots, x_{x-1}]) \quad (3)$$

In Eq. (3), x_l indicates the input of layer and $[x_0 x, \dots, x_{l-1}]$ refers to the concatenation of feature map from 0 to $l-1$ layer. Transition block (TB) and dense block (DB) are the study 2 major components of the network. A dense block comprises several dense layers (DL) whereby every dense layer comprises 3×3 and 1×1 convolutinal layers. Concerning the dense block, there exist transition block that comprises 1×1 convolutinal and 2×2 average pooling and normalization layers. The DenseNet design is implemented as DenseNet121, which has 4 dense blocks, each of which has 6, 12, 24, and 16 dense layers.

3.2.3 EfficientNet Model

The EfficientNet family's EfficientNetB0 baseline network is recommended to have the lowest memory requirements while having the most excellent accuracy and FLOPS requirements feasible [25–27]. The network's width and depth (number of channels and layers) can be changed, or the picture

resolution can be raised so that the network can learn fine-grained features from the image. However, as this method includes dimensioning gain as scaling rises, multiple manual adjustments should be made to obtain an optimal match for each model. The EfficientNet family has emerged as a solution to the problem using the compound scaling strategy that the researcher provided. The method usages a multiple constant to uniform scaling network resolution, depth, and width.

$$d = \alpha, w = \beta, r = \gamma$$

$$s.t. \alpha \times \beta^2 \approx 2, \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \quad (4)$$

In Eq. (4), the network depth is signified by the period, the network width is signified as w , and the resolution can be signified as r . α, β, γ are constant that is calculated by the smaller grid search. The researcher found the optimal values α, β, γ to be $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$ on fixed values of $\Phi = 1$. In addition, the network structure makes use of somewhat large mobile inverted bottleneck convolution (MBCConv).

3.3 Hyperparameter Tuning

This study uses the MIWO technique to change the DL model hyperparameters in the best possible way. The IWO algorithm is a population-based optimization technique that finds a mathematical function by mimicking the randomness and compatibility of weed colonies [28–30]. Weed is one of the powerful herbs that aggressive growth habit provides a significant risk to the crops. It proves to be adaptable and very resistant to changes in the environment. As a result, considering its features, an effective optimization technique is accomplished. This approach tries to imitate a weed community's randomness, resistance, and adaptability. This technique is based on the agricultural phenomenon named colonies of IWs. A plant is called a weed if it grows unintentionally in an area and obstructs human needs and activities, even though it may have other uses and benefits in some places. Mehrabian and Lucas 2006, developed a mathematical optimization algorithm inspired by the colonized weed named the "Invasive Weed Optimization (IWO) Algorithm." Using essential weed colony characteristics, including growth, competition, and seeding, this method is efficient and straightforward in its convergence to the best solution [31–33].

To imitate the habitat behaviors of weeds, some key characteristics of the procedure are given in the following:

Initial populace: A limited number of seeds are discrete in the search space.

Reproduction: Each seed grows into a flowering plant that then releases another source, depending on the fitness levels. The number of grains of grass linearly reduces from S_{max} to S_{min} in the following:

$$n(w_i) = \frac{S_{max} (\max fit - fit(w_i)) + S_{min} (fit(w_i) - \min fit)}{\max fit - \min fit} \quad (5)$$

Spectral Spread: The seed generated by the group in the standard distribution with a standard deviation (SD) and mean planting location is generated using the following formula:

$$\sigma_r = \left(\frac{T-f}{T} \right)^n (\sigma_{final} - \sigma_{final}) + \sigma_{final} \quad (6)$$

In Eq. (6), the number of maximum iterations can be represented as T , the current SD is denoted as σ_r , the non-linear modulation index is signified by n .

This guarantees that the fall of grain in the variety reduces non-linearly at every stage, which eliminates unfitting plants and results in more appropriate plants and presentations of the transfer mode from to selection of K . Fig. 2 demonstrates the flowchart of IWO.

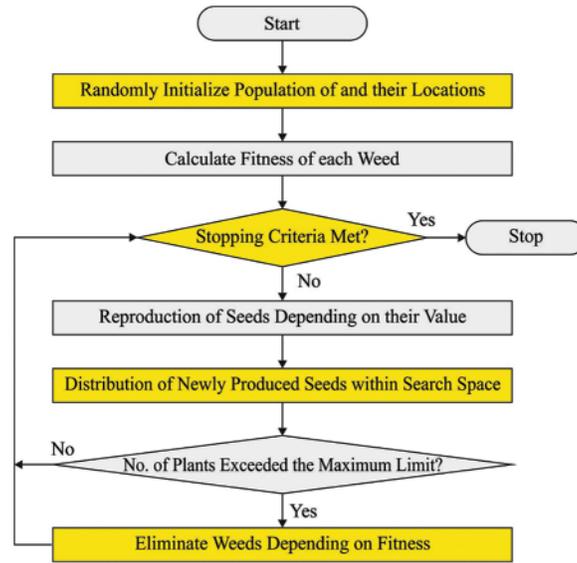


Figure 2: Flowcharts of IWO

Competitive deprivation: The herb population remains constant by removing the grass with the lowest fitness when the number of grasses in the colony reaches its maximum (P_{max}). This process is done as often as necessary, and then the grass's most minor colony cost function is saved. The oppositional-based learning (OBL) notion is used to compute the MIWO algorithm [34–36].

