MARINE MATE: AQUA META DATA EXPLORATION AND FISH PREDICTION USING DEEP LEARNING

A PROJECT REPORT

Submitted by

NAVEEN R	**310520104077**
NITHISH KUMAR S	**310520104080**
SIVABALA R	**310520104119**
SURYA K S	**310520104130**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

DHANALAKSHMI SRINIVASAN

COLLEGE OF ENGINEERING AND TECHNOLOGY

MAMALLAPURAM, CHENNAI – 603109

ANNA UNIVERSITY :: CHENNAI -600 025

MAY 2024

ANNA UNIVERSITY :: CHENNAI – 600 025

BONAFIDE CERTIFICATE

A Certificate that, this project report **"MARINE MATE: AQUA META DATA EXPLORATION AND FISH PREDICTION USING DEEP LEARNING"** is the bonafide work of "**NAVEEN R (310520104077)**, **NITHISH KUMAR S(310520104080), SIVABALA R(310520104119), SURYA KS(310520104130)"** who carried out the project work under mysupervision.

SIGNATURE

HEAD OF THE DEPARTMENT

Dr. K. JOHN PETER, Ph.D,

Department of Computer Science and Engineering,

Dhanalakshmi Srinivasan College of Engineering & Technology,

Mahabalipuram.

SIGNATURE

SUPERVISOR

Ms. S. ABHINAYA, M.E,

Department of Computer Science and Engineering,

Dhanalakshmi Srinivasan College of Engineering & Technology,

Mahabalipuram.

Submitted for the Practical Examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

For the past decade Artificial Intelligence (AI) is becoming a beneficial part for most. Starting from entertainment to productivity AI can't be missed out. Similarly, India is showcasing its own growth in a whole new genre. Thus, utilizing marine resources wisely will also be a crucial part for highly emerging economy since most of the parts are costal in nature. Despite the fact, analyzing the ocean and its environment still remains unexplored which brings a halfway impossible situation to protect endangered species or to classify such species. Our project is web based and it uses moderate AI interacting tool the website can also detect the species of fish using "ABH fish prediction" and additional information, Meta data and ideology of southern India that initially starts from Chennai. Our project provides data regarding oceanography, seabound geographic, aquaculture, sea surface temperature, PH level and basic web sites that you can visit and get knowledge about fishing zone, minerals, etc. Our progressing project will aim to give a detail description about ocean and its necessary information in a single platform.

Keywords—AI, Oceanography, meta-data, marine data, classification, deep learning.

ACKNOWLEDGEMENT

First of all, we thank, **Our Almighty** for his blessings upon us to strengthen our minds and soul to take up this project. We owe many thanks to many people who helped and supported us in this project.

We thank Our **Chairman, THIRU. A. SRINIVASAN**, who allowed us to do the project.

We would like to thank our **Director, THIRU P. MANI**, who encourage us to do the project.

We would like to thank our **Managing Trustee, Dr. M. KANISHKA REDDY, MBBS**, who encourage us to do the project.

We are also thankful to Our **Principal**, **Dr. T. MANVEL RAJ,Ph.D.,** for his constant support to do the project.

We extremely thank Our **Dean Academic**, **Dr.S. D.GOVARDHAN, Ph. D.,** for his constant support in selecting the project.

We are grateful to Our **Head of the Department**, **Dr. K . JOHN PETER , Ph.D.,** who expressed her interest and guided our work, and supplied us with some useful ideas.

We would like to thank our **Guide Assistant Professor Ms. S. ABHINAYA, M.E** for following our project with interest and for giving me constant support. **She** taught us not only how to do the project, but also how to enjoy the project.

We wish to extend our grateful acknowledgment and sincere thanks to our **project coordinators**, **Dr. K.John peter , Mrs. K.Senbagam** and **Ms. S.Abhinaya** , for their constant support and encouragement and in completing the project.

Furthermore, we would like to thank all our **Teaching Faculty and Non-teaching Faculty** for their timely help in solving any project queries.

