Received: 10 March 2024 | *Revised: 13 July 2024* | *Accepted: 19 July 2024*
DOI: 10.37917/ijeee.22.1.32 *Vol. 22 | Issue 1 | June 2026*

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Enhancing Underwater Search and Rescue Operations: A CNN Approach for Human, Fish, and Plant Classification

Ali Fathel Rasheed*[1], Rabee M. Hagem[2], Abdul Sattar Mohammed Khidhir[3]
[1]Mathematics Department, College of Basic Education, University of Telafer, Mosul, Iraq
[2]Computer Engineering Department, College of Engineering, University of Mosul Mosul, Iraq
[3]Electronics Technology Department, Mosul Technical Institute, Northern Technical University, Mosul, Iraq

Correspondance
*Ali Fathel Rasheed
Mathematics Department, College of Basic Education,
University of Telafer, Mosul, Iraq
Email: alifadilrasheed@gmail.com.

Abstract
In recent times, artificial intelligence has become an essential part of our lives, particularly in tasks involving object recognition. This paper explores the use of convolutional neural networks (CNNs) for enhancing underwater search and rescue operations by classifying images of humans, fish, and plants. Leveraging the OpenCV library for preprocessing and the Keras library with a TensorFlow backend for recognition, this study utilizes a dataset captured through field experiments. The methodology involved preprocessing the images for segmentation, followed by training a CNN model to classify these images with high accuracy. The CNN model demonstrated a remarkable classification accuracy of 99.6 %, significantly outperforming other modern machine-learning methods. This work suggests that CNNs can greatly improve the speed and effectiveness of underwater search and rescue operations by accurately identifying and locating submerged persons, which is critical for timely rescue missions.

Keywords
Artificial Intelligence, Convolutional Neural Network CNN, OpenCV, Underwater Search and Rescue.

I. INTRODUCTION

Image processing significantly enhances operational efficiency by reducing workload and time requirements without compromising the classification accuracy of the objects in question [1,2]. The advent of deep learning, a sophisticated branch of machine learning, offers unparalleled capabilities for automatically monitoring and analyzing underwater digital data, presenting new opportunities for addressing complex underwater challenges [3]. Unlike traditional machine learning, which relies on human-classified images with pre-defined labels for model training, deep learning methods, particularly convolutional neural networks (CNNs), excel in directly extracting features from images. CNNs are at the forefront of deep learning techniques, known for their enhanced predictive capabilities and, in some applications, accuracy surpassing human performance [4]. The unique challenges of underwater object detection, such as water turbidity and limited lighting conditions, necessitate robust computer vision techniques. This paper proposes the use of CNNs to enhance the detection and classification of underwater objects, aiming to significantly contribute to rescue operations by locating drowned individuals more effectively. The choice of CNNs over other models is motivated by their proven strength in handling spatial hierarchies in images, which is critical in the dynamic and visually complex in underwater environment. A dataset comprising images categorized into humans, fish, and plants was collected using a webcam, and subsequently divided into training and testing sets. This structured approach not only tests the efficacy of CNNs in underwater conditions but also sets a foundational methodology for future researchers to extend and refine. This study aims to:

* Demonstrate the effectiveness of CNNs in the context

of underwater object classification, detailing the technical process and the model's performance compared to existing methods.

∗ Offer insights into why CNNs are preferable for this specific application, discussing their advantages over other deep learning models in handling the visual challenges presented by underwater environments.

∗ Provide a reproducible model and dataset that can serve as a basis for further research, encouraging advancements in the field of underwater image processing.

A. Real life commentary

The ability to detect and classify underwater objects has real-life important implications. For example, it can be used to search for people lost in water. Looking for bodies in the water soon after an accident is very important. However, extreme conditions on land, such as high currents, poor visibility, or formidable obstacles in the water do not guarantee the safety of rescuers If roads based on CNN can help detect and classify underwater objects, including humans, . plants and fish, safely and effectively [5, 6]. Moreover, despite the efforts made to enhance image resolution and collect extensive real-time data, the process remains tedious and often problematic. The challenge persists in creating a model capable of providing high accuracy and effectively detecting objects in images affected by distortion, low-light conditions, and segmentation errors [7, 8]. Additionally, existing models, despite employing various image processing and recognition approaches, tend to overlook the significant impact of each feature and adapt a limited number of features [9, 10]. Addressing these issues requires a system with feature variability. Several researchers have proposed diverse models for this problem, utilizing various data mining techniques within the realm of machine learning.