To improve the convergence speed and the possibility of defining the global optimum of the typical IWO technique [37–39], this work presents an improved version of the proposed method named the MIWO approach. During the primary iteration of the technique [40–42], afterward creating the immediate random solutions (that is, rats' positions), the opposite places of all the answers are made dependent upon the model of the opposite number [43,44]. To describe the initialized novel population, it is necessary to determine the model of opposite numbers. Assume that N -dimension vector X is as follows:

$$X = (x_1, x_2, \dots, x_N) \quad (7)$$

whereas $x_i \in [x_{min}, x_{max}]$. Afterward the opposite point of x_i that represented by \bar{x}_i , is expressed as:

$$\bar{x}_i = (x_{i_{max}} + x_{i_{min}}) - x_i, i = 1, 2, \dots, N \quad (8)$$

For applying the model of opposite numbers from the population initialized of AO technique, assume x_i that an arbitrarily created solution from N dimension problems space (that is, candidate solution). To this arbitrary solution, their opposite is created utilizing Eq. (8) and expressed by \bar{x}_i . Next, both solutions (viz., x_j and \bar{x}_i) are estimated by the objective function f . Thus, if $f(\bar{x}_i)$ is superior to $f(x_i)$ (that is, $f(\bar{x}_i) < f(x_i)$), the agent x_i is exchanged by \bar{x}_i ; else, it continues with x_i . So, during the primary iteration, a primary solution and its opposite were estimated concurrently to continue with superior (fitter) beginning agents.

The MIWO system generates an FF to enhance classifier presentation. It resolves a positive integer to indicate the higher version of the applicant outcomes. This study assumed that FF was provided in Eq. (9), which would reduce the classification error rate.

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{numberofmisclassifiedsamples}{Totalnumberofsamples} * 100 \quad (9)$$

Algorithm 1: Pseudocode of IWO Algorithm

```

create arbitrary population of  $N_0$  solutions
for iter = 1: maximal amount of generations
    calculate the maximal and minimal fitness from the colony
    for all the individuals,  $w \in W$ 
        calculate the number of samples for  $w$  based on their fitness
        randomly distribute created sample on the searching space with standard distribution around
the parent sample  $w$ 
        the created sample is added to solution set,  $W$ 
    if ( $|W| = N$ ) >  $p_{max}$ 
        sorted the population.  $W$  in descending order of its fitness
        truncate population of weeds with lesser fitness until  $N = p_{max}$ 
next iter

```

3.4 Softmax Classification

The feature extraction map is used to learn mappings between output classes and features that typically involve the FC layer and have each input from one layer connected with each activation unit of the succeeding layers. They are used in the network tail's classification function, where a layer with many neurons matching several output classes is used. In a multiclass classification problem, the *softmax* the function is applied to normalize the output into decimal values for the input probability. The class with maximum likelihood is represented as a target predictive class for the input. For organization problems with K class, the *SoftMax* purpose is arithmetically expressed as

$$\sigma(Z)_i = \frac{e^{z_i}}{\sum_{i=1}^k e^{z_i}} \quad (10)$$

In Eq. (10), i indicates several classes from one to K , x_i denotes the i^{th} dimensional output, and $\sigma(Z)_i$ characterizes the probability of input as i^{th} class.

4 Results and Discussion

Three publicly available datasets, VGG-19, EfficientNet, and DenseNet, are used in this study to assess and associate the performance of the proposed framework with other existing methodologies. Pytorch was used to create the recommended network, and an NVIDIA Titan Xp GPU was used to train it. CUDA Toolkit 10.2, Pytorch 1.4, Anaconda 3, and Ubuntu 16.04 (64-bit) were all used in the test setup.

The experimental validation of the IFBC-MIWOF technique is examined through several simulations. The dataset is represented in [Table 1](#) as 5000 samples with five class labels.

Table 1: Details of the dataset

Class	No. of key frames
Feeding (Fed)	1000
Hypoxia (Hyp)	1000
Hypothermia (Hypot)	1000
Frightening (Frg)	1000
Normal (Nor)	1000
Total number of key frames	5000

The confusion matrices for classifying fish behavior produced by the IFBC-MIWOF model are shown in [Fig. 3](#). The IFBC-MIWOF approach identified 788 samples under the FED class, 734 samples under the HYP class, 758 samples under the HYPOT class, 770 examples under the FRG class, and 771 examples under the NOR class with 80% of the TR dataset. Additionally, the IFBC-MIWOF approach identified 679 samples under the FED class, 663 samples under the HYP class, 657 samples under the HYPOT class, and 681 samples under the FRG class 691 examples under the NOR class with 70% of the TR data. The IFBC-MIWOF approach has also classified 290 examples under the FED class, 292 models under the HYP class, 281 samples under the HYPOT class, 295 under the FRG class, and 276 examples under the NOR class using only 30% of the TS data.

