Amalgamation of Wireless Sensor Network and Artificial Intelligence for Underwater Security

Mr. Animesh Kalita

B.Tech Computer Science and Engineering, Assam Don Bosco University, Azara Guwahati

Abstract
As mankind have explored much more about the space technology but there is very less concerned is given to underwater research. So, this paper is about underwater security with amalgamation of wireless sensor network and artificial intelligence. Here, we have shown a simulated project to detect unidentified object underwater.

Keywords: ns-simulator, machine learning, image processing, acoustic signal

1. INTRODUCTION

In the current era, the world is covered with 80% of water so, the communication in underwater wireless sensor networks is essential for human to avoid electromagnetic fields. In underwater wireless sensor networks security is the major concern to do the communication in secure way. To improve the security and authentication in underwater, artificial intelligence will be used to understand the components, or what sort of object will be there to cause any harm and wireless sensor network will be used to send the message to the base which will help to alert others.

As there are still research going on underwater communication, our motive is to help underwater military basement from enemies. Our work will help to detect what type of object is following them, as we all know that in radar we can only detect that some object is following towards the basement but the object type cannot be detected. So, artificial intelligence will help in figuring out the object type and an alert message will be send to the basement by underwater wireless sensor network (UWSN).

In [1] this research paper working on underwater wireless sensor network for routing difficulties in underwater as nodes are dynamic and stability is low. So, decentralized communication is used in underwater wireless sensor network (UWSN) to figure out theinterfaces between attackers and defenders and the game theory approach is very useful in decentralized communications.

In [14], the paper is on acoustic channel which is highly variant because of unique challenges example- narrow bandwidth, long propagation delays, speed of sound, etc. All these challenges creates problem regarding media access control protocols which have two categories-scheduled protocols and contention- based protocols.

In [15], the paper describes about wireless sensor network where large number of sensor nodes are deployed and networked to monitor or survey targeted area, such that interested data can be sensed, processed, stored and collected. Through WSNs, we can build bridge between the physical world and the cyber space, which composes the foundation for developing new smart applications

2. LITERATURE STUDY OF EXISTING SYSTEM

In [4] the paper provides with three research works towards intelligent and secure UWA communication networks exploiting machine learning and signal processing. A RL-based adaptive transmission framework was proposed for point-to-point UWA communication networks. A RL-based adaptive trajectory planning of multiple AUVs for water parameter field reconstruction was developed in the under-ice environment. A signal alignment method was designed for underwater coordinated multipoint (Comp) transmissions to secure a legitimate user against eavesdropping. The transmission schedule, the active DAE, and the transmission power were determined to degrade the decoding performance of the eavesdropper. The machine learning techniques offer a new perspective to rethink about research problems in UWA communication networks. In this dissertation, the RL framework wasexploited to study the sequential decision-making problem for underwater applications which operatein the long term. It showed that applying machine learning techniques to UWA communication networks significantly improved their system performance. About UWA network security, due to large sound prorogation delay in water, conventional security mechanisms designed for terrestrial radio networks cannot be applied directly to UWA networks, and many algorithms were proposed to overcome that delay challenge.

In [5] the paper focuses on analyzing a measured sea trial dataset by using a rule-based strategy (i.e., three- dimensional analysis, modulation-wise analysis, and a fixed-SNR strategy) to find the suitable link adaptation procedure depending on the channel quality. Due to non-reversibility limitation of rule- based strategy, the work further extends to use machine learning (ML) algorithms to classify the MCS levels by investigating the channel characteristics. Boosted regression tree, from among the four ML algorithms we adopted for the analysis, shows formidable accuracy of 99.97% in classifying MCS levels. This ensemble of trees learns from the uplink data of the buoy and the base station and relates the MCS levels to channel metrics and signal characteristics especially subject to SNR (Signal to Noise Ratio) and BER (Bit Error Rate) constraints.

In [6] the paper provides with a combination of max-RGB method and shades of grey method is applied to achieve the enhancement of underwater vision, and then a CNN (Convolutional Neutral Network) method for solving the weakly illuminated problem for underwater images is proposed to train the mapping relationship to obtain the illumination map. After the image processing, a deep CNN method is proposed to perform the underwater detection and classification, according to the characteristics of underwater vision, two improved schemes are applied to modify the deep CNN structure. In the first scheme, a $1 * 1$ convolution kernel is used on the $26 * 26$ feature map, and then a down sampling layer is added to resize the output to equal $13 * 13$. In the second scheme, a down sampling layer is added firstly, and then the convolution layer is inserted in the network, the result is combined with the last output to achieve the detection. Through comparison with the Fast RCNN,Faster RCNN, and the original YOLO V3, scheme 2 is verified to be better in detecting underwater objects. The detection speed is about 50 FPS (Frames per Second), and MAP (mean Average Precision) is about 90%. The program is applied in an underwater robot; the real-time detection results show that the detection and classification are accurate and fast enough to assist the robot to achieve underwater working operation.

In [7] the author provides with underwater acoustics sensing networks (UWASN) have the potential and ability to explore the ocean, underwater acoustics networks can facilitate us in many applications. Underwater acoustic (UWA) channels have large temporal variations, abundance of transmission paths, and wide property in nature. For underwater wireless communication system, acoustics waves area used as the primary carrier due to their relatively low absorption in underwater environment, the complexity of underwater acoustic transmission medium and low propagation speed of sound in water, regarded as the most challenging task for communication. Along with this challenge a wide range of underwater exploration and applications have emerged. The PAPR reduction techniques which are mainly considered in underwater acoustic channels are coding technique, probability technique, and signal predistortion technique. The characteristics of OFDM signal is stated with the system model in underwater acoustic OFDM communication. The PAPR lowering strategies and PTS method is described and the power efficiency thorough partial transmit sequence is derived. MATLAB simulation and numerical results are analyzed. Partial Transmit sequence method gives the better outcome in terms of PAPR reduction performance.

In [8], the paper focuses on security in underwater wireless communication networks which are vulnerable to malicious attacks due to high bit error rates, large and variable propagation delays and low bandwidth of acoustic channels. In this paper a complete survey for security for UWCN's is presented. The paper explains the particular characteristics of UWCN's compared with their ground based counterparts. Here, the author also introduced about double attacks and counter- measures square measure. At last, the paper provides the analysis challenges involving secure time synchronization, localization and routing square measure summarized.

In [9], the paper is presented on underwater drone with panoramic camera for automatic fish recognition. The drone is equipped with fisheye lenses and with the function of a 360 degree panoramic camera for taking images by using an image generation algorithm. The design is based on the open source hardware and will be shared as an open-source for contributing to the innovation of manufacturing including drone. The drone focuses on fish recognition for investigating fish species. Fish recognition is based on deep learning. In the paper the experimental results show that the function of the underwater drone achieved at diving in the leak automatically. Fish recognition achieved 87% accuracy by deep learning.

In [10], the authors mentioned about autonomous underwater monitoring system in which computer vision cloud services is used for detecting life on the seabed. Here author mentioned about an AUV model system designed to overcome latency challenges in the supervision and tracking process by using edge computing in an IoT

gateway. The IoT gateway is used to connect the AUV control system to the internet. It highlighted the cloud and edge architecture performances and indicated a hybrid cloud/edge architecture to ensure a real-time control loop for better latency .Here the author proposed an algorithm to autonomously track the target species by integrating the object detection system into the AUV without human intervention.

In [11], the author mentioned about the security challenges and applications for UWSN .Here the author mentioned about the different UWSN applications like environmental and infrastructure monitoring, border petrol and security monitoring, seismic monitoring etc. The paper summarized the challenges grouped in terms of general and in terms of research like extraction of data, localization, distributed clock synchronization, power conservation topology design etc. while implementing the UWSN devices.

In [12], this paper focuses on gathering recent developments in UWSN applications and deployments. Here, the underwater applications are classified into five main classes namely, monitoring, disaster, military, navigation, and sports. These applications are again divided into relevant subclasses. The paper also shows the challenges and opportunities faced by the deployments of UWSN.

