

Fuel Spill Monitoring for Fishing Smack using Raspberry Pi

Manisha N L, Silpa P A



Abstract: Fuel spill monitoring for fishing smacks is a live fuel leak detector that can alert the vessel's crew and captain about the leakage by using a web camera connected to a Raspberry Pi. The fuel spill was resolved using the Convolution Neural Network (CNN). Also, the Coast Guard and the Environmental Protection Agency were informed about the location of the oil discharge through telegrams. Here, a picture of the spill, as well as its latitude and longitude, a live Google map location, and a no spill picture with a GPS location whenever the spillage stops, will be shared. As a result, the team could take immediate action without delay. This spill detection system is linked to an accident detection system. Hence, we can safeguard fishing vessels and marine activities without any harm to human kind, as well as to the living beings in the sea.

Keywords: Convolution Neural Network, Machine Learning, Oil Spill, Raspberry Pi

I. INTRODUCTION

Marine accidents [1] and water pollution are common nowadays. Some common reasons for maritime collision cases are lack of sleep, inexperience, long voyages, reckless behavior, poor decision making, due to the weather, etc... And major causes of ocean pollution [2] are sewage, toxic chemicals from industries, land runoff, large scale oil spills, ocean mining, littering, effect of toxic wastes on marine animals, disruption to the cycle of coral reefs, [12] etc... When petroleum pours onto the surface of a massive water body, an oil spill occurs. In the 1960s, oil spillage in the seas became a serious environmental issue, owing to increased petroleum drilling and development on continental shelves, as well as the usage of supertankers capable of transporting more than 500,000 metric tons of oil. Because of strict maritime and environmental laws, spectacular oil spills from wrecked or damaged supertankers are now unusual. Despite this, each year, hundreds of minor and significant oil spills involving well discharges and tanker operations are reported, with the total amount of oil discharged into the world's oceans topping one million metric tons. Manufacturers and people releasing used gasoline solvents and crankcase lubricants

unintentionally or negligently exacerbate the overall environmental problem. These sources, when combined with natural seepage from the ocean floor, add 3.5 million to 6 million metric tons of oil to the world's waterways each year. Oil spills [3], in particular, which are largely caused by human action, negligence, or faulty equipment, have the potential to cause significant damage to the deep ocean as well as the fishing sector. Accidental or deliberate operational discharges and spills of oil from ships, notably tankers, offshore platforms, and pipelines, are the most visible and visible cause of oil pollution in the aquatic region. According to biochemical studies, those who are exposed to oil and gas leaks suffer irreversible harm. Cardiovascular damage, autoimmune disease, lowered immunity, cancer risk, fertility problems, and higher levels of some toxics are just a few of the consequences (hydrocarbons and heavy metals). Fish suffocate, becoming entangled in the feathers of birds and mammals, and blocking photosynthetic light. As a result of the spill, plants in the water were also injured. Toxic metals can also find their way into the marine food chain. In marine life, this can cause changes in tissue matter, biochemistry, behavior, reproduction, and growth suppression. In addition, many animal feeds contain a high percentage of organic or fish hydrolyses. Marine poisons are routinely transferred this way to land animals, eventually manifesting in meat and dairy products. As a result, it's vital to maintain a close eye on bodies of water, particularly those in the sea.

In addition, due to a lack of a reliable communication network, fishermen are unable to get real-time services such as collision avoidance alerts [12], maritime border crossing, obstacle alerts, live market updates, and so on. During the poll, the fishermen identified collisions between fishing vessels [4] and large ships as the most pressing concern. Ships use radar to detect obstacles, but they occasionally miss small unlighted fishing vessels, especially in the dark, resulting in a tragedy. The fishermen have no way of knowing when the ships will arrive. We could restrict the after-effect impacts of accidents to some extent by adopting a lot better system.

Excessive marine pollution and vessel/boat crashes endanger aquatic wildlife and humans, causing a host of health issues, deaths, and other catastrophes. As an outcome, the importance of marine safety is growing.

In this paper, a real-time system that detects oil spills at sea and automates object identification of the incoming obstacle, as well as a collision reporting system for fishing vessels [12], will be introduced in this study, with a focus on lower costs and improved equipment communication, with the safety of human lives and the environment as a top priority.

Manuscript received on June 25, 2021.

Revised Manuscript received on July 02, 2021.

Manuscript published on July 30, 2021.

