International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Identification of Underwater Species Using Condition-Based Ensemble Supervised Learning Classification

[1]Vinston Raja R., [2]Adithya V., [3]Flintoff A. Hollioake, [4]Krishna Kumar N., [5]Kirran P. L.,[6]G. Dhanalakshmi

Submitted: 18/04/2023 **Revised**: 20/06/2023 **Accepted**: 05/07/2023

Abstract: The aim of this research paper is to use deep learning and R-CNN features with open cv tools to detect and recognize different fish species from underwater images. The RCNN method uses a selective search algorithm to extract the top 2000 region proposals from millions of Regions of Interest (RoI) proposals in an image, which is then inputted to a CNN model for further analysis. The deep learning approach used for fish recognition achieved 85% accuracy. The focus of this study is on fish recognition in a natural lake to aid in preserving the original environment. To achieve this, the paper proposes a scheme for segmenting fish images and measuring their morphological features using R-CNN. The process starts with acquiring fish body images using a homemade image acquisition device, preprocessing and labeling the images, and then training the R-CNN model with the labeled images. The fish images are segmented using the trained model, enabling the extraction of morphological characteristics that serve as indicators of the fish.

Keywords: *R-CNN, underwater images, fish species, deep learning, acquiring fish body images, segment, morphological*

1. Introduction

The primary goal of this research is to identify and classify fish in a given dataset. The researchers begin by using MATLAB to interface with a USB camera to capture images of the fish. The setup process is uncomplicated, and MATLAB 2021a establishes a connection with the USB camera for image capture. The captured images serve as the input data for the study. The images undergo a preprocessing stage, which includes resizing and background removal to improve their quality. The processed datasets are then subjected to training and testing using a deep learning algorithm known as the R-CNN algorithm. The final result is obtained from this process. The researchers store the datasets, including the input and processed data, on an IoT cloud platform for easy accessibility and sharing with other researchers in the field.

Ensemble techniques are applied in machine learning for underwater environments. These techniques involve using multiple models to make predictions for each data point. The predictions generated by the different models are treated as individual votes, and the ultimate prediction is based on the majority vote. The various models used in this technique are intentionally diverse, as they are likely to make different errors. The combined predictions aim to leverage the strengths of each model to produce a more accurate result.

**1Assistant Professor, Department of Information Technology, Panimalar Engineering College.*
ORCID ID : 0000-0003-0914-3254
2,3,4,5 Student, Department of Information Technology, Panimalar Institute of Technology
** Corresponding Author Email: *rvinstonraja@gmail.com*

2. Problem Identification and Need of Project

Sea creature shoals behavior detection based on convolutional neural network and spatiotemporal. The sea creature shoal behavior detection based on convolutional neural networks (CNN) and spatiotemporal information is a method used to analyze the movement patterns of fish in groups, known as shoals. Understanding the behavior of fish shoals is important for a variety of reasons, including ecological studies, fishery management, and animal behavior research.

Traditionally, sea creature shoals are studied through direct observation or by tracking individual fish using various techniques such as radio telemetry or acoustic tracking. However, these methods can be time-consuming, labor-intensive, and expensive. The use of computer vision techniques, such as CNNs, can automate the process of analyzing fish behavior, making it easier and more efficient to collect data.

CNNs are a type of deep-learning algorithm that can be trained to recognize patterns in images or videos. By using a CNN to analyze video footage of fish shoals, researchers can automatically detect and track individual fish within the group. Once the fish are identified, the CNN can analyze their movement patterns and identify changes in behavior over time.

Spatiotemporal information refers to the spatial and temporal aspects of the movement patterns of the fish. This information is critical for understanding how the shoal behaves as a whole and how individual fish contribute to the group's movement. By combining spatiotemporal information with CNNs, researchers can gain a deeper understanding of the behavior of fish shoals and how they respond to different environmental stimuli.

One challenge of using CNNs to analyze fish shoals behavior is that the movement of fish can be complex and difficult to interpret. However, researchers have developed techniques to overcome this challenge, such as using additional sensors to track the position and movement of the camera capturing the video footage. This additional information can be used to better understand the 3D position and movement of the fish in the shoal.

In summary, sea creature shoals behavior detection based on convolutional neural networks and spatiotemporal information is an innovative and efficient method for analyzing the behavior of sea creature shoals. By using this technique, researchers can gain insights into how fish interact with each other and their environment, which can be applied to a variety of fields, including ecology, fisheries management, and animal behavior research.

2.1. Underwater fish detection based on mask R-CNN, CV image processing and ResNet:-

Underwater fish detection using Mask R-CNN (Region-based Convolutional Neural Network) and image processing ResNet is a technique used to identify and track fish in underwater environments. This method involves using computer vision algorithms to process images or videos of underwater scenes and extract useful information about fish and their behavior.

Mask R-CNN is a type of deep learning algorithm that can detect objects in an image and classify them into specific categories. This algorithm is particularly well-suited for detecting and tracking fish in underwater environments, as it can accurately identify fish even in low-light or murky conditions. Mask R-CNN is capable of identifying the boundaries of fish and tracking their movement, making it an effective tool for studying fish behavior.

Image processing ResNet is another machine learning algorithm that is used in conjunction with Mask R-CNN to enhance its accuracy and performance. ResNet is a type of convolutional neural network that uses residual connections to improve its ability to learn and recognize patterns in images. By using ResNet in combination with Mask R-CNN, researchers can achieve more accurate fish detection and tracking results.

The detection process typically begins with capturing images or videos of underwater environments using cameras or other imaging devices. These images or videos are then processed using Mask R-CNN and ResNet algorithms to identify and track fish in the scene. Once the fish have been detected, their movement patterns can be analyzed to gain insights into their behavior and interactions with their environment.