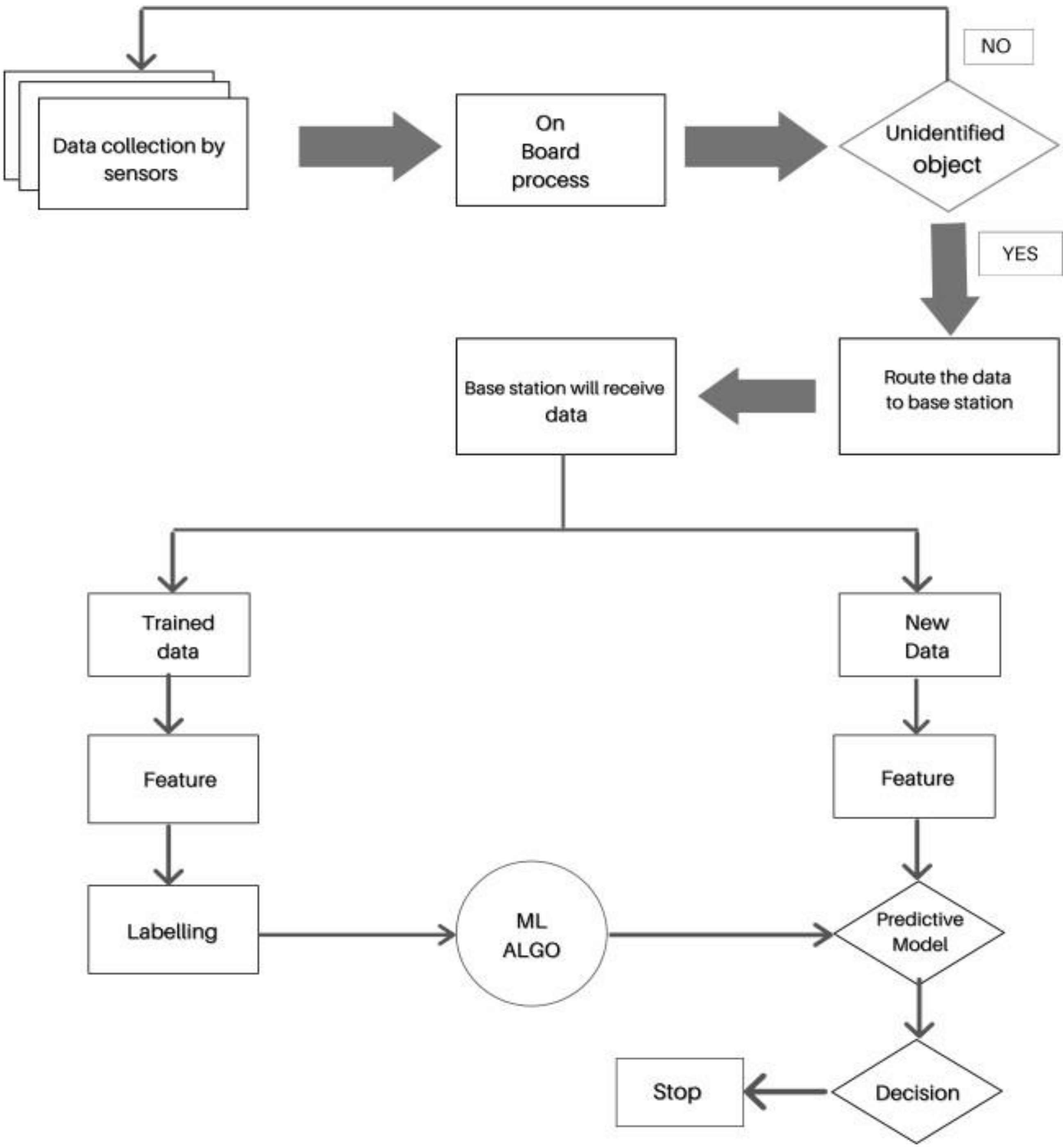
In [13], this paper focuses on gathering recent developments in UWSN applications and deployments. Here, the underwater applications are classified into five main classes namely, monitoring, disaster, military, navigation, and sports. These applications are again divided into relevant subclasses. The paper

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

also shows the challenges and opportunities faced by the deployments of UWN.

3. PROPOESD SYSTEM

The existing system will be studied thoroughly and then identify the pros and cons of the system.
Project implementation scope will be identified.
Then simulation and implementation of the proposed algorithm.
Testing and analysis of the results.
At last, documentation.



The goal of this project is to identify the unidentified objects in underwater with the help of wireless sensor network and artificial intelligence. So, the above block diagram is about how the system is going to process. Firstly, some sensor nodes will be installed inside water and those nodes will collect the data from their surroundings. The nodes will process the data to check the unidentified objects, if yes then the data will be routed to the base station, else again the data will be collected from the sensors. Now, after receiving the data the base station will first compare the data with their trained data along with the features as mentioned in the figure. Once data is matched the machine learning algorithm will help to identify the object, if the received data is not matching with the trained data, the information will be proceed to the predictive model where machine learning algorithm will help to identify the object.

4. IMPLEMENTATION

SIMULATION ENVIRONMENT

Here NS-2 simulator is used to implement the network topology, including nodes (7, 20, 32), links, created some UDP connections, called Agents, and attach them to nodes. The simulation runs for the designated amount of time and generates packets for transmission from one node to another. The nodes are segregated from each other and communication links shows the communication between the nodes while data transferring which is then communicated to the base administrative node. The scripting interface for ns-2 uses the language Tcl.

Simulation Table

PARAMETERS	VALUES
Channel Type	Wireless channel
Radio-propagation model	Two Ray Ground
Network interface type	Wireless Phy
MAC	802.11
Interface queue type	Drop tail/ PriQueue
Link Layer type	LL
Antenna model	Omni Antenna
Max packet in ifq	50
Number of mobile nodes	7
Routing protocol	AODV
X dimension of topography (meters)	500
Y dimension of topography (meters)	500
Simulation time (seconds)	2.0

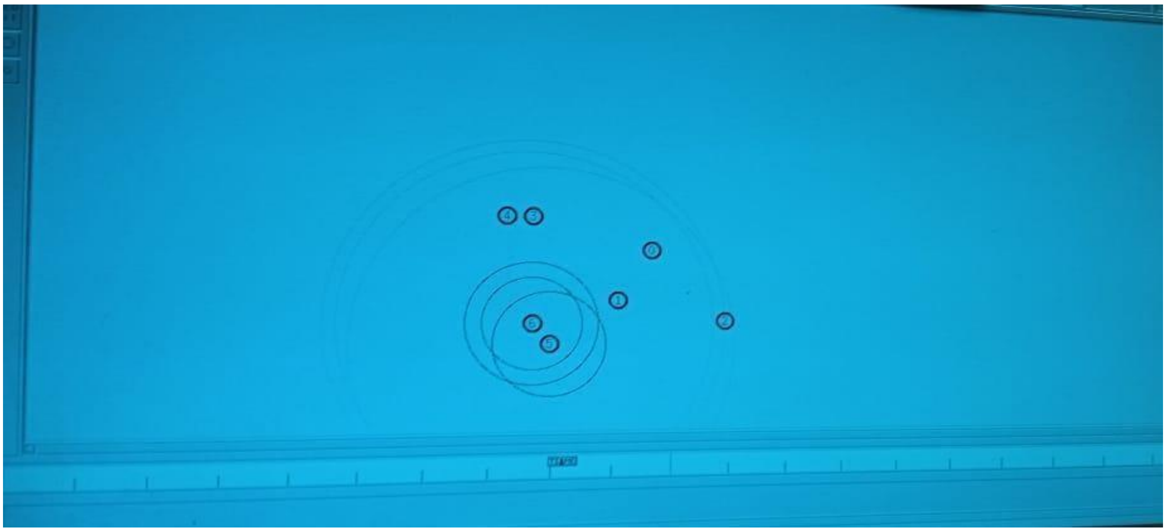


Figure (1):- 7 NODES NETWORK TOPOLOGY

Figure (2):- 7 NODES NETWORK TOPOLOG

The implemented network topology is been modified with more number of nodes, links, created more UDP connections and attached them to nodes. The simulation runs for 2 seconds of time and generates packets for transmission from one node to another. The nodes are segregated from each other and communication links shows the communication between the nodes while data transferring.