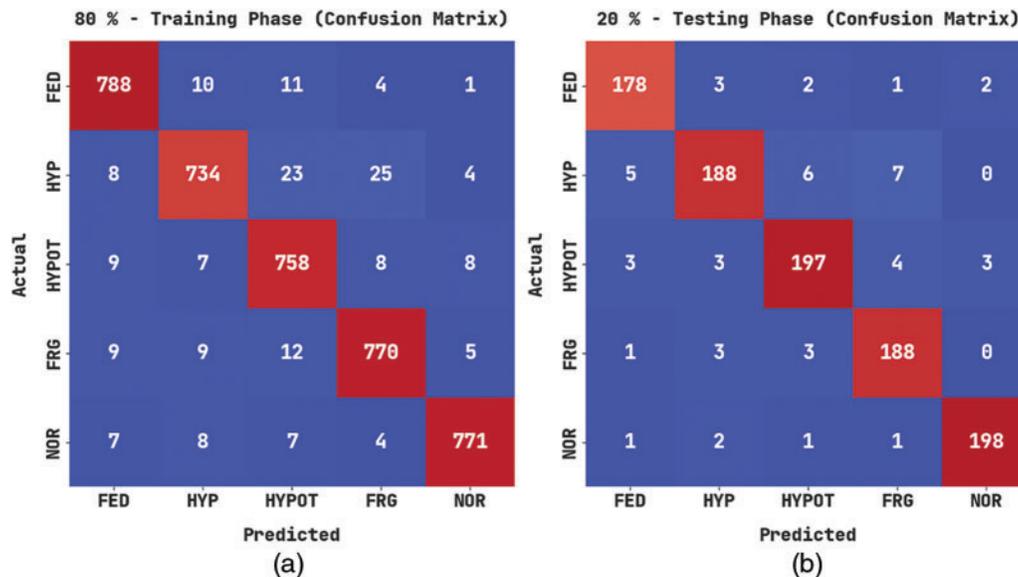


Figure 3: (Continued)

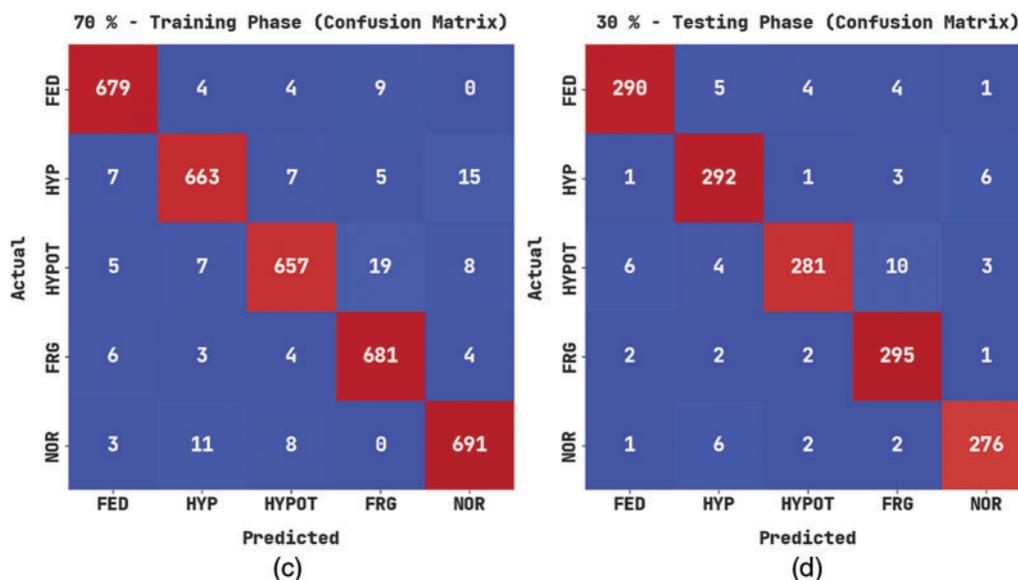


Figure 3: Confusion matrices of IFBC-MIWOEF approach (a) 80% of TR data, (b) 20% of TS data, (c) 70% of TR data, and (d) 30% of TS data

The consequences of the IFBC-MIWOEF model using 80% TR and 20% TS data are briefly presented in Table 2. On 80% of the TR dataset, Fig. 4 shows the general organization result of the IFBC-MIWOEF model. According to the figure, the IFBC-MIWOEF approach has shown improved outcomes across the board. For illustration, the IFBC-MIWOEF method has categorized examples under the FED class with $accu_y$ of 98.52%, $prec_n$ of 95.98%, $reca_l$ of 96.81%, $spec_y$ of 98.96%, and F_{score} of 96.39%. Meanwhile, the IFBC-MIWOEF technique has classified samples under HYP class with $accu_y$ of 97.65%, $prec_n$ of 95.57%, $reca_l$ of 92.44%, $spec_y$ of 98.94%, and F_{score} of 93.98%. Eventually, the IFBC-MIWOEF approach has classified samples under NOR class with $accu_y$ of 98.90%, $prec_n$ of 97.72%, $reca_l$ of 96.74%, $spec_y$ of 99.44%, and F_{score} of 97.23%.