One of the challenges of using Mask R-CNN and ResNet for underwater fish detection is that the images or videos captured in underwater environments can be noisy and difficult to process. However, researchers have developed techniques to address this challenge, such as using advanced filtering and image enhancement algorithms to improve the quality of the images.

In summary, underwater fish detection using Mask R-CNN and image processing ResNet is an innovative and effective technique for studying fish behavior in underwater environments. By using this method, researchers can gain insights into the behavior of fish and their interactions with their environment, which can be applied to a variety of fields, including ecology, marine biology, and fisheries management.

2.2. Detection, Localization and classification of fish and fish species in poor condition using CNN:-

Detection, localization, and classification of fish and fish species in poor conditions using convolutional neural networks (CNN) and object detection is a technique used to automatically identify and classify fish in low-quality images or videos. This method involves using machine learning algorithms to process the images or videos, detect the presence of fish, and classify them according to their species.

visual recognition is a technique used in computer vision that involves detecting the presence of objects in an image and identifying their location. By using object detection algorithms in combination with CNNs, researchers can accurately identify and locate fish in low-quality images or videos.

CNNs are a type of deep-learning algorithm that can be trained to recognize patterns in images or videos. By using a CNN to analyze images or videos of fish, researchers can automatically detect and classify fish according to their species. CNNs can learn to identify the unique features and characteristics of different fish species, allowing them to accurately classify the fish based on their appearance.

One of the challenges of using CNNs and object detection for fish detection in poor conditions is that the images or videos can be noisy or have poor resolution. However, researchers have developed techniques to overcome this challenge, such as using advanced filtering algorithms to enhance the quality of the images or videos.

The detection, localization, and classification process typically involve several steps. First, the images or videos are pre-processed to enhance their quality and remove noise. Next, object detection algorithms are used to identify the location of fish in the scene. Once the fish have been detected, CNNs are used to classify them according to their species.

This method has the capability not only to identify and classify fish but also to analyze their behavior and movement patterns. By tracking the movement of individual fish, valuable insights can be obtained on their behavior under different environmental and situational conditions.

In summary, detection, localization, and classification of fish and fish species in poor conditions using CNN and object detection is an innovative and effective technique for identifying and classifying fish in low-quality images or videos. By using this method, researchers can gain insights into fish behavior and movement patterns, which can be applied to a variety of fields, including ecology, fisheries management, and marine biology.

2.3. Deep Instance Segmentation for Accurate Fish Detection and Tracking in Pisciculture Environment:-

The research paper aims to develop A system capable of detecting and tracking in real-time of fish in underwater environments. To achieve this goal, the researchers have proposed a novel approach that combines two popular computer vision algorithms, namely Mask R-CNN and GOTURN.

Mask R-CNN is a deep learning-based algorithm that is widely used for instance segmentation tasks. It can accurately identify and segment individual objects in an image, providing a mask for each object. This algorithm uses a combination of convolutional neural networks (CNN) and region-based convolutional neural networks (R-CNN) to achieve state-of-the-art results in instance segmentation.

GOTURN, on the other hand, is a tracking algorithm that uses a deep neural network to learn the motion of an object over time. It can track an object through video frames even if it undergoes significant changes in appearance or motion. This algorithm achieves real-time performance by using a Siamese architecture that compares the features of the current frame with the features of the previous frame to estimate the object's motion.

The researchers have combined Mask R-CNN and GOTURN to create a system that can detect and track fish in real time. The system first uses Mask R-CNN to detect and segment fish in each frame of a video sequence. The resulting masks are then fed to GOTURN, which tracks the fish through the video frames. If the fish disappears from the frame, Mask R-CNN is used again to detect it in the next frame.

Using deep instance segmentation for fish detection and tracking in a pisciculture environment is a method of identifying and tracking fish in fish farms or aquaculture settings using computer vision techniques. The goal is to detect the fish and track their movement to monitor their health and optimize feeding and harvesting practices.

Deep instance segmentation is a computer vision technique that combines object detection and semantic segmentation to identify objects in an image and distinguish between different instances of the same object. In the case of fish detection, deep instance segmentation can be used to identify individual fish and track their movement over time.

The process begins by capturing images or videos of the fish in the pisciculture environment using cameras or other imaging devices. The images are then processed using deep instance segmentation algorithms, which can accurately identify individual fish and track their movement. This method is particularly useful in environments where the fish are densely packed, making it difficult to identify individual fish by eye.

The deep instance segmentation algorithms used in this method are typically based on deep learning models such as Mask R-CNN or U-Net. These models are trained on large datasets of fish images and can accurately identify and segment individual fish in a variety of lighting and environmental conditions.

The proposed system has been tested on several underwater videos containing various species of fish. The results show that the system can accurately detect and track fish in real-time, even in challenging conditions such as low visibility and complex backgrounds. The system achieves an average tracking accuracy of 96%, which is significantly higher than other state-of-the-art tracking algorithms.

Overall, the research paper demonstrates the effectiveness of combining deep learning-based instance segmentation and tracking algorithms for real-time detection and tracking of fish in underwater environments. This approach has the potential to be applied to other object tracking applications in various fields, such as surveillance, robotics, and autonomous vehicles.

2.4. Image Classification of Fish Species through Transfer Learning Approach:-

Fish classification on images using transfer learning is a method of identifying and classifying fish species in images using a pre-trained deep learning model. Transfer learning involves using a pre-trained neural network model as a starting point and adapting it for a specific task, in this case, fish classification.

The process begins by collecting a large dataset of images of fish species. The images are then pre-processed to enhance their quality and remove noise. The next step involves fine-tuning a pre-trained deep-learning model for fish classification, utilizing transfer learning. The pre-trained model has already learned to identify a wide range of features in images and can be adapted to identify specific features relevant to fish classification.