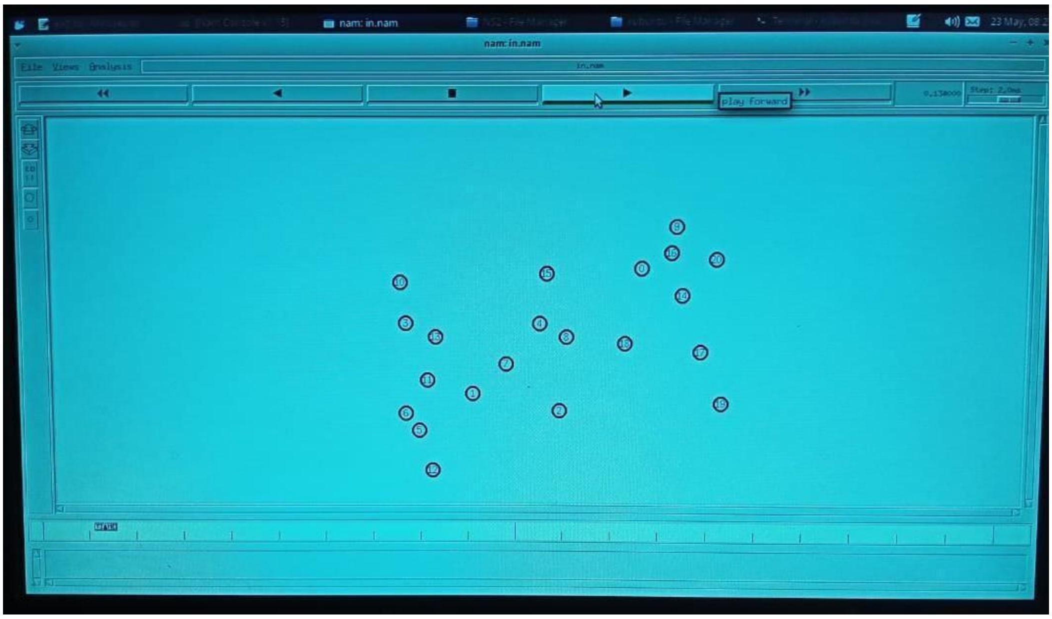


Figure 3:- 20 NODES NETWORK TOPOLOGY

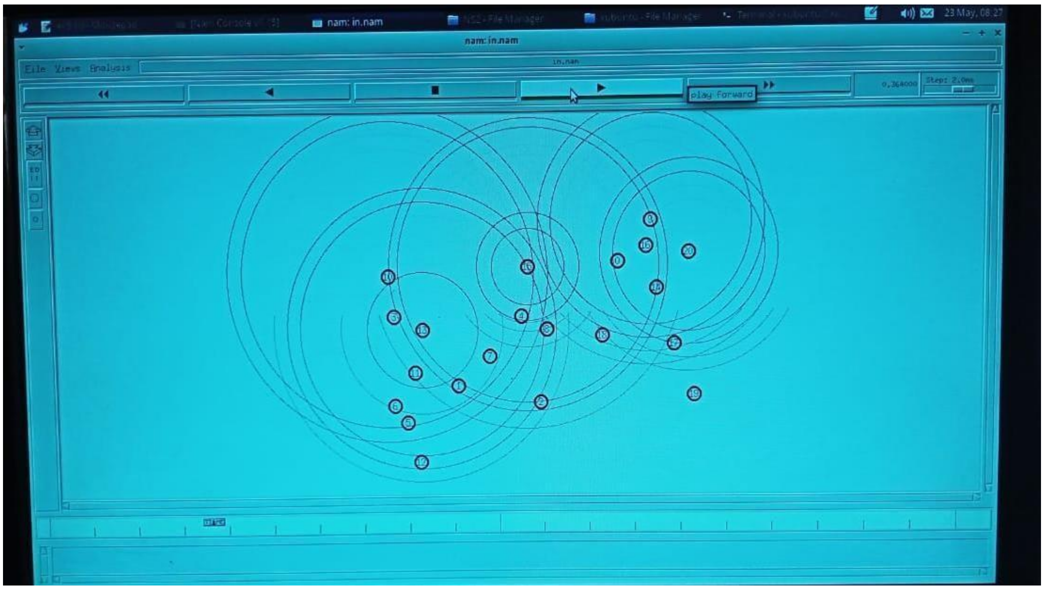


Figure 4:- 20 NODES NETWORK TOPOLOGY

The network topology simulation program generates a NAM trace file that is going to be used as an input toNAM and a trace file for storing whole information in a network called "in.tr" that will be used for our simulation analysis.

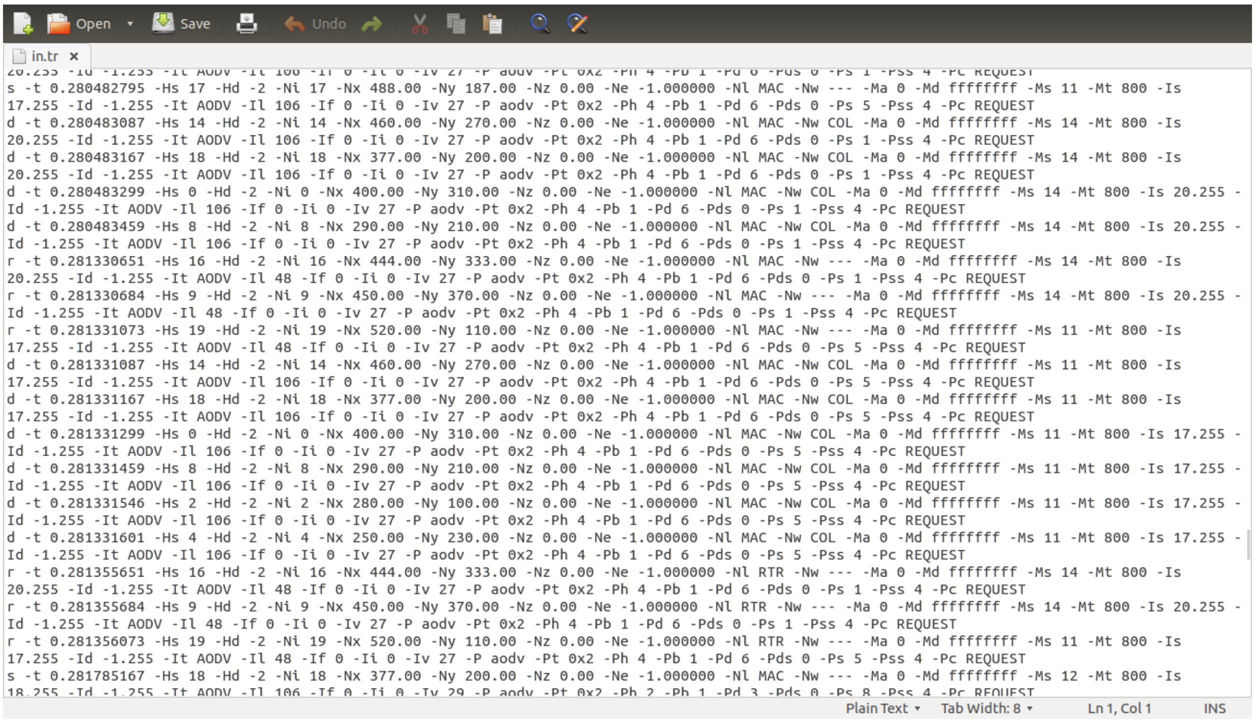


Figure 5: Trace File

<u>Trace file Analysis</u>

To analyze the trace file we will be using AWK scripts. AWK scripts are very good in processing the datafrom the trace files which we get from NS2.

1. We used an AWK script to analyze the trace and calculate throughput for the network topology of source and destination nodes.