B. Related Work

This section discusses all known machine learning methods, especially those utilizing deep neural networks, in the analysis of marine digital data, image annotation, object detection, and classification. The methods are categorized based on the detection subject. Emphasis is also placed on the features and classifiers employed. In the field of machine learning, various techniques have been used to achieve high accuracy rates in different domains. For instance, Andrew Rova et al. [10] explored fish classification by pre-processing the images through a warping technique before classification. Utilizing Support Vector Machines (SVM) on a dataset consisting of 320 images, they achieved an accuracy rate of 90% [11]. Similarly, Raine et al. developed a multi-species seagrass detector and classifier using a deep convolutional neural network (CNN), achieving an impressive 92.4% accuracy. Their method also introduced a semi-automated process for labeling image patches, significantly reducing manual effort in underwater image analysis. The study's dataset, code, and pre-trained models were made publicly available to facilitate reproducibility [12]. Han et al. utilized Convolutional Neural Networks (CNN) to enhance and detect underwater images, achieving 90% mAP and 50 FPS in object detection. Their method combined max-RGB and shades of gray techniques for image enhancement, outperforming traditional models in real-time underwater robotics [13]. Rismiyati et al, successfully utilized transfer learning and Artificial Neural Network models to classify food/nonfood images, yielding impressive accuracy results of 95.83% [14]. IBRAHIM et al. explored the classification of tea leaf shoot maturity levels using two CNN architectures, namely VGGNET 19 and ResNet50. The VGGNET19 architecture demonstrated the highest accuracy value, reaching 97.5% in test results [15]. In addition, Kratzert et al., proposed a system for categorizing fish into distinct classes, utilizing Singular Value Decomposition (SVD) and feature extraction. The model is trained using Artificial Neural Networks (ANN), achieving an impressive accuracy of 94 percent on the testing dataset [16]. Salman et al., employed innovative techniques to develop a method for accurate fish detection in videos. Their approach involved training the model with regional CNN, followed by applying background subtraction and optical flow to raw images to identify fish movement regions. This method demonstrated accuracy rates of 87.44 percent and 80.02 percent on two different dataset repositories [17]. Almero et al. utilized an artificial neural network along with a feature selection algorithm to address the fish classification problem. They applied a classification tree algorithm for feature selection, and the selected features were input into a 100-hidden-layer ANN model, resulting in a testing accuracy of 78.0 percent [18]. Villon et al. researched accurately identifying fishes using Convolutional Neural Networks (CNN) and compared it to human identification. The results revealed that CNN achieved an accuracy of 94.9 percent, surpassing human identification accuracy, which stood at 84.9 percent [19]. It can be noted that all previous works, focused on the classification of objects in the water such as fish or plant types without addressing the subject of discovering drowned people. therefore this paper presents multiclass detection and classification of underwater objects.

II. Research Method

The research method used is experimental research, which is a systematic scientific approach to understanding cause-and-effect relationships between variables in a particular context

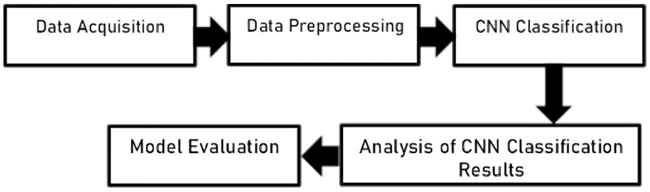

Fig. 1. Research stages.

A. Materials and Tools

The techniques used in this study included obtaining images of different fish, plants, and people through video recordings. Subsequently, a collection of photographic photographs turned into fashion. The information acquisition technique utilized an Acer 5742G computer prepared with an Intel Core i5 processor, 4 gigabytes of Random Access Memory (RAM), and a 500-gigabyte Solid State Drive (SSD). The research also employed OpenCV, Keras, and TensorFlow Python libraries within a Jupyter Notebook environment.

B. Time and Place of the Search

This research was conducted during the various periods from September to December 2023 AD in the computer laboratory of the University of Mosul campus. In September, fish and plant image data were collected from multiple locations in the city of Mosul, including the Tigris River and some visible water ponds. During this period, an innovative central processing process was performed, represented by changing its dimensions to 200 x 200 pixels. The blocking process uses the Python language, and the website was also used.https://www.kaggle.com/datasets/alirasheed2020/human-fish-and-plant/settings for the best trained Convolutional Network (CNN) model.

C. Search Process

The conducted research process included data acquisition, data preprocessing, classification using the Convolutional Neural Network (CNN) method, analysis of classification results, and model evaluation has been illustrated in Fig. 1

1) Data Acquisition:

The dataset consists of 6071 samples. Fig. 2 include some images of fish, plants, and humans Fig. 3. The dataset is divided into 1691 fish, 1448 plants, and 2932 humans. The images were collected from various locations in Mosul city, including the Tigris River, several water ponds with reduced visibility, and closed swimming pools. Please note that the data collection was done personally by the authors through videos.

2) Data Preprocessing:

The preprocessing stage is implemented to prepare the images before segmentation. In this stage, image resizing is carried out by proportionally changing its dimensions. Upon completing the image resizing, the image samples are divided into three classes: fish, plants, and humans 70 % of the data is allocated for training the model using the CNN method, while the remaining 30% is designated for testing the CNN model. This approach follows the CNN method.

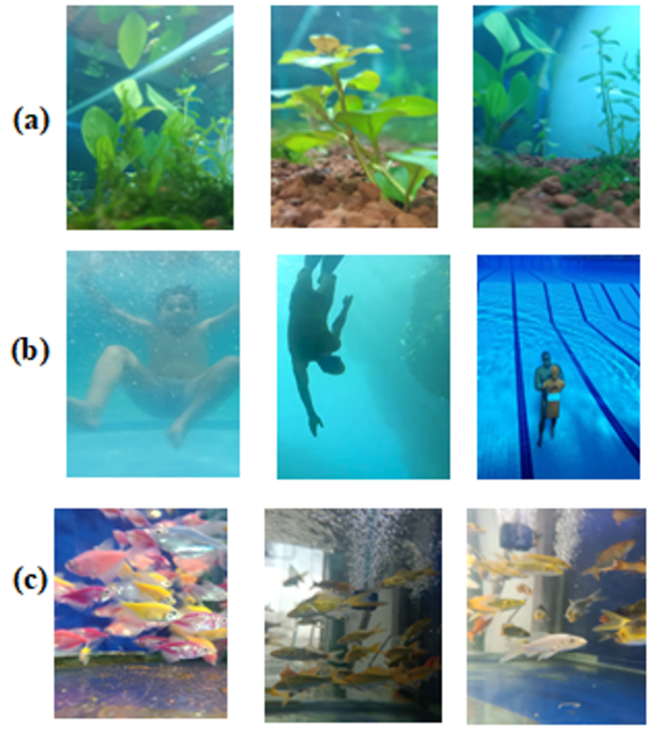

Fig. 2. Illustration of some underwater samples: (a)Plants, (b)Humans, and, (c) Fish.