* Correspondence Author

Manisha N L*, Student, Department of Electronics and Communication Engineering, Sahrdaya College of Engineering and Technology, Kodakara, India. Email: manishalalu@gmail.com

Dr. Silpa P A, Assistant Professor, Department of Electronics and Communication Engineering, Sahrdaya College of Engineering and Technology, Kodakara, India. Email: silpapa@sahrdaya.ac.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

II. SYSTEM DESCRIPTION

Fig.1 shows the block diagram of the proposed system for fishing vessels. The basic work of the system is as follows: When the power is in an ON state, all components are supplied with the required amount of current and, especially, the LCD displays a message stating "ROBOTIC SECURITY" [12], which makes sure the robotic system is ready for its action. In the first case, when an obstacle is found within a specified limit of the LIDAR [9] sensor TF-Luna, the buzzer will be activated and a display message on the LCD stating "OBSTACLE DETECTED" will be shown. These allow the crew members and captain on the bridge to take immediate precautions. The second case is one of the risky situations in which a boat/ship collides. This could be found out with the help of a vibration sensor [10], which will be activated at the accident time. As a result, when the sensor is triggered, the buzzer sounds and asks for aid, and a nearby boat/ship may come for assistance at any time of day or night, while the word "COLLISION DETECTED" is displayed on the LCD. As a result, immediate assistance could be sought prior to the arrival of the rescue team. The location of the boat/ship is also shared, as well as details such as the ship/boat's name, IMO number, Google map, latitude and longitude of the accident occurrence, indicating a collision occurred. A message will be sent to the nearby coastal area service sector named International Search and Rescue Advisory Group (INSARAG) via a telegram [6] group chat named "EMERGENCY SERVICE DEPARTMENT," which is able to communicate the seriousness of the incident.

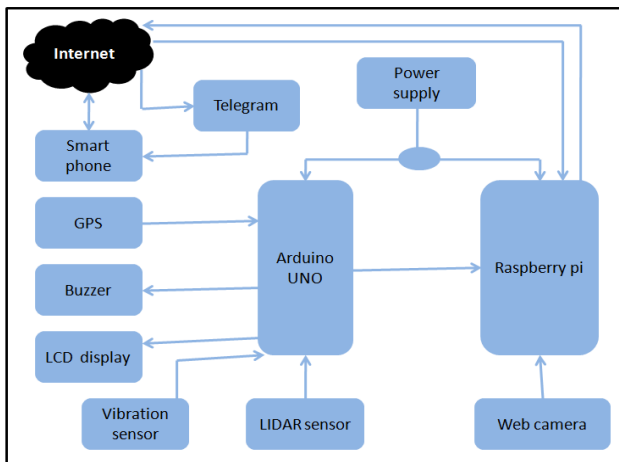


Fig. 1. Block diagram of marine monitoring system for fishing vessels

The oil detection system is also simultaneously working, which will be watched via the web camera attached to the Raspberry Pi that is coupled with the accident detection system. Once situations like oil is spilt while being transported, or as a result of collisions between boats/ships or as a result of a leak from its own fuel tank happens, an alert message for the captain, crew members, the coastal guard, Environmental Protection Agency team gets notified with the help of a Service Bot which is being shared in the telegram group chat named "EMERGENCY SERVICE DEPARTMENT". Thus, quick measures could be taken regarding this issue.