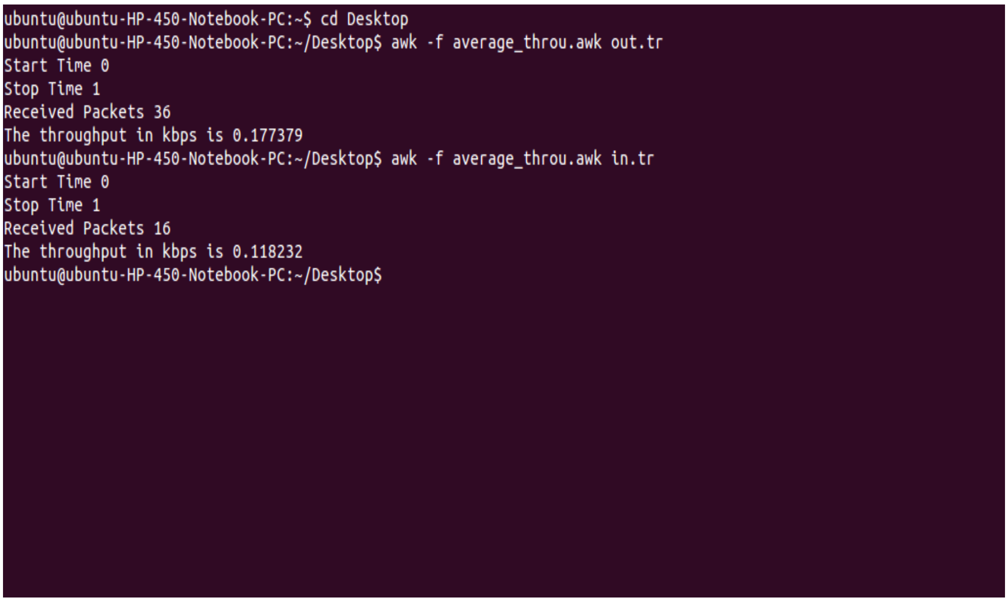


Figure 6: Average throughput

Here, in comparison to network topology of 7 nodes the average throughput is less than the network topology with 20 nodes.

2. We used an AWK script to calculate the packet delivery ratio for the network topology of source and destination nodes.

```
ubuntu@ubuntu-HP-450-Notebook-PC:~/Desktop$ awk -f pdr.awk out.tr
the sent packets are 60
the received packets are 36
the forwarded packets are 0
Packet Delivery Ratio is 0.600000
ubuntu@ubuntu-HP-450-Notebook-PC:~/Desktop$
```

Figure 7: PDR (7 nodes)

PDR: 0.600000

```
ubuntu@ubuntu-HP-450-Notebook-PC:~/Desktop$ awk -f pdr.awk in.tr
the sent packets are 24
the received packets are 16
the forwarded packets are 15
Packet Delivery Ratio is 0.666667
ubuntu@ubuntu-HP-450-Notebook-PC:~/Desktop$
```

Figure 8: PDR (20 nodes)

PDR: 0.666667

Here, packet delivery ratio for network topology with 7 nodes (=0.600000) is less than the PDR of network topology with 20 nodes (= 0.666667)

3. To calculate the energy of the nodes we used an AWK script.

```
ubuntu@ubuntu-HP-450-Notebook-PC:~/Desktop$ awk -f resenergy.awk out.tr
0 0.000000
1 0.000000
2 0.000000
3 0.000000
4 0.000000
5 0.000000
ubuntu@ubuntu-HP-450-Notebook-PC:~/Desktop$ awk -f resenergy.awk in.tr
0 0.000000
1 0.000000
2 0.000000
3 0.000000
4 0.000000
5 0.000000
ubuntu@ubuntu-HP-450-Notebook-PC:~/Desktop$
```

Figure 9: Energy

4. For analyzing the trace for instant throughput of the nodes we used an AWK script.

Figure 10

The instant throughput for the network topology with 7 nodes.

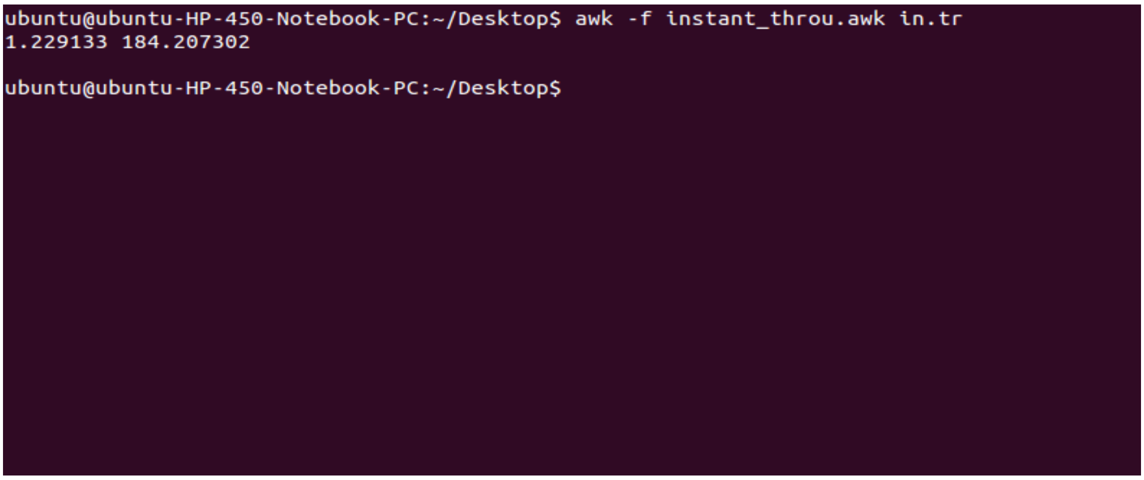


Figure 11

This is the instant throughput for the network topology with 20 nodes.

As we will deploy sensor nodes underwater there will be no external backed power if due to some reason packet loss ratio is increasing the throughput will decrease and the nodes will be dead or will be out of order so, for this reason we have to keep on checking the performance metrics of the network.

(32 NODES NETWORK TOPOLOGY)

We have deployed all the sensor nodes and one mobile node that will depict as unidentified object with one base station node which will receive the data from the sensor nodes and thus implemented the overall network topology.