3) Convolutional Neural Network:

The Convolutional Neural Network (CNN) is one type of architecture in artificial deep learning specifically designed for processing structured data such as images and videos [20]. CNN has become one of the most widely used techniques in the field of pattern recognition and com puter vision. (CNN), also known as ConvNet, is a deep learning approach composed of multiple layers. CNN boasts versatile applications, including segmentation [21], object detection [22], and face recognition [23]. The classification of CNN involves the modeling process of the CNN method Fig. 2. The CNN method consists of two steps. The first step involves training using the backpropagation method, which is one of the approaches constituting the CNN method. It aims to later train the data during the modeling process, specifically in this case, the image data of fish [24]. The second step involves image classification using the feedforward method. This approach is one of the techniques constituting the CNN method, which, during modeling, focuses on the classification of data. In this stage, the machine classifies fish image data, determining whether the image belongs to a specific genus or another type [25]. The first step entails configuring the optimizer type (utilizing stochastic gradient descent), specifying the number of epochs, and determining the batch size. Subsequently, the second step employs the feedforward method with updated weights and biases. In some instances, dropout parameters are incorporated during the model design to mitigate overfitting and underfitting issues. Following that, the images that have undergone the preprocessing stage will be tested on the created model. The image computation process is depicted by the following equation [26]:

$$E_{i,j} = \{1,\ldots,n_2\} \times \{1,\ldots,n_3\} \tag{1}$$

This is an image of an object n_1 which possesses features with dimensional size $n_1 \times n_2$. If the CONV layer, e, contains an input feature equal to the output of the previous layer, $n_1^{(e-1)}$, with each size $n_2^{(e-1)} \times n_3^{(e-1)}$, the output e is as large as n_1^e the feature with size $n_2^e \times n_3^e$. Feature (i) in layer e, W_1^e can be calculated using the following equation [26]:

$$W_1^e = M_1^e + \sum_{j=1}^{n_1^{(e-1)}} \left(D_i^e * W_j^{(e-1)} \right) \tag{2}$$

M_1^e is the bias of the matrix, D_i^e is the filter for the j th feature in the layer (e-1), and the $i-th$ feature in layer e, the filter size $(2g_1^{e-)}) \times (2g_2^{e-1})$, then D can be written in the following equation [26]:

$$D = \begin{bmatrix} D_{g_1-g_3} & \cdots & D_{g_1,g_3} \\ \vdots & K_{0,0} & \vdots \\ D_{g_1-g_3} & \cdots & D_{g_1,g_3} \end{bmatrix} \tag{3}$$

Calculating the output feature size of layer e can use the following equation [26]:

$$n_2^e = n_2^{e-1} - 2g_1^e \tag{4}$$

$$n_3^e = n_3^{e-1} - 2g_2^e \tag{5}$$

CNN employs a Rectified Linear Unit (ReLU) activation function to accelerate the convergence of the training process. This is achieved by amplifying the loss of the network through the utilization of the gain coefficient z_i. This approach simplifies the correlation, as expressed by the following equation [26]:

$$W_1^e = z_i f(W_i^{e-1}) \tag{6}$$

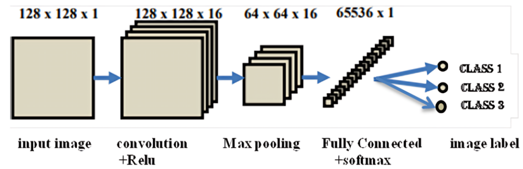

Fig. 3. Illustration of CNN architecture [26].

$$W_1^e = \max(0, W_i^{e-1}) \tag{7}$$

The input image undergoes convolution in the CONV layer using a specific kernel to generate multiple output features. The training process aims to optimize these features, ensuring the model can produce the most relevant features during the image classification process. The max-polling algorithm was added to the CNN architectural structure to reduce computational complexity [27]. The function of this algorithm is to improve translation invariance so that the size of the feature map can be reduced. After that, a normalization process is carried out so that calculations are simplified, robustness is increased, and errors are avoided when carrying out the weighting process, by following the following equation [26,27].

$$W_i^e = \frac{W_i^{e-1}}{c + \mu \sum_{j=1}^{n_1^{e-1}} \left(W_j^{e-1} \right)^2} \tag{8}$$

The Fully Connected layer is the final layer in CNN, responsible for classification based on the number of features generated in the preceding layer's calculations. The activation function in the *fully connected layer* is the *softmax* function [28]. The *softmax* function is employed to minimize the *error* value of the *cross-entropy*function, commonly used in solving classification problems. For instance, n_1^{e-1} in a fully connected layer e with a specific number of input features $W_1^e = f(p_i^e)$, p_i^e the calculation for the $i-th$ unit in layer $e$is determined by the following equation [26,28].

$$p_i^e = \sum_{j=1}^{n_1^{e-1}} \sum_{r=1}^{n_2^{e-1}} \sum_{s=1}^{n_3^{e-1}} O_{j,r,s}^e \left(W_j^{e-1} \right)_{r,s} \quad (9) \tag{9}$$

$O_{j,r,s}^e$ by showing the weight connecting the jth unit at position (r,s) in layer $(e-1)$ and the ith unit in layer e .

D. Proposed system

The proposed Convolutional Neural Network (CNN) model is crafted to identify individual photographs after the completion of the segmentation process and isolate each image The architecture of the proposed CNN model is illustrated in Fig. 4.