Table 2: Result analysis of IFBC-MIWOEF methodology with dissimilar class labels under 80:20 of TR and TS dataset

Labels	Accuracy	Precision	Recall	Specificity	F-score
Training phase (80%)					
FED	98.52	95.98	96.81	98.96	96.39
HYP	97.65	95.57	92.44	98.94	93.98
HYPOT	97.88	93.46	95.95	98.35	94.69
FRG	98.10	94.94	95.65	98.72	95.30
NOR	98.90	97.72	96.74	99.44	97.23
Average	98.21	95.54	95.52	98.88	95.52
Testing phase (20%)					
FED	98.20	94.68	95.70	98.77	95.19
HYP	97.10	94.47	91.26	98.61	92.84

(Continued)

Table 2 (continued)

Labels	Accuracy	Precision	Recall	Specificity	F-score
HYPOT	97.50	94.26	93.81	98.48	94.03
FRG	98.00	93.53	96.41	98.39	94.95
NOR	99.00	97.54	97.54	99.37	97.54
Average	97.96	94.90	94.94	98.72	94.91

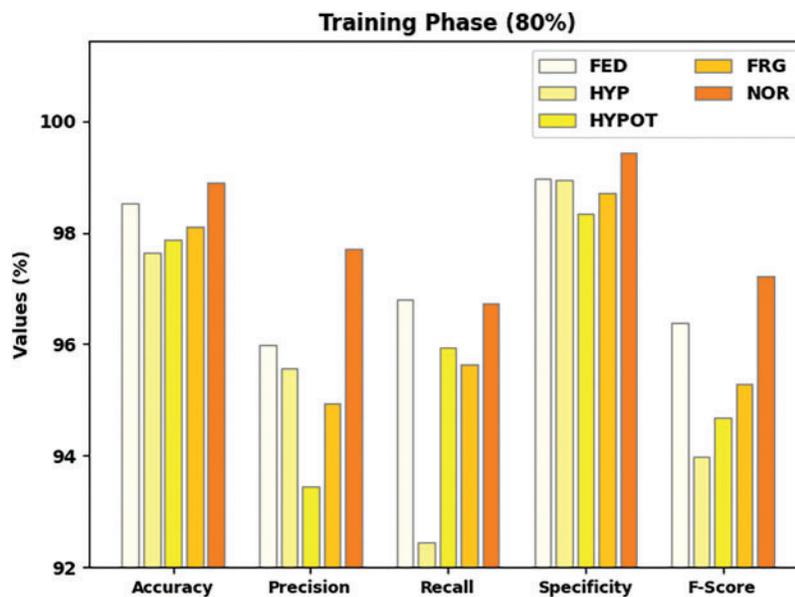
**Figure 4:** Result analysis of IFBC-MIWOEF method under 80% of TR data

Fig. 5 depicts the total organization consequences of the IFBC-MIWOEF technique on 20% of TS data. The figure indicates that the IFBC-MIWOEF technique has demonstrated improved outcomes under each class label. For example, the IFBC-MIWOEF method has classified samples under the FED class with $accu_y$ of 98.20%, $prec_n$ of 96.68%, $reca_l$ of 95.70%, $spec_y$ of 98.77%, and F_{score} of 95.19%. Meanwhile, the IFBC-MIWOEF method has classified samples under HYP class with $accu_y$ of 97.10%, $prec_n$ of 94.47%, $reca_l$ of 91.26%, $spec_y$ of 98.61%, and F_{score} of 92.84%. Eventually, the IFBC-MIWOEF model has classified samples under NOR class with $accu_y$ of 99%, $prec_n$ of 97.54%, $reca_l$ of 97.54%, $spec_y$ of 99.37%, and F_{score} of 97.54%.

A brief fish behavior categorization result using the IFBC-MIWOEF methodology under the 70% TR and 30% TS dataset is provided in Table 3. The overall classification results of the IFBC-MIWOEF technique are shown in Fig. 6 using data from 70% of the TR dataset. According to the figure, the IFBC-MIWOEF technique has improved results for each class label. For example, the IFBC-MIWOEF technique has classified samples under the FED class with $accu_y$ of 98.91%, $prec_n$ of 97%, $reca_l$ of 97.56%, $spec_y$ of 99.25%, and F_{score} of 97.28%. Meanwhile, the IFBC-MIWOEF technique has classified samples under HYP class with $accu_y$ of 98.31%, $prec_n$ of 96.37%, $reca_l$ of 95.12%, $spec_y$ of 99.11%, and F_{score} of 95.74%. Eventually, the IFBC-MIWOEF organization has classified examples

under NOR class with $accu_y$ of 98.60%, $prec_n$ of 96.24%, $reca_l$ of 96.91%, $spec_y$ of 99.03%, and F_{score} of 96.58%.