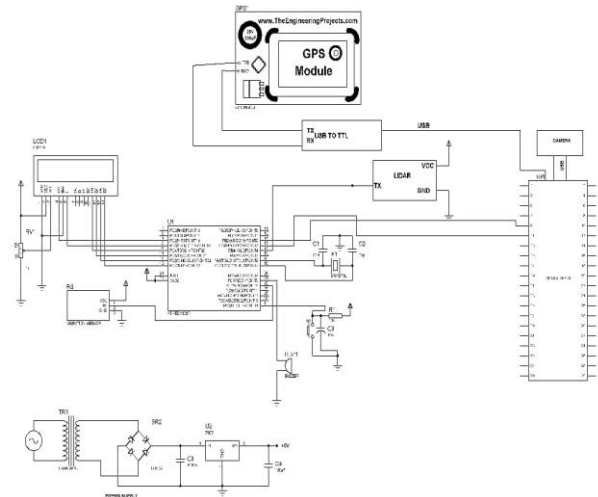


Fig. 2. Circuit diagram for proposed system

The Fig.2 presents the hardware connection for the embedded system. A 230V alternating current (AC) is given as an input to the step-down transformer and an output voltage of 9V will be given to the bridge rectifier as an input that converts AC to direct current (DC). A voltage regulator named 7805 is used that provides 5V as an output to the bridge rectifier along with 9V from the transformer. The current of up to 12V from the bridge rectifier will be supplied to the Arduino UNO and the LCD display. The LIDAR sensor TF-Luna, buzzer and vibration sensor LM393 get 5V from the Arduino board. For message passing, the (transmitter) TX pin of the Raspberry Pi (Rpi) will be connected to the RX pin (receiver) of the GPS module and the RX pin of the Rpi will be attached to the TX pin of the GPS module. A 5-meter voltage will be provided by the Rpi to the GPS module. Also, the transmitter pin of LIDAR TF-Luna is allied to the receiver pin of the Arduino UNO board and the receiver pin of LIDAR TF-Luna is infused with the TX pin of the Arduino board. In the case of the 16 x 2 LCD display, the GPIO (General Purpose Input/output) pins 8, 9, 10, 11, 12, 13 (8 for Register Select, 9 for Enable, from 10 to 13 to Data pin 4 to 7) will be linked with an Arduino UNO board. This is the circuit diagram's primary explanation.

III. METHODOLOGY

A. Oil Spill Detection

To carry out the machine learning process, deep learning, a subtype of machine learning, uses artificial neural networks (ANN) at the level of a hierarchy. Artificial neural networks are built similarly to the human brain, with neuron nodes linked in a web-like structure. The hierarchical function of deep learning systems enables machines to analyze data in a nonlinear fashion, whereas traditional programmers construct data analysis using linear method. Machine learning consists of seven steps, which are as follows:



• **Data Gathering**

We will use approximately 5130 (no spill: 2545, oil spill: 2585) images for data collection, with 4107 (no spill: 2036, oil spill: 2071) training images and 1023 (no spill: 509, oil spill: 514) testing images in two classes, i.e., no spill and an oil spill. The images were collected using video of the oil spill, from the initial drop to the final spread of the oil. A total of 5 to 8 video combinations of 200 frames each were captured. A testing tray was taken, and each corner and the middle portion, almost the entire tray, were recorded with different oil spill quantities and lights. Then these videos were converted into separate images and stored for the next stage.

• **Data Preparation**

Here, we have compared two datasets. The evaluation results of the dataset are shown in Table I. When a few test samples were analyzed, the prediction output was found to be false predictions because of over fitting, a situation where any given model performs too well on the training data but performance drops significantly over the test set. When compared to the first dataset that contained 4072 training images and 1018 testing images from two classes, the second dataset that contained 4107 training images and 1023 testing images from two classes performed significantly better. The number of epochs used was 10 with a batch size of 3.

Table- I: Dataset Comparison & Evaluations

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy	ETA in seconds	Batch size
10	7.1495e-06	1.0000	0.0104	0.9980	1793s	3
10	0.0359	0.9939	0.0228	0.9980	1756s	3

The difference in accuracy and loss value after major changes in batch size and epochs is shown in Table II. When the batch size was increased, the results appeared to be better, but the validation accuracy appeared to be in poor condition. The ratio used for splitting the dataset into training and validation is 8:2.

Table- II: Epochs v/s Batch size Evaluation

Epochs	Batch Size	Loss	Accuracy	Validation Loss	Validation Accuracy
10	3	0.0359	0.9939	0.0228	0.9980
100	32	5.2497 e-06	1.0000	0.4837	0.9488

• **Choosing a model**

In my model's architecture, the convolutional layer is followed by ReLu, then a max pooling layer. This pattern is repeated twice, and then a flattened layer and a dense layer with ReLu and sigmoid are added. The number of nodes in each communication layer is as follows: 16, 32 and 64 and the input images are 2D matrices. Then comes the kernel size, which is defined as 3x3, which is nothing but a filter used to extract features from the images. The input shapes of images are 200x200 pixels in an RGB color space. Here, a Max pooling layer is used that takes out the maximum parameters from a pool and the rest is dropped out. We know that the pooling operation has a smaller size than the feature map, i.e. it is almost always 2x2 pixels with a stride of 2 pixels. Then the flattened layer that serves as a connection between the convolution and the dense layers. The flattening is converting

the data into a 1D array and then inputting it to the next layer. We flatten the output of the convolution layer to create a single long feature vector and it is connected to the final classification model, which is called the fully connected layer. The final layer, or the dense layer, which is used for our output layer, will have two nodes, one for each possible outcome (0, 1). The activation is a 'sigmoid' that makes the output sum up to 1, so the output can be interpreted as a probability. The model will make the prediction based on which option has the highest probability. Next, we compiled the model. The parameters used include optimizers, loss and metrics. The Adam optimizer controls the learning rate throughout training. The learning rate determines how fast the optimal weights for the model are calculated. For the loss function, we use binary-cross entropy that computes the cross entropy loss between true labels and predicted labels. To make things even easier, in order to interpret the accuracy score on the validation set, we will use the accuracy metric to see the accuracy score on the validation set when we train the model.