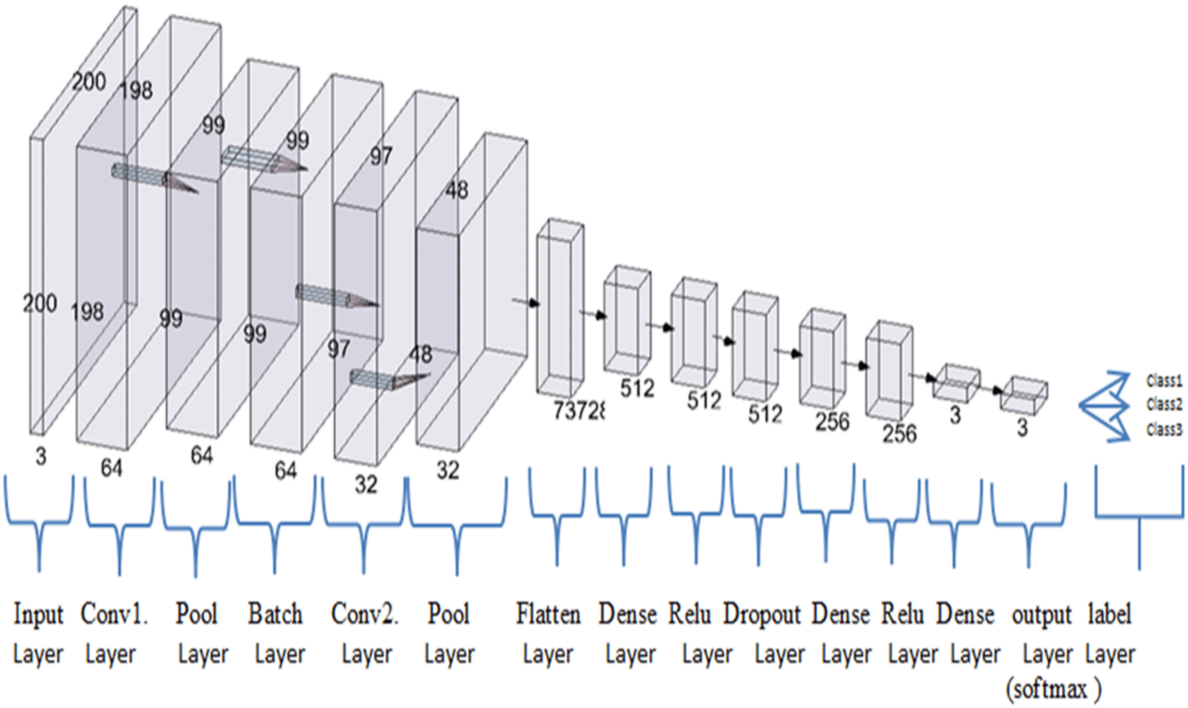

Fig. 4. The proposed CNN model.

TABLE I.
THE ARCHITECTURE PARAMETERS OF THE PROPOSED CNN

Layer No.	Layer (type)	No. Kernel	Kernel size	Output shape	No. parameters
1	Input	-	-	(200, 200, 3)	0
2	conv2d	64	3 × 3	(198, 198, 64)	1792
3	max_pooling2d	-	2 × 2	(99, 99, 64)	0
4	batch_normalization	-	-	(99, 99, 64)	256
5	conv2d	32	3 × 3	(97, 97, 32)	18464
6	max_pooling2d	-	2 × 2	(48, 48, 32)	0
7	flatten	-	-	(73728)	0
8	dense	-	-	(512)	37749248
9	Activation	-	-	(512)	0
10	dropout	-	-	(512)	0
11	dense	-	-	(256)	131328
12	activation	-	-	(256)	0
13	dense	-	-	(3)	771
14	activation	-	-	(3)	0

The model architecture encompasses fourteen primary layers, with layers (L1) to (L7) dedicated to feature extraction, and the final three layers (L9), (L12), and (L14) serving as hidden layers responsible for feature classification. Input Layer (L1): This initial layer receives images of a standardized size, with all training images set to 200 × 200 pixels and three color channels (RGB). There are no trainable parameters at this level. Convolution Layer (L2): Applying 64 filters, each measuring 3 × 3, this layer conducts the convolution operation by convolving filters with the input image. The result is 64 output feature maps sized at 198 × 198. It encompasses 1,792 trainable parameters. Max Pooling Layer (L3): Utilizing nonoverlapping max pooling and a kernel size of 2 × 2, this layer generates 32 feature maps, each measuring 99 × 99. There are no trainable parameters in this layer. Batch Normalization Layer (L4): Expediting learning and stabilizing the neural network, this layer normalizes the output from the preceding layer. Comprising 64 feature maps, each with dimensions of 99 × 99, it contains 256 trainable parameters. Convolution Layer (L5): Similar to L2 but utilizing 32 filters, this layer produces 32 feature maps measuring 97 × 97. It incorporates 18,464 trainable parameters. Max Pooling Layer (L6): Similar to L3, this layer further reduces the size of output feature maps to 48 × 48. There are no trainable parameters in this layer. Flatten Layer (L7): Flattening the 2D arrays from the preceding layer into a 1D array, results in 73,728 trainable parameters. Dense Layer (L8): As the initial hidden layer with 512 neurons, it involves 37,749,248 trainable parameters. Activation Layer (L9): Applying an activation function to the previous layer's output, it contains no trainable parameters. Dropout Layer (L10): Randomly setting a fraction of input units to 0 during training prevents overfitting, and this layer has no trainable parameters. Dense Layer (L11): Serving as the second hidden layer with 256 neurons, it includes 131,328 trainable parameters. Activation Layer (L12): Applying an activation function to the previous layer's output, it contains no trainable parameters. Dense Layer (L13): Representing the third hidden layer with 3 neurons, corresponding to the number of classes in the input images, it has 771 trainable parameters. Activation Layer (L14): Applying an activation function to the output of the previous layer, it contains no trainable parameters. The final three layers in the architecture are designated for feature classification. The initial hidden layer consists of 512 neurons, followed by the second hidden layer with 256 neurons. The third layer comprises 3 neurons, representing the number of classes corresponding to the input images. The CNN model's architecture encompasses 37,901,731 trainable parameters. The training process of the proposed system employs the Adam optimizer with a learning rate of 0.001 and an optimal batch size of 16. Categorical cross-entropy is utilized to compute the loss function, guiding the adjustment of trainable parameters to minimize prediction loss. Table I provides a comprehensive overview of all parameters inherent in the proposed network.