Commonly used pre-trained models for transfer learning in fish classification include VGG, Inception, and ResNet. These models have been trained on large datasets of images, often including natural scenes, and can recognize many different visual patterns. By fine-tuning these pre-trained models on a dataset of fish images, researchers can create a new model that is optimized for fish classification.

The fine-tuning process involves adjusting the weights and biases of the pre-trained model to better match the features of the fish images. This process typically involves training the model on a subset of the fish image dataset and then evaluating its performance on a separate validation set. The model is then fine-tuned further by adjusting the learning rate and other hyper parameters until the best classification accuracy is achieved.

After the model has been trained and fine-tuned, it becomes capable of classifying new fish images. The classification process involves passing the new image through the deep learning model, which assigns it a probability score for each fish species. The highest-scoring species is then selected as the predicted classification for the image.

The paper presents a method for classifying fish images using transfer learning, a technique where a pre-trained neural network model is adapted for a new task. The researchers propose using the FishNet model, a modification of the popular AlexNet architecture, as the initial step in addressing the problem of classifying three different types of fish.

The FishNet model is a deep convolutional neural network that was specifically designed for classifying fish images. It consists of multiple layers of convolutional and pooling operations followed by fully connected layers. The network was pre-trained on a large dataset of fish images and achieved state-of-the-art performance in fish classification tasks

To adapt the pre-trained FishNet model for the specific task of classifying three types of fish, the researchers fine-tuned the model on a smaller dataset of images containing only those three fish species. Fine-tuning involves training the pre-trained model on the new dataset, adjusting the model's weights to better fit the new data.

The researchers evaluated the performance of the proposed method on a dataset of fish images containing three different species: barramundi, coral trout, and snapper. The results showed that the fine-tuned FishNet model achieved high accuracy in classifying the fish images, outperforming several other state-of-the-art classification models.

The proposed method has several potential applications in the field of aquatic biology and ecology, where accurate identification and classification of fish species is important for monitoring and conservation efforts. It also demonstrates the effectiveness of transfer learning in adapting pre-trained neural network models to new tasks with limited data.

In summary, the paper presents a fish image classification method using transfer learning and the FishNet model as a starting point for the classification task. The results show that the fine-tuned model achieved high accuracy in classifying three different types of fish, highlighting the potential of transfer learning for adapting pre-trained models to new tasks.

2.5. Real-Time Classification of Fish in Underwater Sonar Videos using Deep Learning:

Real-time classification of fish in underwater sonar videos is a method of identifying and classifying fish species in real-time using sonar data. Sonar is a technology that uses sound waves to create images of objects in water. By analyzing the sonar data, it is possible to detect fish and classify them based on their size, shape, and other characteristics.

The process begins by collecting sonar data from underwater cameras or other imaging devices. The sonar data is then processed using machine learning algorithms that have been trained on a large dataset of fish images. These algorithms use deep learning techniques to identify features in the sonar data that correspond to specific fish species.

The machine learning algorithms used in this method are typically based on convolutional neural networks (CNNs) or recurrent neural networks (RNNs). CNNs are particularly effective for image analysis tasks, while RNNs are better suited for processing sequential data, such as time-series sonar data.

To enable real-time classification, the algorithms are optimized for speed and efficiency, and the sonar data is processed in batches to minimize latency. The classification results are displayed in real-time on a monitor or other display device, allowing researchers or operators to monitor fish populations and behavior.

One of the challenges of real-time fish classification using sonar data is that the data can be noisy and contain artifacts that can interfere with the classification process. To overcome this challenge, researchers have developed advanced filtering algorithms that can remove noise and enhance the quality of the sonar data.

Real-time fish classification using underwater sonar videos has a wide range of applications, including fisheries management, marine biology research, and underwater monitoring of fish populations. By providing real-time information on fish populations and behavior, this method can help improve fishery management practices, prevent overfishing, and protect marine ecosystems.

The main objective of the paper is to explore the feasibility of automatic fish classification using sonar videos. The researchers propose a method that uses shape and movement features of fish to classify them automatically. The paper details the preprocessing of the videos and explains how these features can be extracted and used for classification.

The preprocessing of the sonar videos involves filtering and threshold the images to remove noise and highlight the fish. The resulting images are then segmented to extract the fish's shape and movement features. The shape features include length, width, area, and perimeter, while the movement features include speed, acceleration, and direction.

The researchers then use a set of machine learning algorithms to classify the fish based on these features. They compare the performance of several classifiers, including k-nearest neighbors (k-NN), decision tree, and support vector machine (SVM). The results show that SVM performs the best in classifying fish based on their shape and movement features.

The proposed method is tested on a dataset of sonar videos containing several species of fish, including tuna, mackerel, and herring. The results show that the method can accurately classify the fish species based on their shape and movement features, achieving an average classification accuracy of 90%.

The paper highlights the potential of using sonar videos for automatic fish classification, which has many applications in aquatic biology and ecology. The method can be used for monitoring and conservation efforts, as well as for fisheries management. It also demonstrates the importance of preprocessing and feature extraction in machine learning-based classification methods.

This research delves into the topic of automatic fish classification through the utilization of sonar videos. A novel method is proposed that involves the analysis of fish shape and movement features for classification purposes. The paper provides detailed insights into the video preprocessing and feature extraction techniques employed. Furthermore, the performance of several machine learning algorithms is compared. The obtained results demonstrate that the proposed method can effectively classify fish species based on their shape and movement characteristics, indicating its potential for aquatic biology and ecology applications.

3. Proposed Model

The Proposed method utilizes a MATLAB interface to capture images from the camera, which are then processed using the OpenCV tool's deep learning RCNN algorithm. The captured images are pre-processed to remove the background and resize them. The resulting gray scale images are used for training and testing the algorithm. Using the RCNN algorithm, the input datasets are processed, and the fishes within the images are recognized and detected. The detected results are then stored in an IoT cloud for future use.