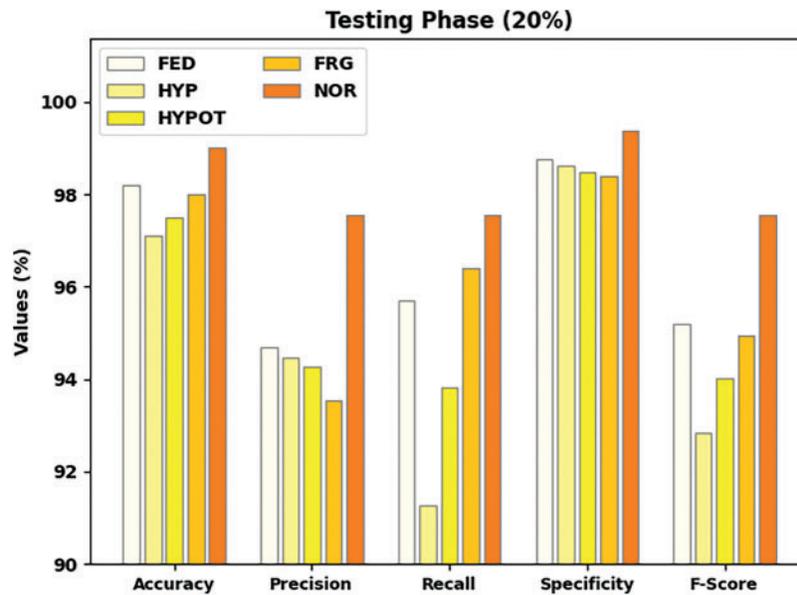


Figure 5: Result analysis of IFBC-MIWOF method under 20% of TS data

Table 3: Result analysis of IFBC-MIWOF method with distinct class labels under 70:30 of data

Labels	Accuracy	Precision	Recall	Specificity	F-score
Training phase (70%)					
FED	98.91	97.00	97.56	99.25	97.28
HYP	98.31	96.37	95.12	99.11	95.74
HYPOT	98.23	96.62	94.40	99.18	95.49
FRG	98.57	95.38	97.56	98.82	96.46
NOR	98.60	96.24	96.91	99.03	96.58
Average	98.53	96.32	96.31	99.08	96.31
Testing phase (30%)					
FED	98.40	96.67	95.39	99.16	96.03
HYP	98.13	94.50	96.37	98.58	95.42
HYPOT	97.87	96.90	92.43	99.25	94.61
FRG	98.27	93.95	97.68	98.41	95.78
NOR	98.53	96.17	96.17	99.09	96.17
Average	98.24	95.64	95.61	98.90	95.60

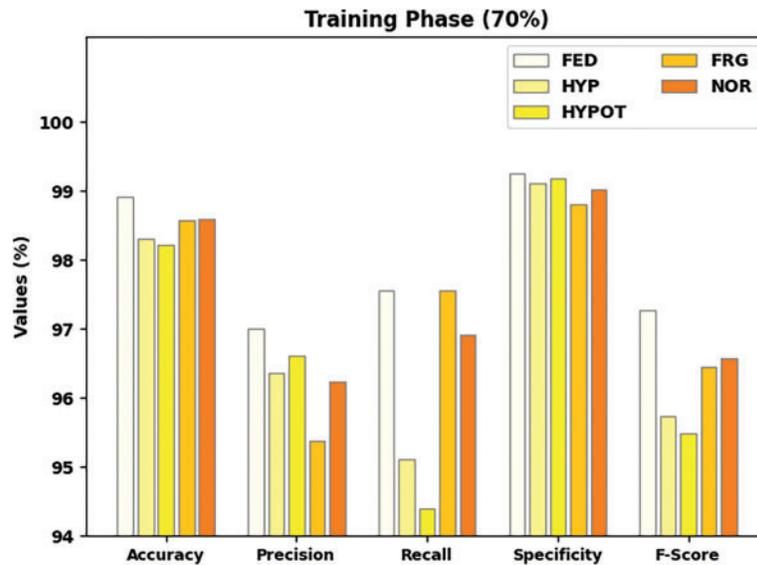


Figure 6: Result analysis of IFBC-MIWOEF method under 70% of TR data

Fig. 7 demonstrates the total organization consequences of the IFBC-MIWOEF method on 30% of the TS dataset. The figure shows that the IFBC-MIWOEF method has demonstrated improved outcomes under each class label. For example, the IFBC-MIWOEF method has classified samples under the FED class with $accu_y$ of 98.40%, $prec_n$ of 96.67%, $reca_l$ of 95.39%, $spec_y$ of 99.16%, and F_{score} of 96.03%. Meanwhile, the IFBC-MIWOEF system has classified samples under HYP class with $accu_y$ of 98.13%, $prec_n$ of 94.50%, $reca_l$ of 96.37%, $spec_y$ of 98.58%, and F_{score} of 95.42%. Eventually, the IFBC-MIWOEF method has classified samples under NOR class with $accu_y$ of 98.53%, $prec_n$ of 96.17%, $reca_l$ of 96.17%, $spec_y$ of 99.09%, and F_{score} of 96.17%.