• **Training the Model**

To train the model, we use the fit () function, where we specify parameters like training data, validation data, and the number of epochs. The epoch's number indicates the number of times the model will cycle through the data. After a certain point, the model will stop improving during each epoch. Table III shows a model summary of the CNN model created.

Table- III: CNN Model Summaries

MODEL: SEQUENTIAL		
LAYER (TYPE)	OUTPUT SHAPE	PARAMETERS
conv2d (Conv2D)	(None, 198, 198, 16)	448
max_pooling2d (MaxPooling2D)	(None, 99, 99, 16)	0
Conv2d_1 (Conv2D)	(None, 97, 97, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 48, 48, 32)	0
Conv2d_2 (Conv2D)	(None, 46, 46, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 23, 23, 64)	0
flatten (Flatten)	(None, 33856)	0
dense (Dense)	(None, 512)	17334784
Dense_1 (Dense)	(None, 1)	153

• **Evaluation of the Model**

Here we test the model against previously unseen data. A good technique to check if the model has been appropriately trained is to visualize the training loss vs. validation loss or training accuracy vs. validation accuracy over a number of epochs. This is necessary to ensure that the model is not undertrained or over trained to the point that it begins to memorize the training data, reducing its capacity to predict effectively. Fig. 3 and Fig. 4 plot the accuracy and loss graph [8] for the training dataset.



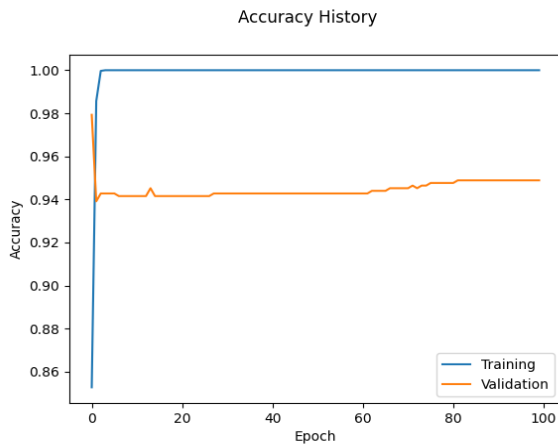


Fig. 3. Accuracy Graph for training dataset

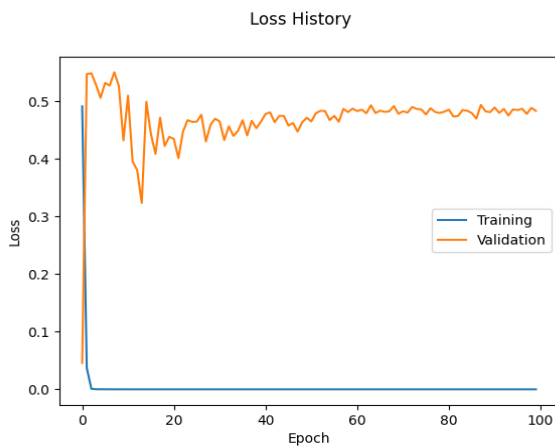


Fig. 4.2 Loss Graph for training dataset

Fig. 5 depicts the confusion matrix [7], which is an $N \times N$ matrix used to evaluate the performance of a classification model, where N is the number of target classes. The matrix compares the actual goal values to the machine learning model's predictions. This provides us with a comprehensive picture of how well our classification model is working and the types of errors it makes. For a binary classification problem, we would have a 2×2 matrix. There are two possible values for the target variable: positive or negative. The target variable's real values are represented in the columns. The rows represent the target variable's expected values. True positive: The anticipated value corresponds to the real value. The model anticipated a positive value, and the actual result was positive. Here, we have 509 no spill class data points that were all correctly classified by the model. True negative: The anticipated value corresponds to the real value. The model anticipated a negative value, and the actual value was negative. Here, we have 465 spill class data points that were correctly classified by the model. False positive: The expected value was incorrectly predicted. Although the actual number was negative, the model projected that it would be positive. Here, we have 0 spill class data points that were incorrectly classified as belonging to the no spill class by the model. False negative: The expected value was incorrectly predicted. Although the actual number was positive, the model predicted that it would be negative. Here, we have 49 no spill class data points that were incorrectly classified as belonging to the spill class by the model.