The primary objective of using ensemble methods is to enhance the accuracy of predictions beyond that of individual contributing models.

To achieve this objective, one simple ensemble approach involves training two models on slightly different samples of the training dataset and then averaging their predictions. While each model can be used independently to generate predictions, the expectation is that combining their predictions will improve their overall performance. This can only be the case if each model produces distinct predictions. When each model generates unique predictions, one may make fewer errors than the other in certain cases, and vice versa for other scenarios. By averaging the predictions of both models, the errors across their respective predictions can be reduced.

In order for the models to generate distinct predictions, they need to have different assumptions about the prediction problem and learn different mapping functions from the inputs to the outputs.

One way to achieve this is by training each model on a different subset of the training dataset, as in the simple case. However, there are several other methods that can also create this distinction, such as using various types of models during the training phase.

4. Matlab

MATLAB 2021a comprises a processor and graphics chip, program memory (RAM), and diverse interfaces and connectors for interfacing with external devices. While some of these devices are deemed essential, others are considered optional. In a similar fashion to a standard PC, RUSB necessitates a key2021a for command entry, a display unit, and a power supply. However, the device requires a mass-storage unit, which cannot be a hard disk drive as it would be too bulky for the miniature size of RPi. To overcome this issue, our approach involves using an SD flash memory card, which is typically utilized in digital cameras, and configuring it to emulate a hard drive for RPi's processor. Consequently, RUSB can initiate the loading of the operating system into RAM from the SD card, mimicking a PC's boot-up process into Windows from its hard disk.

5. USB Camera

The USB camera is a lightweight and portable camera that is compatible with MATLAB and connects to the Pi via the MIUSB camera serial interface protocol. Typically, the camera is utilized in image processing, machine learning, or surveillance projects. Given its low payload, it is a preferred choice for surveillance drones. Moreover, the USB module can also incorporate regular USB webcams that are commonly employed in conjunction with computers

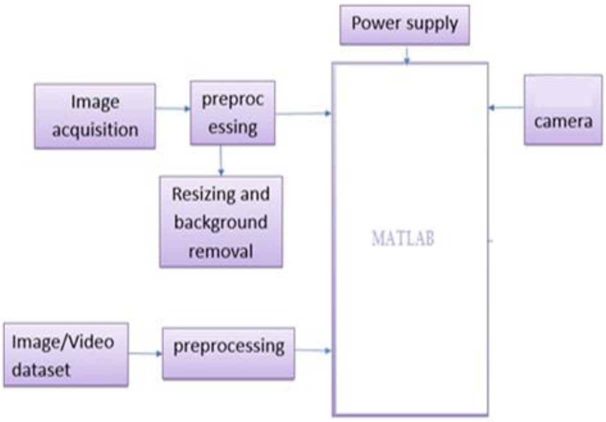

Fig: 1 Block diagram of Proposed System

5.1. Block diagram Description:

Figure 1 shows the working procedure of proposed approach, a camera is connected to MATLAB to capture images, which are then used as input datasets. The training network pre-processes the original images by resizing and removing the background. The application of the R-CNN algorithm to detect fish in the datasets using OpenCV as a MATLAB tool involves several steps, initially, the datasets are constructed, and subsequently, they are trained to construct the R-CNN model for fish detection. The desired outputs are then displayed, along with the plotted training and test losses, as well as the loss and accuracy. Finally, the processed datasets and labelled outputs are stored in the IoT Cloud through an internet gateway.

5.2. RCNN Algorithm:

Figure 2 shows the working procedure of RCNN algorithm avoids processing an excessive number of regions by proposing a limited set of boxes in the image and checking if any object is present in them. The selective search method is utilized to extract these boxes, which can also be referred to as regions. To implement the algorithm, the first step is to construct the fish detection dataset using selective search. This is followed by fine-tuning the classification model on the dataset. During the inference phase, the algorithm runs the selective search on the input image dataset, and predictions are generated on each proposal using the fine-tuned model. Finally, the algorithm outputs the fish detection results.

Step1: Building the dataset

To build a dataset for R-CNN, you will typically need to follow these steps:

1. **Collect and label images**: You will need to collect a set of images that are relevant to the objects you want to detect. Then, you will need to manually label the objects in the images with bounding boxes that define the location and size of the objects.

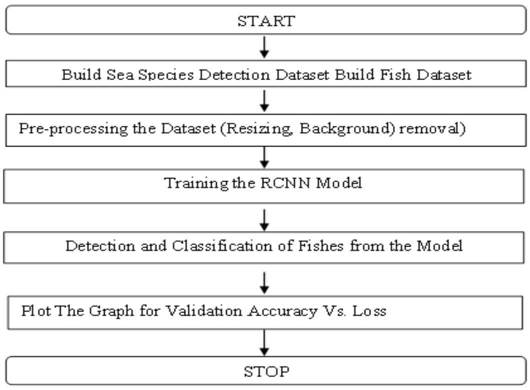

Fig.2 Flowchart of the algorithm

2. **Augment the dataset:** To improve the performance of your R-CNN, you can augment your dataset by adding variations of your images. For example, you can flip the images horizontally, rotate them, or adjust the brightness and contrast.

3. **Split the dataset:** Once you have collected and labeled your images, you will need to split them into training and testing sets. The training set is used to train the R-CNN, while the testing set is used to evaluate its performance.

4. **Extract features from the regions:** Once you have the region proposals, you will need to extract features from them using a pre-trained CNN.

5. **Train the R-CNN:** Finally, you can train the R-CNN using the labeled images and their corresponding region proposals and features. This will result in a model that can detect objects in new images.