E. Results Analysis

In this stage, CNN classification analysis will be conducted using the best model selected based on the smallest loss and highest accuracy. The model was tested using both training and testing data.

F. Model evaluation

This stage involves evaluating the selected model. The evaluation was performed using the confusion matrix in the CNN classification results based on identifying human identity through various objects such as fish and underwater plants. The objective is to determine the extent of error occurring in the classification. According to Santra et al. [29], The confusion matrix serves as a Table II, providing an overview of the classification model's performance on a set of test data. It facilitates a comparison between the predicted labels and the actual labels, enabling the calculation of correct and incorrect predictions for each class. This stage involves evaluating the selected model. The evaluation was performed using the confusion matrix in the CNN classification results based on identifying human identity through various objects such as fish and underwater plants. The objective is to determine the extent of error occurring in the classification. According to Santra et al [29], The confusion matrix serves as a table II, providing an overview of the classification model's performance on a set of test data. It facilitates a comparison between the predicted labels and the actual labels, enabling the calculation of correct and incorrect predictions for each class. This stage involves evaluating the selected model. The evaluation was performed using the confusion matrix in the CNN classification results based on identifying human identity through various objects such as fish and underwater plants. The objective is to determine the extent of error occurring in the classification. According to Santra et al [29], The confusion matrix serves as a Table II, providing an overview of the classification model's performance on a set of test data. It facilitates a comparison between the predicted labels and the actual labels, enabling the calculation of correct and incorrect predictions for each class. This stage involves evaluating the selected model. The evaluation was performed using the confusion matrix in the CNN classification results based on identifying human identity through various objects such as fish and underwater plants. The objective is to determine the extent of error occurring in the classification. According to Santra et al [29], The confusion matrix serves as a Table II, providing an overview of the classification model's performance on a set of test data. It facilitates a comparison between the predicted labels and the actual labels, enabling the calculation of correct and incorrect predictions for each class. In the context of a three-class set, where 0 represents plants, 1 corresponds to humans, and 2 pertains to fish, the confusion matrix takes the following form:

Here,TP_0, TP_1, and TP_2 denote the True positives for each class, indicating the Number Of Times (NOT) the model correctly predicted plants, humans, and fish, respectively. Similarly, FP_{10}, FP_{20}, FP_{01}, FP_{02}, FP_{12}, and FP_{21} are the False positives for each class, representing the instances where the model incorrectly predicted plants, humans, and fish, respectively. For instance, FT_{01} signifies the Number Of Times (NOT) of the model predicted plants when the actual 0 class

TABLE II.
A CONFUSION MATRIX IS A TOOL TO EVALUATE THE PERFORMANCE OF A CLASSIFICATION MODEL OF THE PROPOSED CNN

	Predicted 0	Predicted 1	Predicted 2
Actual 0	TP_{00}	FP_{01}	FP_{02}
Actual 1	FP_{10}	TP_{11}	FP_{12}
Actual 2	FP_{20}	FP_{21}	TP_{22}

was human. The confusion matrix proves valuable in computing various metrics for model evaluation, including accuracy, precision, recall, and F1 score. The model accuracy, for instance, is expressed as the ratio of the total True Positives for each class to the overall number of predictions:

$$\text{Accuracy} = \frac{TP_0 + TP_1 + TP_2}{TP_0 + TP_1 + TP_2 + FP_{10} + FP_{20} + FP_{01} + FP_{02} + FP_{12} + FP_{21}} \times 100 \tag{10}$$

The model *precision* & *recall* same law, for instance, is expressed as the ratio of the total True Positives for each class to the overall number of predictions:

$$\text{(precision) or (recall)} = \left(\frac{\text{True Positives}}{\text{True Positives + False Positives}}\right) \times 100 \tag{11}$$

The model *F1 score*, is the harmonic mean of *precision* & *recall* is a measure that seeks to achieve a balance of measures of precision and recall, giving an overall score that reflects the performance of the model. Its formula is:

$$\text{F1 score} = 2 \times \left(\frac{\text{precision} \times \text{recall}}{\text{precision + recall}}\right) \tag{12}$$

Additionally, the confusion matrix aids in pinpointing areas where the model makes the most errors, offering insights into potential enhancements. For instance, a high FP_{01} suggests confusion between plants and humans prompting the need for adjustment to feature or the model hyperparameters to mitigate this error [30].