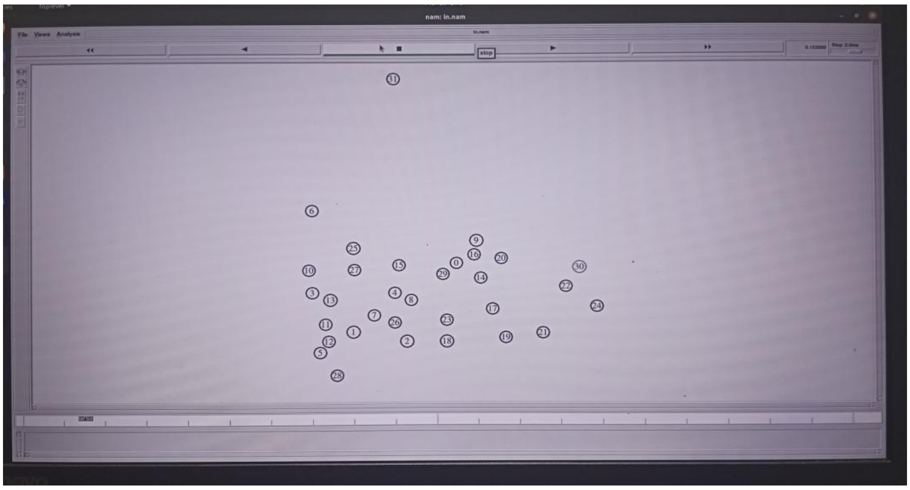


Figure 12:- 32 NODES NETWORK TOPOLOGY

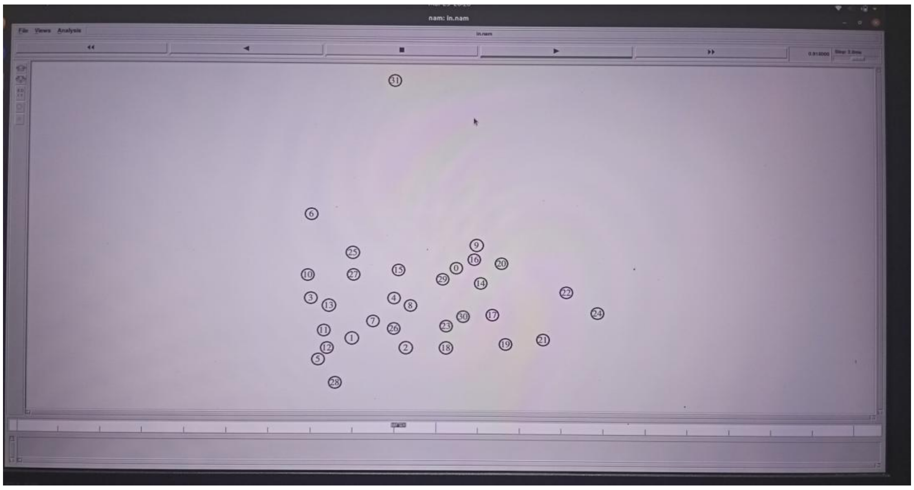


Figure 13:- 32 NODES NETWORK TOPOLOGY

Figure 14:-32 NODES NETWORK TOPOLOGY

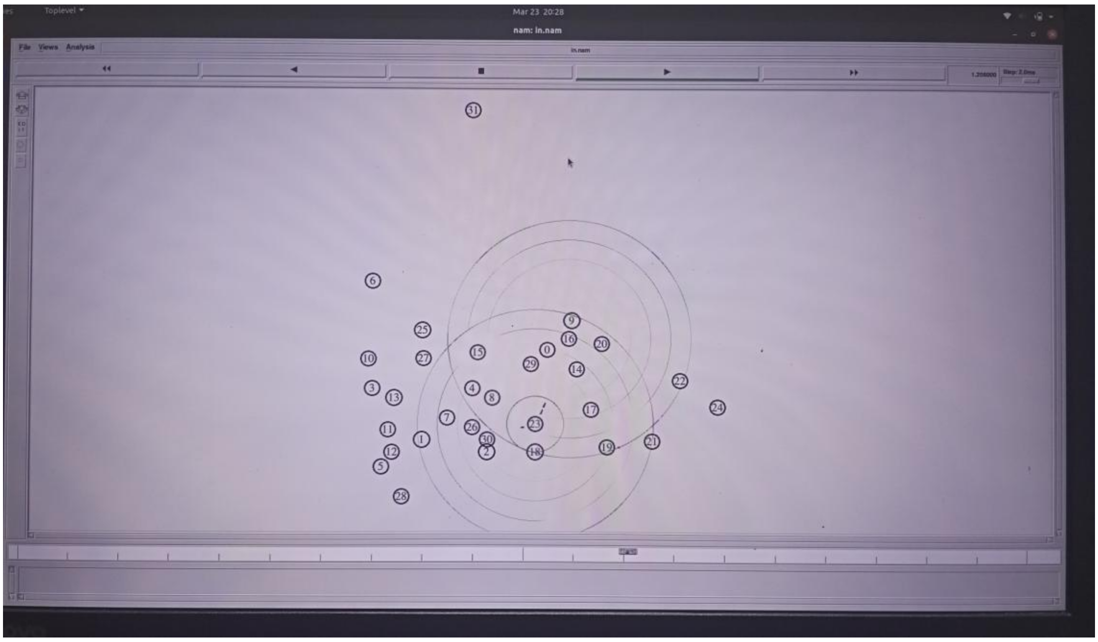


Figure 15:- 32 NODES NETWORK TOPOLOGY

Image Classification

Deep learning is a kind of convolutional neural network (CNN) that has deep layers consisting of convolution layers , polling layers and fully connected layers is used to classify recognize and process unknown inputs. The most commonly used structure of a neural network is formed in three layers called the input layer, hidden layer and output layer.

Dataset and Data Augmentation

We picked four kinds of marine animals namely fish, dolphin, turtle, seal and one fish shape drone for the dataset. There are 20 to 60 images of each kind from Google search engine and various other creative commons sources.

As 200 images are not enough for image classification and because size and the quality were not even we augmented the dataset size to prevent overfitting. We applied some augmented parameters like rotation, zoom, horizontal- flip etc.

Our dataset is divided into train and test. We performed an 80-20 split with the train folder having 113 images and the validation folder has 74 images. This data would be used to train our machine about the different types of images we have. Validation data set would contain 5–10% of the total labeled data. This will test how well our machine performs against known labeled data.

Building Image classification Model

We will be importing the required libraries where will be using the keras library for creating our model and training it. We also use Matplotlib for visualizing our dataset to gain a better understanding of the images we are going to be handling.

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import os
import numpy as np
import pandas as pd

plt.style.use('dark_background')
```

```
import tensorflow as tf
from keras.models import Sequential
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import BatchNormalization
from keras.layers import Dropout
from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten,MaxPool2D
from keras.optimizers import Adam
from keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
```

```
img_height ,img_width =150,150
train_data_dir ='/content/drive/MyDrive/dataset_img'
validation_data_dir ='/content/drive/MyDrive/dataset_img'
batch_size = 10
```

Figure 16

We will be loading our data and define the path to our data then we perform some Data Preprocessing and Data Augmentation before we can proceed with building the model.

```
datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    rotation_range=20,
    validation_split = 0.5

)
train_generator = datagen.flow_from_directory(
    train_data_dir,
    target_size =(img_width, img_height),
    batch_size = batch_size,
    subset ="training",
    class_mode="categorical"
)

val_datagen = ImageDataGenerator(rescale =1./255)

validation_generator = datagen.flow_from_directory(
    validation_data_dir,
    target_size =(img_width, img_height),
    batch_size = batch_size,
    subset ="validation",
    class_mode="categorical"
```

Figure 17

```
activation = 'relu'
model = Sequential()
model.add(Conv2D(16, 3, activation = activation,padding = 'same',  input_shape = (150, 150, 3)))
model.add(MaxPooling2D())
model.add(Dropout(0.3))


model.add(Conv2D(32, 3, activation = activation,padding = 'same',  kernel_initializer = 'he_uniform'))
model.add(MaxPooling2D())

model.add(Conv2D(64, 3, activation = activation, padding = 'same', kernel_initializer = 'he_uniform'))
model.add(MaxPooling2D())
model.add(Dropout(0.3))

model.add(Dense(524, activation = activation))
model.add(Dropout(0.2))

model.add(Flatten())
model.add(Dense(500, activation = activation, kernel_initializer = 'he_uniform'))
model.add(Dropout(0.5))
model.add(Dense(5, activation = 'softmax'))

model.compile(optimizer = 'adam',loss = 'categorical_crossentropy', metrics = ['accuracy'])
print(model.summary())
```

Figure 18

Then we define a simple CNN model with 3 Convolutional layers followed by max-pooling layers. A dropout layer is added after the max pool operation to avoid overfitting. Then we
Compile the model using Adam as our optimizer and Categorical Cross entropy as the loss function. Then we will train our model for 10 epochs.