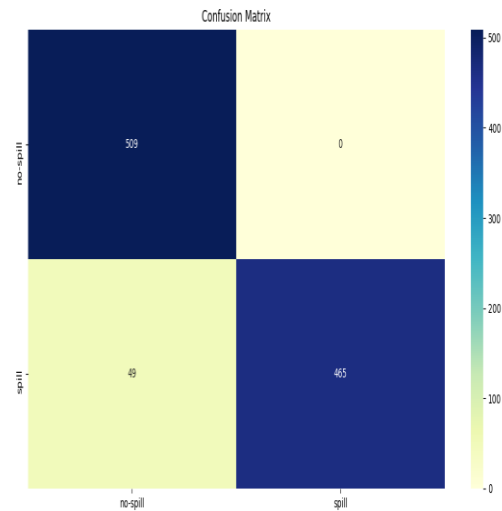


Fig. 5. Confusion Matrix

• **Parameter Tuning**

This improves performance. Each model has its own set of parameters that need to be tuned to get optimal output. For every model, our goal is to minimize the errors or, say, to have predictions as close as possible to the actual values. A threshold value of 0.8 indicates that if the value is greater than 0.8, there is no spill. The value was determined by analyzing the confusion matrix and taking into account the reduction of false value predictions. Ideally, closer to zero means no spill class, or if closer to one, it means a spill class. But there is no strict margin. Hence, using the trial and error method, the value has been obtained.

• **Make Predictions**

The new set of testing data is used to determine the trained model's prediction. The generated model will be compared with each input image and predictions will be made.

IV. RESULTS

Fig. 6 shows the message received in the telegram group chat for emergency cases. The service bot automatically sends messages regarding the alerts. Based on that, each member gets to know what is happening to the specified fishing vessel. When the oil is detected, the bot immediately sends a message to the people in charge. Fig. 7 shows the no spill message after the identification of spillage stops. The environmentalists and members of the group learn about the location of the spilled area. Figure 8 depicts the message sent to the rescue team and the nearby coast guard about the impending disaster. The details shared regarding the fishing vessel will help to identify the boat, its crew members, etc.



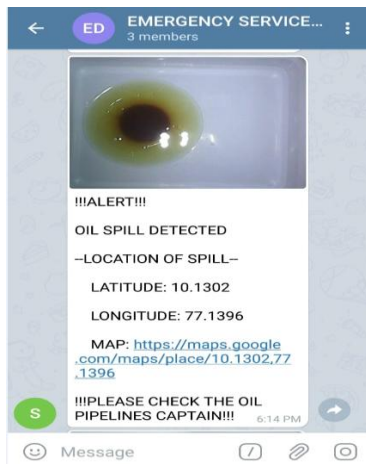


Fig. 6. An oil spill notification will be sent to the telegram group chat "EMERGENCY SERVICE DEPARTMENT" by the service Bot after the leak.



Fig. 7. No Oil Spill messages on the telegram "EMERGENCY SERVICE DEPARTMENT" group chat after the stoppage of the spill by the service Bot.

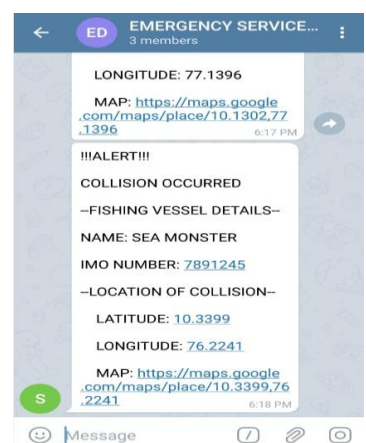


Fig. 8. Collision alert after the accident to the telegram group chat "EMERGENCY SERVICE DEPARTMENT".

V. CONCLUSION AND FUTURE PLAN

A real-time self-monitoring embedded system with an early oil spill detection and collision avoidance alert system that has faster communication while staying within a reasonable budget was developed [12]. Thus the rescue team will be alerted and action can be taken immediately. Death rates due

to collisions have been reduced. Personal information sent via Telegram chat during collisions aids in quickly informing the nearby coastal region's rescue team, as well as their relatives, about the accident. Cleaning costs can also be reduced. Pollution can be reduced and can save living organisms within water bodies, which helps to balance the ecosystem. Diseases caused by eating sea food could also be avoided to a certain extent. The level of leakage of oil could be found out regarding ships/boats, which helps in the self-monitoring 24/7 system for spillage identification. A validation accuracy of 94.88% was obtained, with no false positives.