III. RESULTS AND DISCUSSION

A. Data Preprocessing

The initial step in data preprocessing involves assigning a label to each plant, human, and fish image data for easy differentiation between those images that are not processed and those that are processed. This labeling aims to streamline the subsequent stages. The following step in data preprocessing encompasses the scaling and standardization of all categorized images, specifically resizing them to 200 x 200 pixels. This is done to reduce the image size for processing, thereby minimizing the memory load in subsequent operations. The concluding phase in data preprocessing involves the segmentation of the dataset into training, testing, and validation data Table II. The training data is employed to assess the algorithm, the test data is used to evaluate the algorithm's performance, and the validation data aids in assessing the model's effectiveness.

Fig. 5. The training and validation accuracy of the proposed model.

B. Data Preprocessing

The analysis of CNN classification results entails the meticulous utilization of a randomly shuffled dataset, bifurcated into a two-part training set comprising 6071 images and a validation set encompassing 1822 images. Throughout the iterative training process, the model has consistently demonstrated remarkable validation accuracy, reaching an impressive 99.6%. This accuracy is precisely computed using Equation 10.
Fig. 5 visually depicts the dynamic evolution of accuracy throughout both the training and validation phases. Notably, the training model's accuracy initiates at approximately 70% and exhibits consistent improvement with each successive epoch.
Fig. 6 provides a comprehensive view of the loss incurred at the culmination of each epoch. The proposed model exhibits a substantial classification rate coupled with a minimal loss rate. In the CNN classification stage, careful consideration is given to optimizing the model by fine-tuning parameters, specifically the combination of epoch size and batch size. Here, the epoch symbolizes the entire dataset's traversal through a

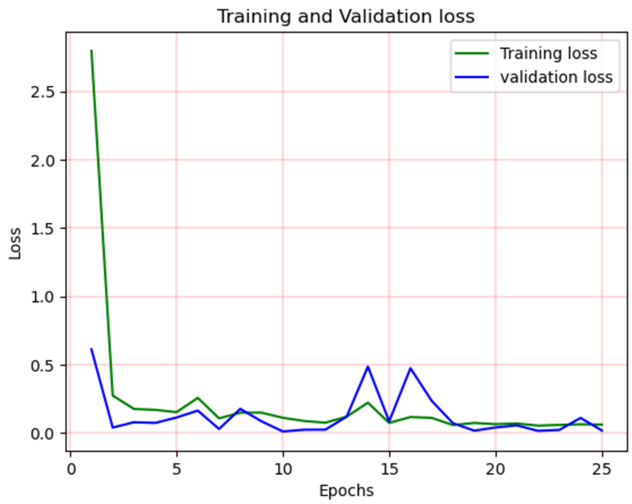

Fig. 6. The training and validation loss of the proposed model.

series of training sessions on the neural network until it completes a single round. Concurrently, the batch size denotes the quantity of samples processed through the neural network at any given moment [31].

1) Optimization of CNN Parameters

The CNN parameter optimization stage involves selecting the CNN model based on the values assigned to the epoch and batch parameters. The tested values for the epoch parameter were [5, 15, 25], and for the batch parameter, the values tested were [16, 32, 64], resulting in a total of 9 combinations of epoch and batch parameter values. The parameter combination was chosen based on achieving the lowest loss value and the highest accuracy in Table III.

2) Classification

The classification stage is a classification process for testing data, using a confusion matrix. The detailed test results have been obtained from 1822 testing data (434 plants, 849 humans, 539 fish), as depicted in Fig. 7

C. Model Evaluation

The model evaluation phase is a vital process to measure the accuracy of the classification model using 1822 validation data see Table Table IV. Note that the average accuracy of the test results for each class may differ from the model accuracy when using test data. This is because the accuracy of test results in test data depends on the discriminant classification function, while the accuracy of testing each class in validation data depends on the probability value of the classifier's prediction results. The Tabel IV shows that the average accuracy obtained from the test results for each class of classification is 99.6%. This is a significant improvement, showing the best class with 99.6% accuracy. Apart from calculations of precision values, the performance of a classification model can be measured by calculations of precision, recall, and F1-score values (see Equations 10, 11, and 12). In addition, the averages of the obtained accuracy values show a rate of up to 99.652%, as well as a recall of 99.584%. In addition, the f1-score was up to 99.618It is observed that the difference between these values is not very large, which shows that the performance of the built classification model is good. Based on these results, it can be concluded that the CNN classification model used is effective, as the results show excellent quality in terms of precision, recall, and f1-score.

IV. Discussion

The results derived from the experiments, as summarized in Table III, indicate an inverse relationship between the batch size used in the CNN algorithm and the accuracy obtained. This aligns with the findings of Kandel and Castelli [32], who noted that larger batch sizes generally do not result in higher accuracy. On the other hand, the number of epochs in a CNN algorithm exhibits a positive correlation with accuracy; as the number of epochs increases, accuracy tends to improve over time. This observation is consistent with Nasrallah et al.'s research [33], which highlighted an increase in the accuracy of the classification system with a higher number of epochs. However, [33] warned about the potential of overfitting when combining increased resolution with a higher number of epochs. Overfitting occurs when an algorithm is trained to model the training data so precisely that its accuracy diminishes significantly when tested with new, unseen

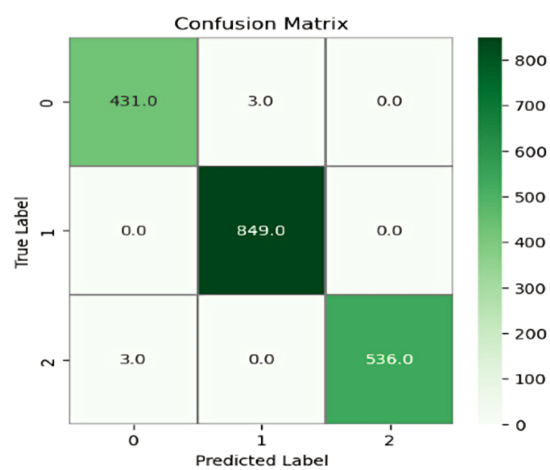

Fig. 7. Confusion matrix for a set of 3 classes, 0 corresponding to plants, 1 corresponding to humans, and 2 corresponding to fish.