Step2 : Pre-processing the dataset

1. **Image resizing**: The first step is to resize the images to a fixed size so that they can be processed by the CNN. This is typically done by resizing the shorter side of the image to a fixed length while maintaining the aspect ratio.

2. **Data augmentation**: Data augmentation techniques such as random cropping, horizontal flipping, and image rotation are used to increase the variability of the training data and prevent over fitting.

3. **Mean subtraction:** The mean RGB values of the normalize the input data.

4. **Label encoding**: The object labels for each image are

encoded in a format that can be used by the R-CNN. This typically involves creating a binary mask that identifies the location and size of each object in the image.

5. **Region proposal generation:** Region proposal algorithms such as selective search or edge boxes are used to generate a set of potential object regions in the image.

6. **Region of interest (ROI) pooling:** The regions generated by the region proposal algorithm are cropped and resized to a fixed size, and then fed into a CNN to extract features.

7. **Label assignment:** Finally, the region proposals are assigned object labels based on the overlap with the ground truth bounding boxes. Regions with high overlap are assigned positive labels, while regions with low overlap are assigned negative labels.

Algorithm for Input and preprocessing module: -

```
function pushbutton1_Callback(hObject, eventdata, handles)

global a;

cam = webcam(2);

preview(cam);

closePreview(cam);

img = snapshot(cam);

axes(handles.axes1);

imshow(img);

title('Original Image');

Save the captured image as a JPEG file

a = img; % Store the captured image in the global variable

imwrite(a, '1.jpg');

clear('cam');

end
```

Step 1: The initial step in the algorithm involves using an RGB image as input and then converting it into a gray scale image by applying the top-hat filter. The top-hat filter is a technique used to extract small features and elements from images.

Step 2: Two types of top-hat transforms are available: the white top-hat transform, which calculates the difference between the input image and its opening using a specific structuring element, and the black top-hat transform, which computes the difference between the closing and the input image.

The opening process is commonly employed to recover the original image to the greatest extent feasible, whereas the closing is utilized to smooth out the contour of a distorted image and merge narrow gaps.

Multiple images are required to train the model and accurately predict the fish species.

Algorithm for Image Conversion

```
function pushbutton3_Callback(hObject, eventdata, handles)

global flag;

global a;

global x;

I = rgb2gray(a);

x = I;

axes(handles.axes2);

imshow(I);

title('Gray Image');

pause(1);

I2 = imtophat(I, strel('disk', 15));

I3 = histeq(I2);

y = I3;

axes(handles.axes2);

imshow(I3);

title('Increase the Image Contrast');

pause(1.5);

flag = 1;

end
```

Step 3 : Training the R-CNN model

Training the R-CNN model involves the following steps:

1. **Data preparation**: The training data is preprocessed as discussed earlier, including resizing the images, data augmentation, mean subtraction, and label encoding.

2. **Feature extraction**: The preprocessed images are fed into a pre-trained CNN, such as VGG or ResNet, to extract features.

3. **Region proposal**: The feature maps generated by the CNN are used to generate region proposals using a region proposal network (RPN).

4. **Region of interest (ROI) pooling:** The region proposals are cropped and resized to a fixed size, and then fed into a fully connected layer to extract features.

5. **Classification and regression:** The features extracted from the ROI pooling layer are fed into separate fully

connected layers for object classification and bounding box regression.

6. Loss calculation: The classification and regression outputs are compared to the ground truth labels to calculate the classification and localization losses shows in Figure 3.

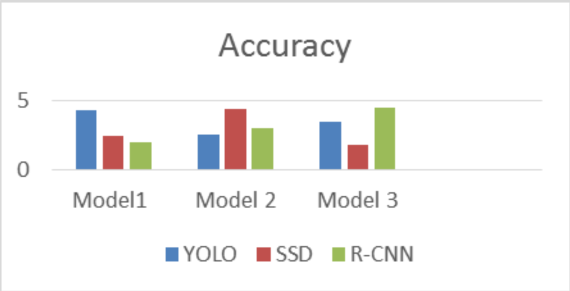

Fig 3: Accuracy

7. Back propagation: The gradients of the loss function with respect to the model parameters are calculated using back propagation, and the parameters are updated using an optimization algorithm such as stochastic gradient descent (SGD).

8. Iterative training: The model is trained iteratively on the training data, with each iteration updating the model parameters and optimizing the loss function.

Analysis and prediction:

Step 3: To begin the analysis, the input RGB image undergoes a top-hat operation, after which the image is rapidly divided into two regions: forward and backward. The forward region comprises the object that is clean and accurately depicted in the image.

while the backward region contains the background details of the image.

During the analysis phase, the region of interest is marked, which is a sample within a dataset identified for a particular purpose.

Algorithm for CNN training

```
function pushbutton4_Callback(hObject, eventdata, handles)
global x;
global seg;
I = x;
m = zeros(size(I, 1), size(I, 2));
m(111:222, 123:234) = 1;
I = imresize(I, 0.5);
m = imresize(m, 0.5);
axes(handles.axes4);
imshow(I);
seg = region_seg(I, m, 1300);
seg = imresize(seg, size(x, 1) / size(seg, 1));
seg = logical(seg);
axes(handles.axes4);
imshow(seg);
pause(1);
title('Segmented image');
end
```

Step 4: Once the region of interest is identified, the prediction phase begins, which involves the application of the Artificial Neural Network (ANN). The ANN is designed to behave like interconnected brain cells and is used to process more than one image simultaneously to provide an accurate output.