```
history = model.fit_generator(
        train_generator,
        steps_per_epoch = 9,
        epochs = 10,
        validation_data = validation_generator,
        validation_steps = 2,
        verbose=1
)
```

```
ython3.7/dist-packages/tensorflow/python/keras/engine/training.py:1844: UserWarning: `Model.fit_generator` is deprecated anc
'`Model.fit_generator` is deprecated and '

=================] - 20s 2s/step - loss: 20.9535 - accuracy: 0.2648 - val_loss: 1.5598 - val_accuracy: 0.4000

=================] - 18s 2s/step - loss: 2.3555 - accuracy: 0.3205 - val_loss: 1.5907 - val_accuracy: 0.1000

=================] - 17s 2s/step - loss: 1.6392 - accuracy: 0.2914 - val_loss: 1.6017 - val_accuracy: 0.3500

=================] - 18s 2s/step - loss: 1.4765 - accuracy: 0.3370 - val_loss: 1.6030 - val_accuracy: 0.5000

=================] - 18s 2s/step - loss: 1.3598 - accuracy: 0.4541 - val_loss: 1.5809 - val_accuracy: 0.2000

=================] - 19s 2s/step - loss: 1.4360 - accuracy: 0.4312 - val_loss: 1.5071 - val_accuracy: 0.4500

=================] - 17s 2s/step - loss: 1.3886 - accuracy: 0.3784 - val_loss: 1.4376 - val_accuracy: 0.4500

=================] - 18s 2s/step - loss: 1.2793 - accuracy: 0.4916 - val_loss: 1.5212 - val_accuracy: 0.4000