TABLE III. ACCURACY AND LOSS VALUES FOR EACH PAIR OF EPOCH AND BATCH PARAMETERS

Batch	5 Epochs		15 Epochs		25 Epochs	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
16	97.08%	0.1514	98.25%	0.0728	98.6%	0.0604
32	90.08%	0.2701	95.48%	0.0901	97.5%	0.0685
64	80.58%	0.4701	91.45%	0.2421	96.2%	0.0697

TABLE IV.
CLASSIFICATION REPORT OF MODEL

	precision	recall	f1-score	support
0	0.99309	0.99309	0.99309	434
1	0.99648	1.00000	0.99824	849
2	1.00000	0.99443	0.99721	539
accuracy	0.99671			1822
macro avg.	0.99652	0.99584	0.99618	1822
weighted avg.	0.99671	0.99671	0.99671	1822

data. To assess whether our model is prone to overfitting, an evaluation was conducted by testing the model with validation data, the details of which are outlined in TableII. (representing new data not used during model training) [34] . Table IV illustrates that the accuracy achieved using the validation data is exceptionally high, reaching 99.6%. This substantial accuracy on validation data provides a strong indication that the CNN model created for classifying plant, human, and fish species is robust and does not suffer from overfitting issues. Table IIcompares the proposed CNN model's performance with other state-of-the-art systems for underwater recognition tasks. The table below summarizes the comparison: Table V

TABLE V.
COMPARES THE PROPOSED CNN MODEL'S PERFORMANCE WITH OTHER STATE-OF-THE-ART SYSTEMS FOR UNDERWATER RECOGNITION TASKS. THE TABLE BELOW SUMMARIZES THE COMPARISON

ref	Model Use	dataset	accuracy
[11]	SVM	320	90%
[35]	KNN	200	94.65%
[36]	Deep CNN	1525	88%
[37]	MobileNetV2	2000	93.5%
Proposed Model	CNN	6071	99.6%

SVM (Support Vector Machine), **KNN** (K-Nearest Neighbors).

underscores the superior performance of the proposed CNN model, showcasing its capacity to handle a larger dataset effectively while achieving the highest accuracy among the compared methods

V. CONCLUSION

This research has demonstrated the effectiveness of using a Convolutional Neural Network (CNN) for classifying underwater images into humans, fish, and plants. The dataset used has 6071 samples, these samples are divided into two parts, 70% (4249) of samples are used for the training model, and 30% (1828) are used for testing. The proposed CNN model achieved an impressive accuracy of 99.6% on the testing dataset, outperforming other modern machine-learning approaches. The high accuracy highlights CNN's strength in extracting relevant features from image data for accurate classification. The findings have significant implications for enhancing underwater search and rescue operations. By accurately distinguishing humans from other underwater objects like fish and plants, rescue teams can pinpoint the location of potential drowning victims more efficiently. This capability can lead to faster response times and improved chances of survival in emergencies. Moreover, the research contributes to the broader field of underwater image analysis and computer vision. The CNN model's robustness to challenges and limitations in underwater conditions, such as water movement, low visibility, and distortions, showcases its applicability in various marine environments and scenarios. Future work could explore integrating the CNN model into real-time underwater monitoring systems or expanding the classification to include additional classes of interest. Additionally, investigating techniques to further improve the model's generalization capabilities across diverse underwater environments would be valuable. Overall, this study highlights the potential of deep learning, particularly CNNs, in tackling complex underwater image classification tasks. The proposed approach paves the way for more effective and innovative solutions in marine research, exploration, and emergency response operations.

CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

REFERENCES

[1] Y. S. Samant, "Image processing: Challenges and application," *IJARSCT*, vol. 5, no. 2, 2021.

[2] Y. Kaya, S. Hong, and T. Dumitras, "Shallow-deep networks: Understanding and mitigating network overthinking," in *International Conference on Machine Learning*, pp. 3301–3310, 2019.

[3] H. Qin, X. Li, Z. Yang, and M. Shang, "When underwater imagery analysis meets deep learning: a solution at the age of big visual data," in *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–5, 2015.

[4] D. Cireșan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[5] D. P. Williams and J. Groen, "A fast physics-based, environmentally adaptive underwater object detection algorithm," in *Proceedings of the OCEANS*, pp. 1–7, 2011.

[6] E. Galceran, V. Djapic, M. Carreras, and D. P. Williams, "A real-time underwater object detection algorithm for multi-beam forward looking sonar," in *IFAC Proceedings Volumes*, vol. 45, pp. 306–311, 2012.

[7] R. Aziz, C. Verma, and N. Srivastava, "A fuzzy-based feature selection from independent component subspace for machine learning classification of microarray data," *Genomics Data*, vol. 8, pp. 4–15, 2016.

[8] T. Anwar and S. Zakir, "Effect of image augmentation on ecg image classification using deep learning," in *2021 International Conference on Artificial Intelligence (ICAI)*, pp. 182–186, 2021.

[9] M. Saberioon, P. Císař, L. Labbé, P. Souček, P. Pelissier, and T. Kerneis, "Comparative performance analysis of support vector machine, random forest, logistic regression and k-nearest neighbors in rainbow trout (oncorhynchus mykiss) classification using image-based features," *Sensors*, vol. 18, no. 4, p. 1027, 2018.