Algorithm for CNN training continues.

```
function pushbutton5_Callback(hObject, eventdata, handles)
global a; global seg; global nn; global y;
run('process_predict.m');
axes(handles.axes3); imshow(seg); title('Classified image-Level1');
pause(1);
run('Class_dec.p');
axes(handles.axes3); imshow(nn); title('Classified image');
meann = mean2(seg);
meann= meann*100;
y = round(meann);
disp(y);
run('predict.p');
pause(1);
end
```

6. Background Subtraction

Figure 4 shows the working procedure for detecting fish in a video involves separating all moving objects from the background. To achieve this, a background subtraction (BS) approach was chosen due to its simplicity, speed, and ability to handle various challenges. The BS algorithm analyzes the raw video and produces binary maps, where "0" represents background pixels and "1" represents the foreground, indicating moving objects in Figure 5. This process can be performed in real-time, with each frame being analyzed immediately after it's read. The initial frames are used to build the model, while subsequent frames are used to detect motion.

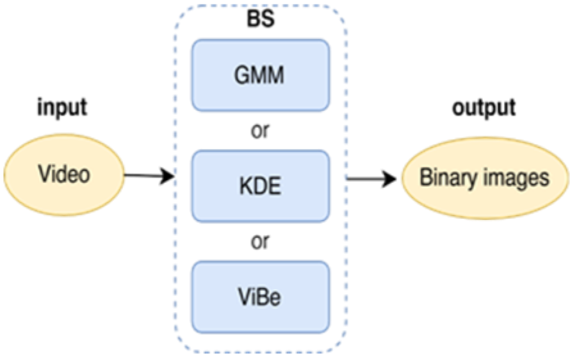

Fig 4: Background Subtraction

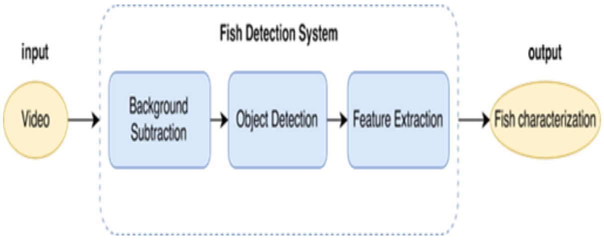

Fig 5: Fish Detection System

Several factors were taken into account when selecting the appropriate BS method, including accuracy, robustness, and ability to handle noise, moving background, illumination changes, computational speed, and ease of implementation. A comprehensive analysis of different BS methods can be found in [18, 37]. After reviewing the available information, three algorithms were selected for the task: Adaptive GMM, KDE, and Visual Background Extractor (ViBe).

7. Adaptive Gaussian Mixture Model

The first selected algorithm is based on the Gaussian Mixture Model (GMM) and has been described in [30, 38]. One of the significant advantages of this algorithm is that it represents the color of each pixel using multiple Gaussians and automatically determines the appropriate number of Gaussians. This feature makes it highly adaptable to sudden and gradual scene changes. As fish counting videos often feature changing illumination and moving objects that remain stationary for extended periods, choosing this algorithm appears to be a sensible choice.

The GMM method is a statistical approach to background subtraction, where the background image is assumed to be well-described by a statistical model due to its regular behaviour. A scene model is constructed by estimating a Probability Density Function (PDF) for each pixel, with GMMs used to model complex pixel distributions rather than a single PDF. Any pixels that do not fit the model are considered to be foreground.

The background model is denoted by $p(\vec{x}\,|BG)$, where $\vec{x}$ is a pixel value in some color space. A training set X is used to estimate the model, with $p(\vec{x}\,|X\ ,\ BG)$ representing the obtained model. At each time t, a decision is made on whether the pixel belongs to the background or foreground using Bayesian decision B is

$$p(BG|\vec{x}(t)) \quad p(\vec{x}(t)|BG)p(BG)$$

$$B = p(FG|\vec{x}(t)) = p(\vec{x}(t)|FG)p(FG)\ . \qquad (1)$$

Typically nothing is known about the foreground. So p(FG) is set equal to, p(BG) and the distribution is assumed to be uniform $p(\vec{x}(t)|FG) = cFG$.

Then the pixel is referred to the background if

$$p(\vec{x}(t)|BG) > cthr, \qquad (2)$$

where cthr = BcFG is a threshold value ion-making technique.

Since we expect that the scene may vary over time, it is essential that the training set should adjust to changes by adding or eliminating samples. So at time t the training set is XT = {x(t), ..., x(t−T)}, where T is a selected time period. The set XT and the model itself are recalculated with each new sample. As our goal is to detect the foreground, some of the samples will contain foreground pixels, which should not affect the update of the background. So in general, the proposed GMM approach with M components is described as

8. Visual Background Extractor

Algorithm 1 : ViBe

Read the first frame.

for all pixels *x* in frame **do**

d Background initializationRandomly choose *N* pixels from the neighbourhood of *x*.

Store these values $v_1, \ldots v_N$ in the pixel model (*x*).

end for

for all remaining frames **dofor all** pixels *x* **do** *d*

Foreground detection Compare current pixel value v(x) to the background Model (x).

Find the number *num* of close samples in the model.

if *num* $\#_{min}$ **then**

Classify pixel as a background.

d Background updateGet random sample from pixel model and update it with the current pixel value.

Randomly choose one neighbour and update its model.

else

Classify pixel as a foreground.

end if

end for

8.1. Optical Flow

Optical Flow provides an opportunity to measure a motion from video frames [43]. The technique is based on assumptions that pixel values do not vary significantly between consecutive frames and that pixels in one neighborhood move in a similar way. This hypothesis is called brightness constancy constrain with the pixel displacement (*dx*, *dy*) after time *dt*:

$$\boldsymbol{l(x, y, t) \approx l(x + dx, y + dy, t + dt)}, \quad \textbf{(1)}$$

By expanding with Taylor series, we get

$$\boldsymbol{l(x + dx, y + dy, t + dt) = l(x, y, t) + \frac{dl}{dx}dx + \frac{dl}{dy}dy + \frac{dl}{dt}dt + 0^2}, \quad \textbf{(2)}$$

where O^2 are the second and higher order terms, which are neglected. From this equation,it follows that:

$$\boldsymbol{\frac{dl}{dx}dx + \frac{dl}{dy}dy + \frac{dl}{dt}dt = 0}, \textbf{(3)}$$

which can be transformed into:

$$\boldsymbol{\frac{dl}{dx}\frac{dx}{dt} + \frac{dl}{dy}\frac{dy}{dt} + \frac{dl}{dt}\frac{dt}{dt} = 0}$$

$$\boldsymbol{\frac{dl}{dx}V_x + \frac{dl}{dy}V_y + \frac{dl}{dt} = 0}, \qquad \textit{(4)}$$

where V_x and V_y are the components of the image velocity.