```

Figure 19

Then plotting our training and validation accuracy along with training and validation loss.

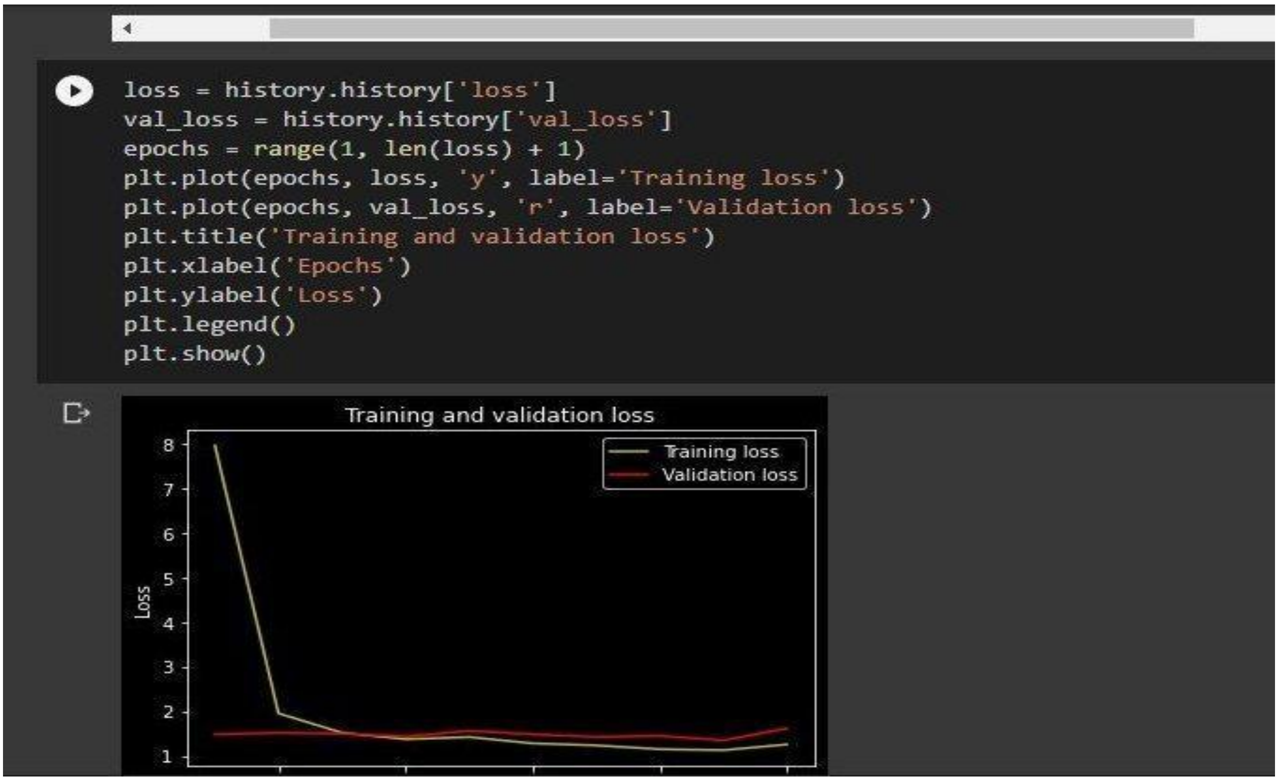


Figure 20

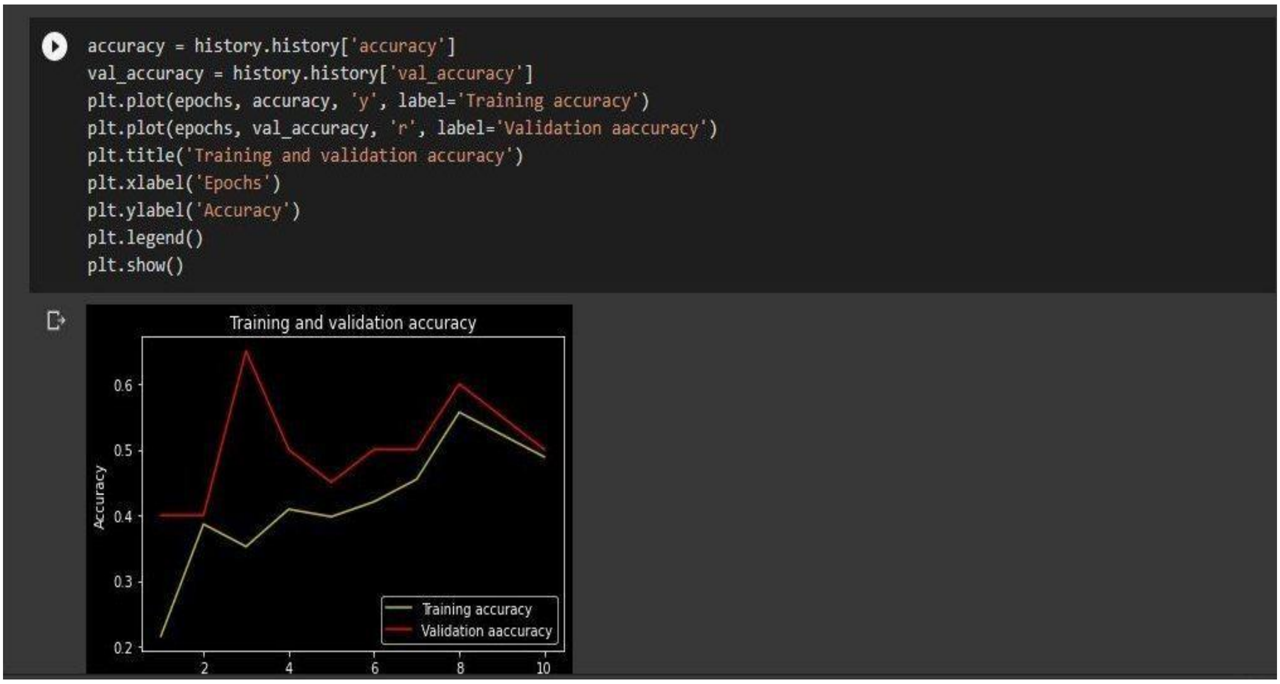


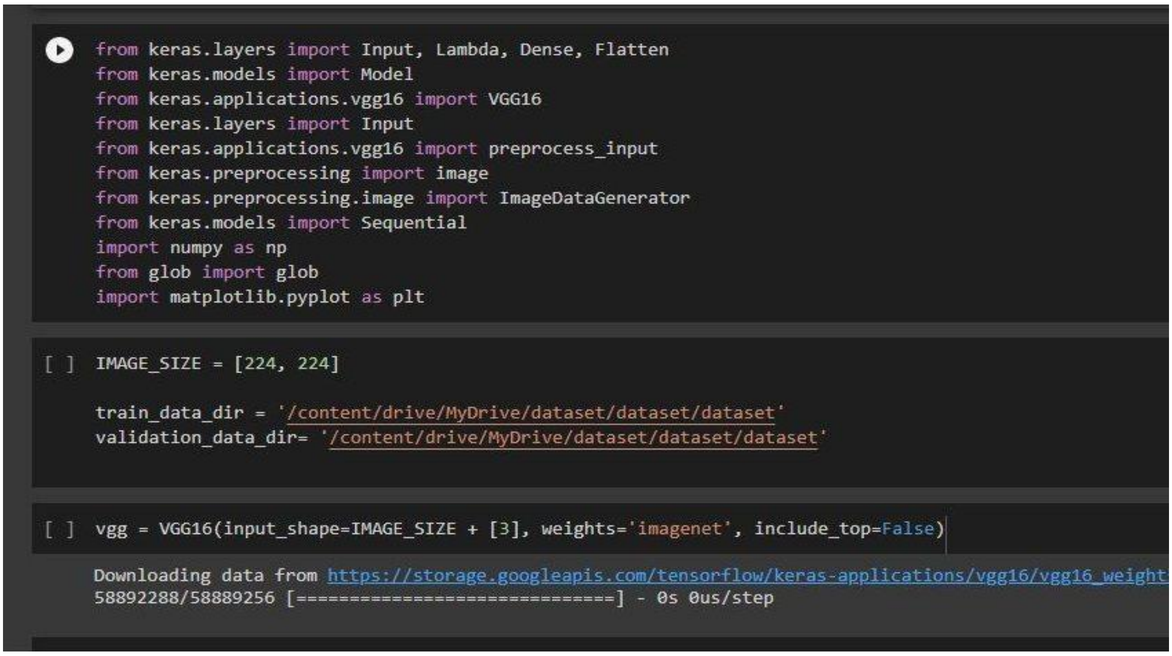
Figure 21

The accuracy is only about 56% with our custom model .So, we use transfer learning technique as our dataset is small and by using a pre-trained model on similar images can achieve high performance.

Our Second model with VGG16

The VGG-16 is one of the most popular pre-trained models for image classification. The model is sequential in nature and uses lots of filters. At each stage, small 3 * 3 filters are used to reduce the number

of parameters all the hidden layers use the ReLU activation function.
We will import all the necessary libraries and then defining dimensions and location images.



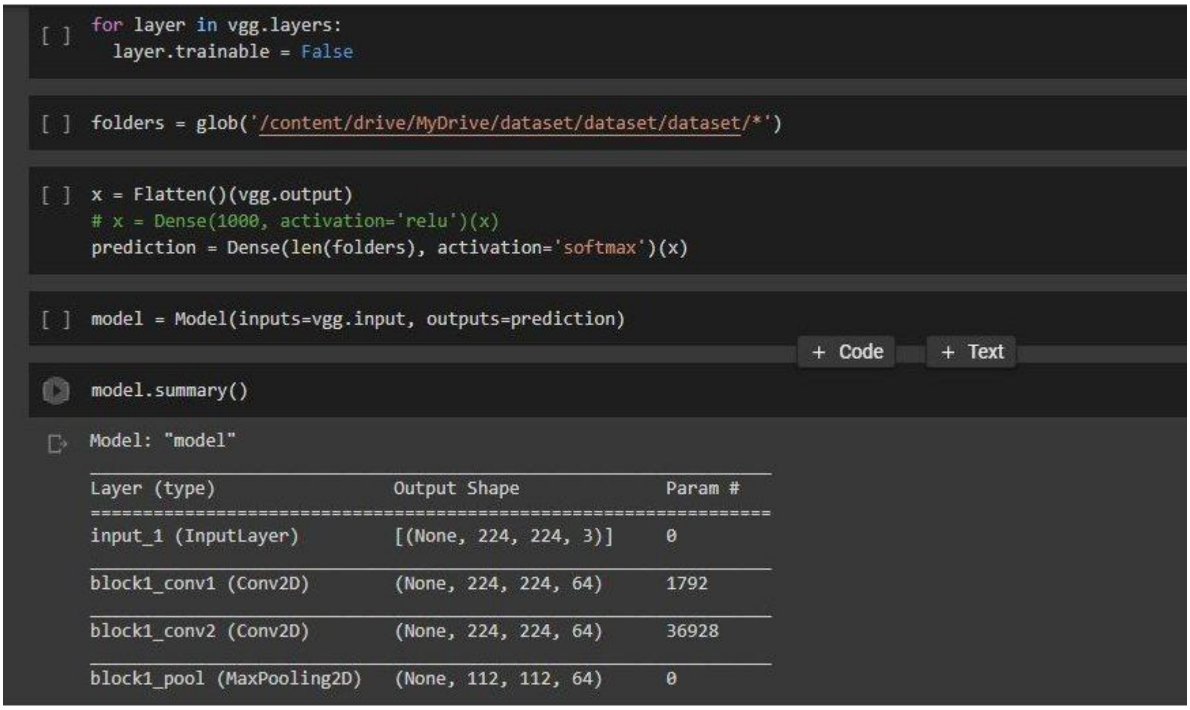
```
from keras.layers import Input, Lambda, Dense, Flatten
from keras.models import Model
from keras.applications.vgg16 import VGG16
from keras.layers import Input
from keras.applications.vgg16 import preprocess_input
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
import numpy as np
from glob import glob
import matplotlib.pyplot as plt
```