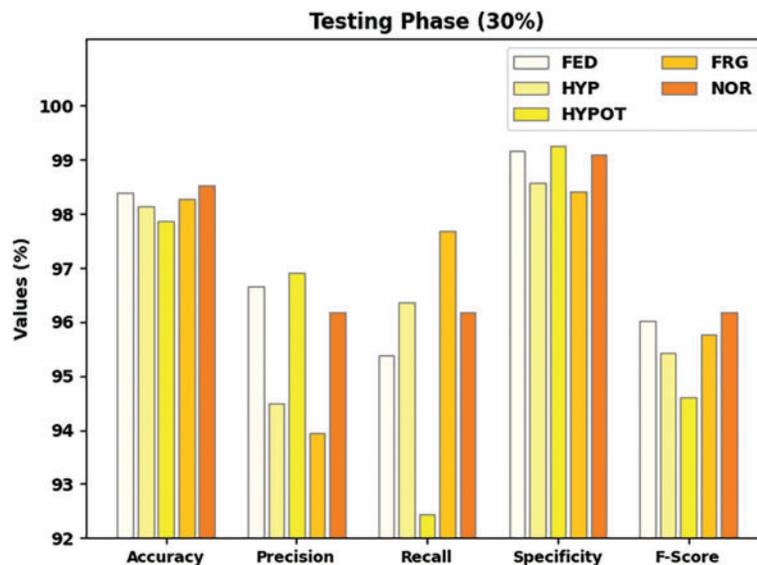


Figure 7: Result analysis of IFBC-MIWOEF method under 30% of TS data

Table 4 demonstrates the comparison study of the IFBC-MIWOEF model with other models. Fig. 8 reports a comparative review of the IFBC-MIWOEF with existing methodologies. The graph showed that the LeNet5 model performed poorly, with an $accu_y$ of only 87.15%. Followed by, the Bi-LSTM, 3D-residual networks, CNN-LSTM, and GRU models have reported slightly improved $accu_y$ of 93.14%, 90.07%, 93.95%, and 92.67% correspondingly. Along with that, the fusion OFD-DL model has proficient reasonable $accu_y$ of 96.18%. However, the IFBC-MIWOEF model has shown maximum performance with $accu_y$ of 98.53%.

Table 4: Comparative analysis of IFBC-MIWOEF approach with existing methodologies

Methods	Accuracy	Computational time (sec)
IFBC-MIWOEF	98.53	6.51
LeNet5	87.15	7.34
Bi-LSTM	93.14	11.90
3D residual networks	90.07	9.42
CNN-LSTM	93.95	7.58
GRU	92.67	7.46
Fusion OFD-DL	96.18	12.59

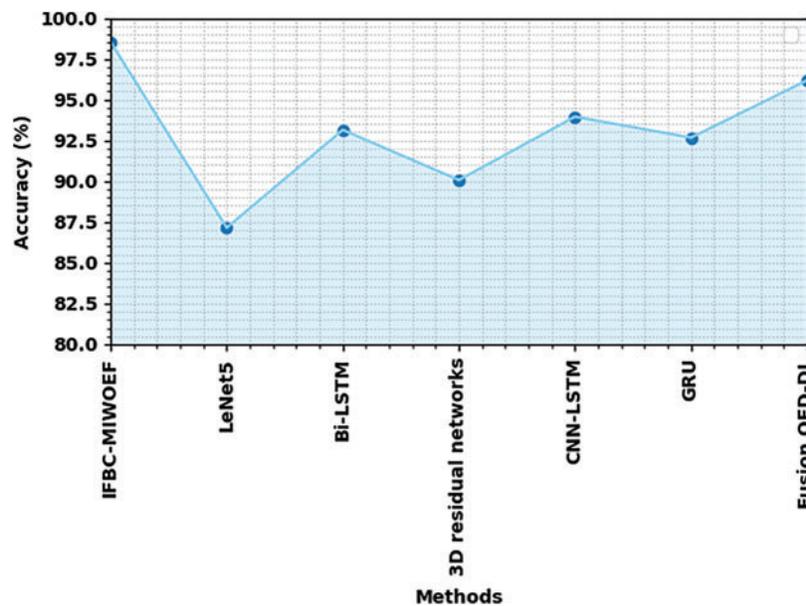


Figure 8: $Accu_y$ analysis of IFBC-MIWOEF approach with existing methods

The experimental results proved that the IFBC-MIWOEF model performed better than other models at recognising fish behaviour.

5 Conclusion

To identify and categorize different forms of fish behavior, a novel IFBC-MIWOEF model was developed in this study. Primarily, the input videos are collected, and the frame conversion process takes place. Then, the fusion-based feature extraction process is carried out through three DL algorithms, namely VGG-19, DenseNet, and EfficientNet. The performance of the DL models can be improvised using the MIWO algorithm-based hyperparameter tuning process. Finally, the SM layer assigns the incoming video frames appropriate class labels. To demonstrate the efficacy of the IFBC-MIWOEF model, a series of imitations were performed using distinct processes. An Extensive comparative study pointed out the better outcomes of the IFBC-MIWOEF model over current methods. Thus, the presented IFBC-MIWOEF approach can be treated as a professional fish behavior recognition and classification tool.

Funding Statement: The authors received no specific funding for this study.