Advantages:

- High accuracy of fish detection
- Less human intervention

Disadvantages:

One of the primary limitations of this project is that the implementation of the camera underwater is not feasible due to the high cost of the panoramic camera.

9. Ensemble Method Algorithm

Figure 6 shows the working procedure for Artificial Neural Networks (ANN) is modeled after Biological Neural Networks that form the structure of the human brain. Just like the neurons in the human brain that are connected to one another, ANNs also consist of neurons, referred to as nodes, that are connected to each other in multiple layers of the network. The design of ANNs involves programming computers to emulate the behavior of interconnected brain cells.

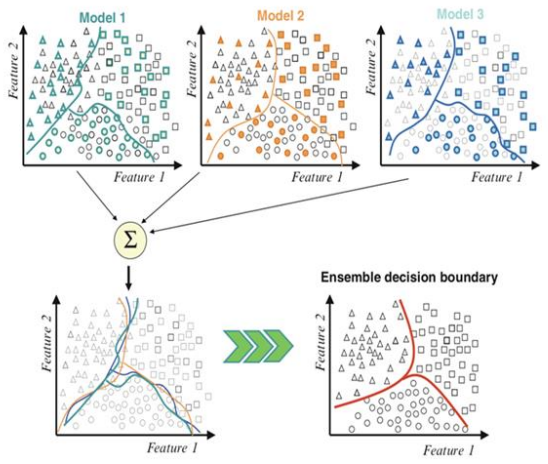

Fig 6: Artificial Neural Network

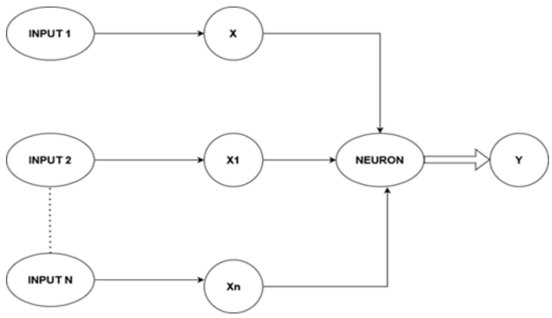

Fig 7: process of using an ensemble technique

Figure 7 shows the general process of using an ensemble technique to improve the accuracy of a predictive model.

The first step is to train multiple base models on the same dataset. These models can be of different types or variations of the same type, and each one generates predictions for a given input.

The second step is to combine the predictions of the base models to create an ensemble model. This can be done through various methods, such as averaging the predictions or using a weighted average based on the performance of each base model.

The final step is to use the ensemble model to make predictions on new data. The ensemble model combines the predictions of the base models, which leads to improved accuracy and robustness compared to any individual base model.

Overall, ensemble techniques are commonly used in machine learning to increase the performance of models and are particularly effective when used in combination with diverse base models.

Ensemble decision boundary refers to the decision boundary of an ensemble model, which is a combination of multiple base models. In binary classification problems, the decision boundary separates the two classes and determines which class an input belongs to fish features.

Ensemble methods combine the predictions of multiple base models to create an ensemble model that can make more accurate and robust predictions than any individual model. The decision boundary of an ensemble model is determined by the combination of the decision boundaries of the base models.

The decision boundary of an ensemble model can be more complex and flexible than that of an individual base model. This is because the ensemble model can capture different patterns and structures in the data that might be missed by any single base model.

Overall, the decision boundary of an ensemble model is a crucial aspect of its performance and can be used to evaluate the model's accuracy and robustness. Ensemble methods, such as random forests and gradient boosting, have become popular in machine learning due to their ability to create powerful and flexible decision boundaries for complex classification problems.

Step 4: Plotting the graph for validation and accuracy loss:

To plot the graph for validation and accuracy loss in R-CNN, you can use the training logs generated during the training process which is mentioned in Figure 8, 9.

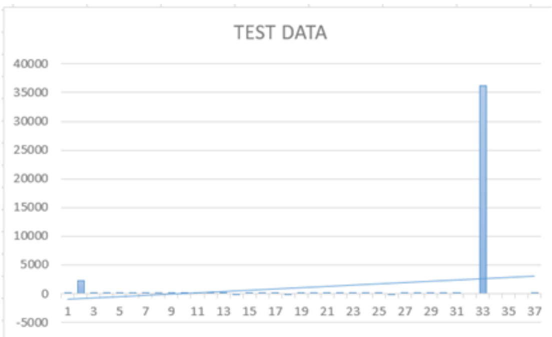

Fig 8: Test Data

Fig 9: Training Data

These graphs indicate the accuracy rate when compared to the test data and training data.

Step 5: Results

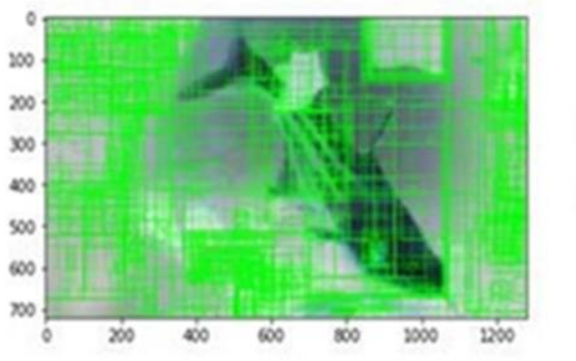

Fig 9: Region Selection

Figure 9 shows the Region Selection of the captured fish; Figure 10 shows the Region Selection of the captured fish Recognition Result and Figure 11and 12 shows the real-time dataset of the given research.