The following are some of the system's drawbacks: A strong internet connection is required to send a message that will help to spot oil patches quickly & accurately by reducing pollution. The accuracy of prediction of oil spill seemed to be slow due to raspberry pi processor. The obtained false negative value is around 49. The strength of the dataset could be broadened.

In future work, the quantity of leakage from pipelines could be measured based on the time at which no spill message is received after the spillage has stopped. Also, the establishment of a database [11] for private fishing vessels in which received messages could be recorded. Then, a better processor could be used for larger dataset handling. Also, an additional safety service Bot could be added to the specified group chat, assisting in the simple connection of activities.

REFERENCES

1. Huanxin Wang, Zhengjiang Liu, Xinjian Wang, Tony Graham, Jin Wang, "An analysis of factors affecting the severity of marine accidents", Reliability Engineering & System Safety, Volume 210, June 2021, 107513.
2. Altin, A., Ozolcer, I. H, Yildirim, Y, (2009) "Water pollution in the southern coastal region of the black sea", Fresenius Environmental Bulletin, 18(11A), 2170-2180.
3. Baiyu Zhang, Ethan J. Matchinski, Bing Chen, Xudong Ye, Liang Jing, Kenneth Lee, "Marine Oil Spills—Oil Pollution, Sources and Effects", World Seas: an Environmental Evaluation (Second Edition), Volume III: Ecological Issues and Environmental Impacts 2019, Pages 391-406.
4. Ugurlu F, Yildiz S, Boran M, Ugurlu O, Wang J, "Analysis of fishing vessel accidents with Bayesian network and Chi-square methods", Ocean Engineering, Volume 198, 15 February 2020, 106956.
5. Ronald Mutegeki; Dong Seog Han, "A CNN-LSTM Approach to Human Activity Recognition", 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIC), IEEE Conference.
6. Choirul Huda; Fitra Abdurrachman Bachtiar; Ahmad Afif Supianto, "Reporting Sleepy Driver into Channel Telegram via Telegram Bot", 2019 International Conference on Sustainable Information Engineering and Technology (SIET), IEEE Conference.
7. O Caelen, "A Bayesian interpretation of the confusion matrix", Annals of Mathematics and Artificial Intelligence, 2017 – Springer.
8. Tinghuai Maa,b, Wenye Shaoa, Yongsheng Haoa, Jie Caoc, "Graph Classification Based on Graph Set Reconstruction and Graph Kernel Feature Reduction", Neurocomputing 00 (2018) 1–21, Elsevier.
9. Riccardo Polvara, Sanjay Sharma, Jian Wan, Andrew Manning, Robert Sutton, "Obstacle Avoidance Approaches for Autonomous Navigation of Unmanned Surface Vehicles", The Journal of Navigation, Volume 71, Issue 1, January 2018, pp. 241 – 256, DOI: https://doi.org/10.1017/S0373463317000753.
10. Dr. R. Dhaya, Dr. R. Kanthavel, "A Wireless Collision Detection on Transmission Poles through IoT Technology", Journal of trends in Computer Science and Smart technology (TCSST) (2020) Vol.02/ No. 03 Pages: 165-172 https://www.ijournals.com/tcsst/ DOI:



<https://doi.org/10.36548/jtcsst.2020.3.006>.

11. R Parluka, A Pratama, "The Online Test application uses Telegram Bots Version 1.0", Journal of Physics: Conference Series, 2020 - iopscience.iop.org.
12. Manisha N L, Silpa P A, "Automatic Obstacle Identification and Collision Reporting System for Fishing Vessels", International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE), Vol 8, Issue 5, May 2021[Online]. Available: <http://ijerece.com/abstract.php?id=14520&issue=Issue5>.

AUTHORS PROFILE



Manisha N L, M. tech student specialized in Embedded Systems at Sahrdaya College of Engineering And Technology, Kodakara under APJ Abdul Kalam Technological University. She has B. tech degree in Computer Science and Engineering under APJ Abdul Kalam Technological University.



Dr. Silpa P A, Assistant professor in Electronics and Communication Engineering at Sahrdaya College of Engineering And Technology. She completed her Master's degree in VLSI and embedded system from Rajagiri School of Engineering & Technology. Her research interests are Nanotechnology, MEMS, VLSI and Biosensors.