Availability of Data and Materials: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. Zhao, S. Zhang, J. Liu, H. Wang, J. Zhu *et al.*, “Application of machine learning in intelligent fish aquaculture: A review,” *Aquaculture*, vol. 540, pp. 736724, 2021.
- [2] M. Sun, X. Yang and Y. Xie, “Deep learning in aquaculture: A review,” *Journal of Computers*, vol. 31, no. 1, pp. 294–319, 2020.
- [3] C. Xia, L. Fu, Z. Liu, H. Liu, L. Chen *et al.*, “Aquatic toxic analysis by monitoring fish behavior using computer vision: A recent progress,” *Journal of Toxicology*, vol. 2018, no. 2, pp. 1–15, 2018.
- [4] M. Susmita and P. Mohan, “Digital mammogram inferencing system using intuitionistic fuzzy theory,” *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 1099–1115, 2022.
- [5] B. V. Deep and R. Dash, “Underwater fish species recognition using deep learning techniques,” in *6th Int. Conf. on Signal Processing and Integrated Networks (SPIN)*, Noida, India, pp. 665–669, 2019.
- [6] L. Yang, Y. Liu, H. Yu, X. Fang, L. Song *et al.*, “Computer vision models in intelligent aquaculture with emphasis on fish detection and behavior analysis: A review,” *Archives of Computational Methods in Engineering*, vol. 28, no. 4, pp. 2785–2816, 2021.
- [7] W. Xu and S. Matzner, “Underwater fish detection using deep learning for water power applications,” in *Int. Conf. on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, pp. 313–318, 2018.
- [8] D. Li, Z. Wang, S. Wu, Z. Miao, L. Du *et al.*, “Automatic recognition methods of fish feeding behavior in aquaculture: A review,” *Aquaculture*, vol. 528, pp. 735508, 2020.
- [9] A. Banan, A. Nasiri and A. Taheri-Garavand, “Deep learning-based appearance features extraction for automated carp species identification,” *Aquacultural Engineering*, vol. 89, pp. 102053, 2022.
- [10] B. Jaishankar, A. Pundir, I. Patel and N. Arulkumar, “Blockchain for securing healthcare data using squirrel search optimization algorithm,” *Intelligent Automation & Soft Computing*, vol. 32, no. 3, pp. 1815–1829, 2022.
- [11] M. A. Iqbal, Z. Wang, Z. A. Ali and S. Riaz, “Automatic fish species classification using deep convolutional neural networks,” *Wireless Personal Communications*, vol. 116, no. 2, pp. 1043–1053, 2021.

- [12] A. Jalal, A. Salman, A. Mian and F. Shafait, "Fish detection and species classification in underwater environments using deep learning with temporal information," *Ecological Informatics*, vol. 57, pp. 101088, 2020.
- [13] A. B. Tamou, A. Benzinou, K. Nasreddine and L. Ballihi, "Underwater live fish recognition by deep learning," in *Int. Conf. on Image and Signal Processing*, Cherbourg, France, Springer, pp. 275–283, 2018.
- [14] H. S. Chhabra, A. K. Srivastava and R. Nijhawan, "A hybrid deep learning approach for automatic fish classification," in *Proc. of ICETIT 2019*, Institutional Area, Janakpuri New Delhi, Springer, pp. 427–436, 2020.
- [15] C. Zhou, D. Xu, L. Chen, S. Zhang, C. Sun *et al.*, "Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision," *Aquaculture*, vol. 507, pp. 457–465, 2019.
- [16] M. U. Farooq, M. N. M. Saad and S. D. Khan, "Motion-shape-based deep learning approach for divergence behavior detection in high-density crowd," *Visual Computer*, vol. 38, pp. 1553–1577, 2022.
- [17] S. D. Khan, Y. Salih, B. Zafar and A. Noorwali, "A deep-fusion network for crowd counting in high-density crowded scenes," *International Journal Computer Intelligence System*, vol. 14, pp. 168, 2021.
- [18] A. J. Alzahrani and S. D. Khan, "Characterization of different crowd behaviors using novel deep learning framework," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 29, pp. 169–185, 2021.
- [19] S. D. Khan, S. Bandini, S. Basalamah and G. Vizzari, "Analysing crowd behavior in naturalistic conditions: Identifying sources and sinks and characterizing main flows," *Neurocomputing*, vol. 177, no. 5, pp. 543–563, 2016.
- [20] K. M. Knausgård, A. Wiklund, T. K. Sjørdalen, K. T. Halvorsen, A. R. Kleiven *et al.*, "Temperate fish detection and classification: A deep learning-based approach," *Applied Intelligence*, vol. 52, no. 6, pp. 6988–7001, 2020.
- [21] M. Revanesh, V. Sridhar and J. M. Acken, "CB-ALCA: A cluster-based adaptive lightweight cryptographic algorithm for secure routing in wireless sensor networks," *International Journal of Information and Computer Security*, vol. 11, no. 6, pp. 637–662, 2019.
- [22] G. Wang, A. Muhammad, C. Liu, L. Du and D. Li, "Automatic recognition of fish behavior with a fusion of RGB and optical flow data based on deep learning," *Animals*, vol. 11, no. 10, pp. 2774, 2021.
- [23] M. R. Hassan, M. F. Islam, M. Z. Uddin, G. Ghoshal, M. M. Hassan *et al.*, "Prostate cancer classification from ultrasound and MRI images using deep learning based Explainable Artificial Intelligence," *Future Generation Computer Systems*, vol. 127, pp. 462–472, 2022.
- [24] C. Saravanakumar, R. Priscilla, B. Prabha, A. Kavitha and C. Arun, "An efficient on-demand virtual machine migration in cloud using common de-ployment model," *Computer Systems Science and Engineering*, vol. 42, no. 1, pp. 245–256, 2022.
- [25] R. R. Bhukya, B. M. Hardas and T. Ch, "An automated word embedding with parameter tuned model for web crawling," *Intelligent Automation & Soft Computing*, vol. 32, no. 3, pp. 1617–1632, 2022.
- [26] C. Pretty Diana Cyril, J. Rene Beulah, A. Harshavardhan and D. Sivabalaselvamani, "An automated learning model for sentiment analysis and data classification of Twitter data using balanced CA-SVM," *Concurrent Engineering: Research and Applications*, vol. 29, no. 4, pp. 386–395, 2021.
- [27] P. V. Rajaram and P. Mohan, "Intelligent deep learning based bidirectional long short term memory model for automated reply of e-mail client prototype," *Pattern Recognition Letters*, vol. 152, no. 10, pp. 340–347, 2021.
- [28] V. Harshavardhan and S. Velmurugan, "Chaotic salp swarm optimization-based energy-aware VMP technique for cloud data centers," *Computational Intelligence and Neuroscience*, vol. 2022, no. 4, pp. 1–16, 2022.
- [29] J. Deepak Kumar, B. Prasanthi and J. Venkatesh, "An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification," *Information Processing & Management*, vol. 59, no. 1, pp. 1–15, 2022.
- [30] D. K. Jain, S. K. Sah Tyagi and N. Subramani, "Metaheuristic optimization-based resource allocation technique for cybertwin-driven 6G on IoE environment," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4884–4892, 2022.