```
[ ] IMAGE_SIZE = [224, 224]

    train_data_dir = '/content/drive/MyDrive/dataset/dataset/dataset'
    validation_data_dir= '/content/drive/MyDrive/dataset/dataset/dataset'
```

```
[ ] vgg = VGG16(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)

    Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weight
    58892288/58889256 [==============================] - 0s 0us/step
```

Figure 22

Importing VGG16 it comes with pre-made neural networks and other necessary components that we would otherwise have to create.
Now, we will create our model. After that we add our layer, create a model object and view the structure of the model. Then after we have created and compiled our model, we fit our training and validation data.



```
[ ] for layer in vgg.layers:
      layer.trainable = False

[ ] folders = glob('/content/drive/MyDrive/dataset/dataset/dataset/*')

[ ] x = Flatten()(vgg.output)
    # x = Dense(1000, activation='relu')(x)
    prediction = Dense(len(folders), activation='softmax')(x)

[ ] model = Model(inputs=vgg.input, outputs=prediction)

    model.summary()

    Model: "model"
    _________________________________________________________________
    Layer (type)                Output Shape              Param #
    =================================================================
    input_1 (InputLayer)        [(None, 224, 224, 3)]     0
    _________________________________________________________________
    block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792
    _________________________________________________________________
    block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928
    _________________________________________________________________
    block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0
```


Figure 23

Here we got the accuracy of 84% and loss of 45% .The accuracy is 20% higher than the custom CNN model .Then plotting our training and validation accuracy along with training and validation loss.

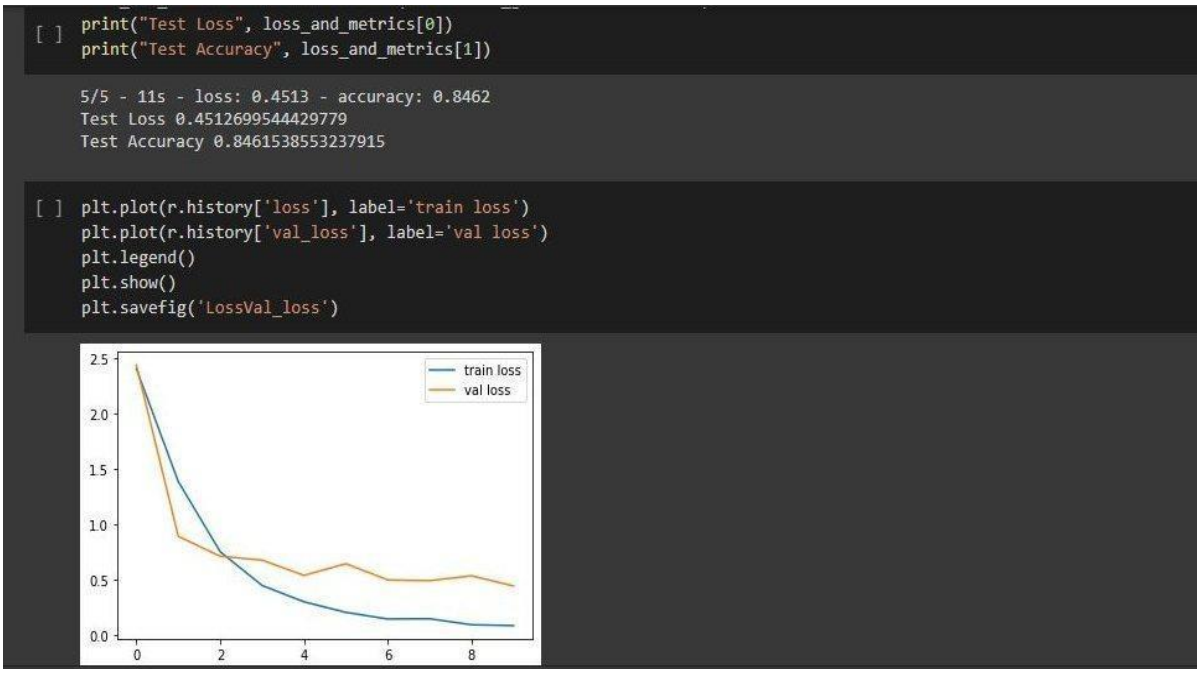


Figure 24

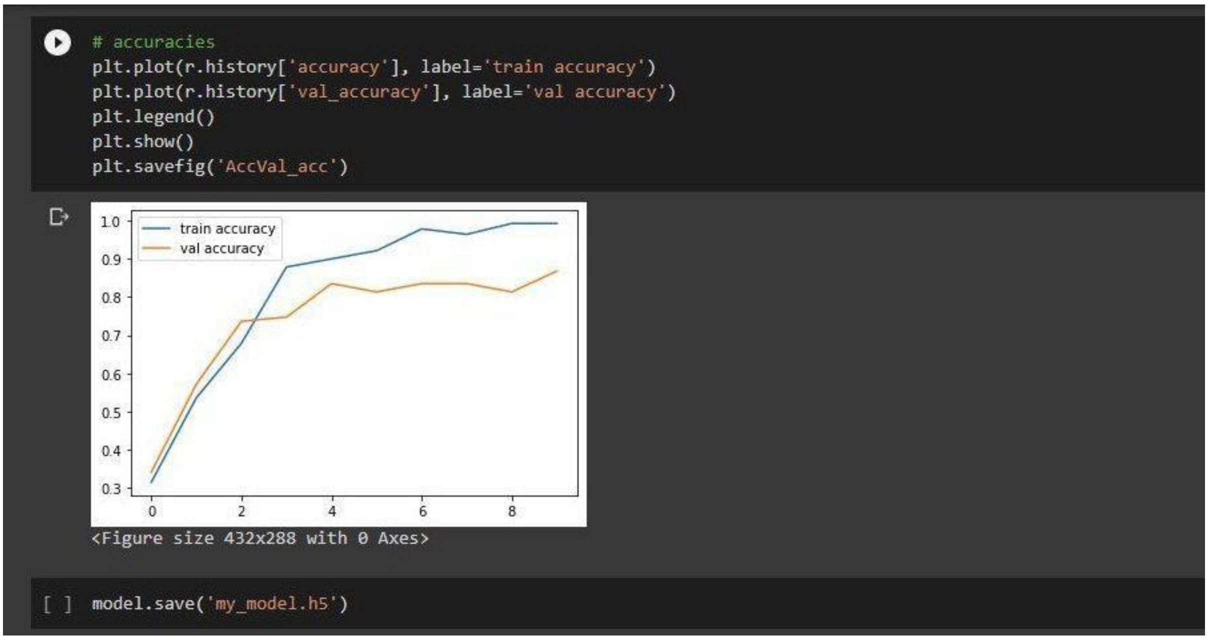


Figure 25

5. REFERENCES

1. A.M.J. Md. ZUBAIR RAHMAN, TAGAJOURNAL, "ENHANCED UNDERWATER WIRELESS SENSOR NETWORKS SECURITY WITH TIC-TAC-TOE AI-MINIMAX ALGORITHM IN GAME THEORY", Al-Ameen
2. Engineering collage, Erode, Tamil Nadu, India.

3. PAL R., GUPTA B., CIANCA E., PATEL A., KALIGOTLA S., GOGAR A., WARDANA S., Lam V.T., GANGULY
4. B. “Playing ‘games’ with human health the role of game theory in optimizing reliability in wireless health networks”; Proceedings of the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL); Rome, Italy. 7–10 November 2010; pp. 1–5.
5. NIYATO D., HOSSAIN E., RASHID M.M., BHARGAVA V.K., “Wireless sensor networks with energy harvesting technologies: A game-theoretic approach to optimal energy management.” IEEE Wirel. Commun. 2007; 14:90–96.
6. CHAOFENG WANG “Intelligent and Secure Underwater Acoustic Communication Networks” Michigan Technological University, Master’s Thesis 2018, cwang8@mtu.edu.
7. ALAMGIR M S M, MST. NAJNIN SULTANA, (Member, and IEEE), AND KYUNGHI CHANG, (Senior
8. Member, and IEEE) Department of Electronic Engineering, “Link Adaptation on an Underwater Communications Network Using Machine Learning Algorithms: Boosted Regression Tree Approach”, Inha University, Incheon 22212, South Korea Corresponding author: Kyunghi Chang (khchang@inha.ac.kr), March 2020.
9. FENGLI HAN, JINGZHENG YAO, HAITAO ZHU, CHUNHUI WANG “Underwater Image Processing and Object Detection Based on Deep CNN Method”, College of Shipbuilding Engineering, Harbin Engineering University, Hindwai, Volume 2020 |Article ID 6707328 | https://doi.org/10.1155/2020/6707328China, May 2020.
10. WALEED RAZA, XUEFEI MA, AMIR ALI, ZUBAIR ALI SHAH, GHAZANFAR MEHDI “An Implementation of
11. Partial Transmit Sequences to Design Energy Efficient Underwater Acoustic OFDM Communication System”,
12. International Journal of Computer Science and Information Security (IJCSIS), Vol. 18, No. 4, April 2020.
13. L.VETRIVENDAN, Dr. R.VISWANATHAN, K. PUNITHARAJA “Security in Underwater Wireless Communication”,
14. International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 4, April 2018 .
15. LIN MENG (IEEE), TAKUMA HIRAYAMA, SHIGERU OYANAGI “Underwater-drone with Panoramic Camera for Automatic Fish Recognition Based on Deep Learning”, Article in IEEE Access March 2018.
16. MAROUANE SALHAOUI, J. CARLOS MOLINA-MOLINA, ANTONIO GUERRERO-GONZALEZ, MOUNIR
17. ARIOUA and FRANCISCO J. ORTIZ, “Autonomous Underwater Monitoring System for Detecting Life on the Seabed by Means of Computer Vision Cloud Services”, Licensee MDPI, Basel, Switzerland, 19 June 2020.
18. SARVESH KUMAR, BERSHA KUMARI, HARSHITA CHAWLA JAYOTI VIDYAPEETH, “Security Challenges
19. and Application for Underwater Wireless Sensor Network”, Kalpa Publications in Engineering Volume 2, 2018, Pages 15– 21 Proceedings on International Conference on Emerging Trends in Expert Applications & Security (2018).

20. EMAD FELEMBAN, FAISAL KARIM SHAIKH, UMAIR MUJTABA QURESHII, "Underwater Sensor Network
21. Applications: A Comprehensive Survey", International Journal of Distributed Sensor Networks, November 1, 2015.
22. GUANG YANG, LIE DAI and ZHIQIANG WEI, "Challenges, Threats, Security Issues and New Trends of Underwater Wireless Sensor Networks", Licensee MDPI, Basel, Switzerland, 13 November 2018.
23. KHALID MAHMOOD AWAN, PEER AZMAT SHAH, KHALID IQBAL, SAIRA GILLANI, WAQAS AHMAD,
24. YUNYOUNG NAM, "underwater wireless sensor network: A review of recent issues and challenges", Volume 2019, Article ID 6470359, 01 January 2019.