Fig 10: Fish Recognition Result

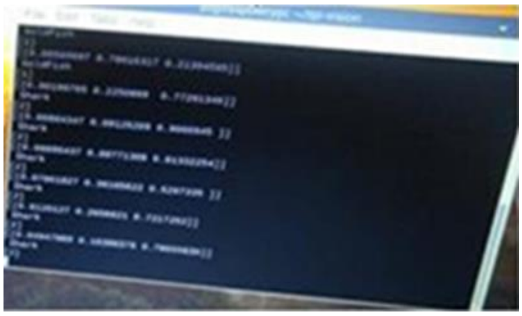

Fig 11: Dataset of Fishes

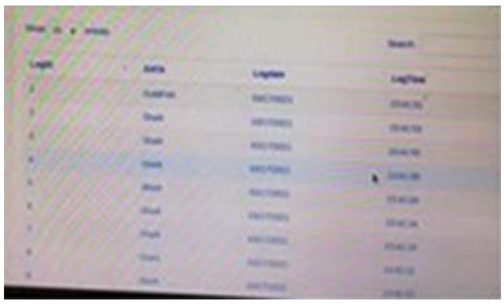

Fig 12: Dataset Stored In IOT Cloud

10. Conclusion:

The datasets were pre-processed and then analysed using deep learning and OpenCV tools to detect and recognize different fish species. The accuracy of the algorithm was demonstrated by displaying the detected fishes and their respective species, as well as the regions where they were found in the image dataset. The resulting datasets were then stored in the IoT cloud. The algorithm achieved an accuracy of 85%. This fish recognition and detection system has potential applications in various fields, such as identifying fish species in water and commercial purposes. To further improve the accuracy of the system in real-time, high-end underwater cameras can be utilized.

Declarations

Competing Interests

The authors declare that they have no conflict of interest.

Authors' Contributions

- The Author **R Vinston Raja** has done his contribution by writing and Defining the problem statement.
- Adithya.V & Kirran P L has done Data analysis
- Flintoff A Hollioake has done Implementation
- Krishna Kumar N has collected data set

Funding

This research work is self-funded.

Availability of Data and Materials

The data shall be made available on request.

Ethical and informed consent for data used

Data used ethically

References:

[1] Jung-Hua Wang, Shi-Kai Lee, Yi-Chung Lai, "Anomalous Behaviours Detection for Underwater Fish Using AI Techniques", IEEE Access, vol. 8, pp 224372-224382, Dec 2020.

[2] Kristian Muri Knausgard, Arne Wilkund , Kim Tallaksen Halvorsen, "Temperate Fish Detection and Classification: a Deep Learning based Approach", ResearchGate ,May 2020

[3] B.Vikram Deep, Ratnakar Dash, "Underwater Fish Species Recognition Using Deep Learning Techniques"-at 6th International Conference on SPIN,Mar.2019.

[4] Lin Meng, Takuma Hirayama, Shigeru Oyanagi,"Underwater-Drone with Panoramic Camera for Automatic Fish Recognition Based on Deep Learning"- at IEEE Access vol.6,pp 17880-17886 Mar 2018 .

[5] Nour Eldeen,Mohamed Hamed N.Taha,Aboul Ella Hassanien,"Aquarium Family Fish Species Identification System Using Deep Neural Networks"-at International Conference on Advanced Intelligent Systems and Informatics pp 347-356,Aug. 2018.

[6] Matzner, S., Hull, R. E., Harker-Klimes, G., & Cullinan, V. I. ,"Studying fish near ocean energy devices using underwater video." IEEE, pp. 1-7 Sep 2017.

[7] Lee.G, Y. W. Tai, and J. Kim, ``ELD-net: An efficient deep learning architecture for accurate saliency,"in IEEE Transactions on Pattern Analysis Machine Intelligence, Vol. 40, Issue detection: 7, pp. 1_13, July 2017.

[8] M.Dumiak,"Lice-hunting underwater drone protects salmon" -at IEEE Spectrum,vol.54, no. 4, pp.9–10, Apr. 2017.

[9] P. Zhuang, L. Xing, Y. Liu, S. Guo, and Y. Qiao, "Marine animal detection and recognition with advanced deep learning models," in Proc. Conf. Labs Eval. Forum (CLEF), pp. 1-9, 2017.

[10] Nhat D. M. Nguyen, Kien N. Huynh, Nhan.N.Vo, Tuan Van Pham,"Fish Detection and Movement Tracking", International Conference on Advanced Technologies for Communications (ATC) ,2015

[11] A.Rova, G.Mori, L.Dill, "One fish two butterfish trumpeter: recognizing fish in underwater video"-IAPR Conference on Machine Vision Applications, pp.404-407, May 2007.

[12] F.Storbeck and B.Daan,"Fish species Recognition using computer vision and a Neural network", Fisheries Research, Vol.51,pp.11-15, Apr 2001.

[13] Reddy, A. ., & Waheeb , M. Q. . (2022). Enhanced Pre-Processing Based Cardiac Valve Block Detection Using Deep Learning Architectures. Research Journal of Computer Systems and Engineering, 3(1), 84–89. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/47

[14] Gandhi, L. ., Rishi, R. ., & Sharma, S. . (2023). An Efficient and Robust Tuple Timestamp Hybrid Historical Relational Data Model. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 01–10. https://doi.org/10.17762/ijritcc.v11i3.6193