- [31] B. M. Hardas and S. B. Pokle, "Optimization of peak to average power reduction in OFDM," *Journal of Communications Technology and Electronics*, vol. 62, pp. 1388–1395, 2017.
- [32] Y. Alotaibi, S. Alghamdi and O. I. Khalaf, "An efficient metaheuristic-based clustering with routing protocol for underwater wireless sensor networks," *Sensors*, vol. 22, no. 2, pp. 1–15, 2022.
- [33] T. Satish Kumar, S. Jothilakshmi, B. C. James and N. Arulkumar, "HHO-based vector quantization technique for biomedical image compression in cloud computing," *International Journal of Image and Graphics*, vol. 23, no. 3, pp. 1–15, 2022.
- [34] H. Y. Sang, Q. K. Pan, J. Q. Li, P. Wang and Y. Y. Han, "Effective invasive weed optimization algorithms for distributed assembly permutation flow shop problem with total flowtime criterion," *Swarm and Evolutionary Computation*, vol. 44, pp. 64–73, 2019.
- [35] Y. Alotaibi, S. Alghamdi, O. I. Khalaf and U. Sakthi, "Improved metaheuristics-based clustering with multihop routing protocol for underwater wireless sensor networks," *Sensors*, vol. 22, no. 4, pp. 1–16, 2022.
- [36] D. Paulraj and P. Ezhumalai, "A deep learning modified neural network (DLMNN) based proficient sentiment analysis technique on twitter data," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–15, 2022. <https://doi.org/10.1080/0952813X.2022.2093405>
- [37] B. T. Geetha, A. K. Nanda, A. M. Metwally, M. Santhamoorthy and M. S. Gupta, "Metaheuristics with deep transfer learning enabled detection and classification model for industrial waste management," *Chemosphere*, vol. 308, no. 10, pp. 1–13, 2022.
- [38] S. Raghavendra, B. T. Geetha, S. M. R. Asha and M. K. Roberts, "Artificial humming bird with data science enabled stability prediction model for smart grids," *Sustainable Computing: Informatics and Systems*, vol. 36, no. 9, pp. 1–17, 2022.
- [39] D. K. Jain and X. Liu, "Modeling of human action recognition using hyperparameter tuned deep learning model," *Journal of Electronic Imaging*, vol. 32, no. 1, pp. 1–10, 2022.
- [40] E. Veerappampalayam, N. Subramani, M. Subramanian and S. Meckanzi, "Handcrafted deep-feature-based brain tumor detection and classification using MRI images," *Electronics*, vol. 11, no. 24, pp. 1–10, 2022.
- [41] A. Mardani, A. R. Mishra and P. Ezhumalai, "A fuzzy logic and DEEC protocol-based clustering routing method for wireless sensor networks," *AIMS Mathematics*, vol. 8, no. 4, pp. 8310–8331, 2023.
- [42] V. D. Ambeth Kumar, S. Malathi, K. Abhishek Kumar and K. C. Veluvolu, "Active volume control in smart phones based on user activity and ambient noise," *Sensors*, vol. 20, no. 15, pp. 1–15, 2020.
- [43] C. Ramalingam, "An efficient applications cloud interoperability framework using I-Anfis," *Symmetry*, vol. 13, no. 2, pp. 1–15, 2021.
- [44] R. Chithambaramani, "Addressing semantics standards for cloud portability and interoperability in multi cloud environment," *Symmetry*, vol. 13, no. 2, pp. 1–14, 2021.