[10] N. J. C. Strachan, P. Nesvadba, and A. R. Allen, "Fish species recognition by shape analysis of images," *Pattern Recognition*, vol. 23, no. 5, pp. 539–544, 1990.

[11] A. Rova, G. Mori, and L. M. Dill, "One fish, two fish, butterfish, trumpeter: Recognising fish in underwater videos," in *IAPR Conference on Machine Vision Applications*, pp. 404–407, 2007.

[12] S. Raine, R. Marchant, P. Moghadam, F. Maire, B. Kettle, and B. Kusy, "Multi-species seagrass detection and classification from underwater images," in *Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2020.

[13] F. Han, J. Yao, H. Zhu, and C. Wang, "Underwater image processing and object detection based on deep cnn method," *First published*, 2020.

[14] R. Rismiyati and A. Luthfiarta, "Vgg16 transfer learning architecture for salak fruit quality classification," *Telematika*, vol. 18, no. 1, p. 37, 2021.

[15] N. Ibrahim, G. A. Lestary, F. S. Hanafi, K. Saleh, N. K. C. Pratiwi, M. S. Haq, and A. I. Mastur, "Classification of tea leaf shoot maturity level using convolutional neural network method," *ELKOMIKA: Journal of Electrical Energy Engineering, Telecommunication Engineering, & Electronic Engineering*, vol. 10, no. 1, p. 162, 2022.

[16] F. Kratzert and H. Mader, "Fish species classification in underwater video monitoring using convolutional neural networks," in *IEEE OCEANS 2018 MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–6, 2018.

[17] A. Salman, S. A. Siddiqui, F. Shafait, A. Mian, M. R. Shortis, K. Khurshid, *et al.*, "Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system," *ICES Journal of Marine Science*, vol. 77, no. 4, pp. 1295–1307, 2020.

[18] V. J. D. Almero, R. S. Concepcion, E. Sybingco, and E. P. Dadios, "An image classifier for underwater fish detection using classification tree-artificial neural network hybrid," in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pp. 1–6, 2020.

[19] S. Villon, D. Mouillot, M. Chaumont, E. S. Darling, G. Subsol, T. Claverie, *et al.*, "A deep learning method for accurate and fast identification of coral reef fishes in underwater images," *Ecological Informatics*, vol. 48, pp. 238–244, 2018.

[20] A. Michele, V. Colin, and D. D. Santika, "Mobilenet convolutional neural networks and support vector machines for palmprint recognition," *Procedia Computer Science*, vol. 157, pp. 110–117, 2019.

[21] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Object instance segmentation and fine-grained localization using hypercolumns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1–14, 2017.

[22] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high-quality object detection," in *CVPR*, pp. 6154–6162, 2018.

[23] N. P. Ramaiah, E. P. Ijjine, and C. K. Mohan, "Illumination invariant thermal face recognition using convolutional neural network," in *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, pp. 1–5, 2015.

[24] C. Buchanan, Y. Bi, B. Xue, R. Vennell, S. Childerhouse, M. K. Pine, D. Briscoe, and M. Zhang, "Deep convolutional neural networks for detecting dolphin echolocation clicks," in *The 6th International Conference on Image and Vision Computing*, pp. 1–6, 2021.

[25] U. A. Badawi, "Fish classification using extraction of appropriate feature set," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, pp. 2488–2500, 2022.

[26] R. Rokhana, J. Priambodo, T. Karlita, I. Sunarya, E. Yuniarno, I. Purnama, and M. Purnomo, "Convolutional neural network untuk pendeteksian patah tulang femur pada citra ultrasonic b-mode," *J. Nasional Teknik Elektro Dan Teknologi Informasi*, vol. 8, no. 1, pp. 59–67, 2019.

[27] I. Putra, A. Wijaya, and R. Soelaiman, "Klasifikasi citra menggunakan convolutional neural network (cnn) pada caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, pp. A65–A69, 2016.

[28] A. Fauzan, M. Wahyuddin, and S. Ningsih, "Pleural effusion classification based on chest x-ray images using convolutional neural network," *Journal of Computer Science and Information*, vol. 14, no. 1, pp. 9–16, 2021.

[29] A. Santra and C. Christy, "Genetic algorithm and confusion matrix for document clustering," *International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 322–328, 2012.

[30] J. Han, K. Micheline, and P. Jian, *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2012.

[31] A. Devarakonda, M. Naumov, and M. Garland, "Adabatch: adaptive batch sizes for training deep neural networks," in *Workshop Track - ICLR*, pp. 1–4, 2017.

[32] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, 2020.

[33] F. Nashrullah, S. Adhi, and G. Budiman, "Investigasi parameter epoch pada arsitektur resnet-50 untuk klasifikasi pornografi," *Journal of Computer, Electronic, and Telecommunication*, vol. 1, no. 1, pp. 1–8, 2021.

[34] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, no. 2, pp. 1–6, 2019.

[35] S. S. Murugan, R. Ganeshbabu, and C. Srinivasan, "Underwater object recognition using knn classifier," *International Journal of Multidisciplinary Sciences and Research*, vol. 9, no. 3, pp. 10–14, 2017.

[36] R. A. Dakhil and A. R. H. Khayeat, "Deep learning for enhanced marine vision: Object detection in underwater environments," *International Journal of Electrical and Electronics Research (IJEER)*, vol. 11, no. 4, pp. 1209–1218, 2023.

[37] G. W. Thum, S. H. Tang, S. A. Ahmad, and M. Alrifaey, "Toward a highly accurate classification of underwater cable images via deep convolutional neural network," *Journal of Marine Science and Engineering*, vol. 8, no. 11, p. 924, 2020.