Finally, we would like to thank our **Parents** for their blessings, support, and encouragement throughout our life.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	iii
	LIST OF FIGURES	viii
	LIST OF ABBREVATIONS	ix
1	**INTRODUCTION**	
	1.1 MARINE BIODIVERSITY	1
	1.2 META DATA	2
	1.3 WEB TECHNOLOGIES	2
	1.4 CHATBOT	3
	1.5 DEEP LEARNING NEURAL NETWORKS	3
	1.6 CONVOLUTIONAL NEURAL NETWORK	4
	1.7 CNN ARCHITECTURE	5
	1.8 SUMMARY	6
2	**LITERATURE SURVEY**	
	2.1 INTRODUCTION	7
	2.2 LITERATURE SURVEY	7
	2.3 SUMMARY	10
3	**DESIGN CONCEPTS AND METHODOLOGIES**	
	3.1 INTRODUCTION	11
	3.2 EXISTING SYSTEM	11
	3.2.1 LIMITATIONS	12
	3.3 PROPOSED SYSTEM	
	3.3.1 ADVANTAGES	14
	3.4 SYSTEM ARCHITECTURE	14
	3.5 UML DIAGRAM	15
	3.5.1 WORK FLOW DIAGRAM	15
	3.5.2 DATA FLOW DIAGRAM	15

3.6 SUMMARY 16

4 MODULES

4.1 INTRODUCTION 17

4.2 MODULES 17

4.2.1 HOME PAGE 18

4.2.2 CHATBOT – ASSITANT 18

4.2.3 IMPORTANT SPECIES 18

4.2.4 ENDANGERED SPECIES 18

4.2.5 WEATHER 19

4.2.6 MAPS 19

4.2.7 FISH PREDICTION 20

4.2.7.1 FEATURE LEARNING AND CLASSIFICATION 20

4.3 DATA PREPROCESSING 22

4.4 EXPERIMANTAL SETUP FOR CNN 23

4.4.1 DROPOUT 24

4.4.2 METHODALOGY 25

4.5 TRAIN AND VALIDATE THE MODEL 25

4.6 HARDWARE REQUIREMENTS 26

4.7 SOFTWARE REQUIREMENTS 27

4.8 SUMMARY 27

5 RESULT AND DISCUSSION

5.1 INTRODUCTION 28

5.2 PYTHON 28

5.3 TENSORFLOW 31

5.4 KERAS 32

5.5 SCIKIT-LEARN 33

5.6 MATPLOTLIB 39

5.7 PILLOW 41

5.8 PYTHONGUI-TKINTER 43

5.9 SYSTEM STUDY 45
5.9.1 FEASIBILITY STUDY 45
5.9.2 ECONOMICAL FEASIBLITY 45
5.9.3 TECHNICAL FEASIBILITY 45
5.9.4 SOCIAL FEASIBILITY 46
5.10 OUTPUT 47
5.11 SUMMARY 53
6 CONCLUTION AND FUTURE WORK
6.1 CONCLUTION 54
6.2 FUTURE WORK 55
APPENDICES 56
REFERENCES 79

LIST OF FIGURES

FIGURE NO	DESCRPTION	PAGE NO
1.1	CNN ARCHITECTURE	5
3.1	CLASSIFICATION ARCHITECTURE	14
3.2	USECASE DIAGRAM	15
3.3	IMAGE CLASSIFICATION ARCHITECTURE	15
3.4	IMAGE PREDICTION ARCHITECTURE	16
4.1	FEATURE LEARNING AND CLASSIFICATION	20
4.2	MAX POOLING	21
4.3	DATA PREPROCESSING	22
4.4	BEFORE AND AFTER USING DROPOUT	24
5.1	TRAINING OF NEURAL NETWORKS	47
5.2	ACCURACY GRAPH	48
5.3	LOSS GRAPH	48
5.4	INITIAL UI OF THE HOME PAGE & CHATBOT.	49
5.5	INITIAL UI OF THE IMPORTANT SPECIES.	49
5.6	INITIAL UI OF THE ENDANGERED SPECIES	50
5.7	INITIAL UI OF THE FISH INFORMATION.	50
5.8	INITIAL UI OF THE PORT INFORMATION.	51
5.9	INITIAL UI OF THE WEATHER FORECAST.	51
5.10	IMAGE UPLOADED FOR PREDICTION	52
5.11	BLURRED IMAGE PREDICTED AND CLASSIFIED	52

LIST OF ABBREVATIONS

S.NO	ABBREVATION	EXPANSION
1	HTML	HYPER TEXT MARKUP LANGUAGE
2	CSS	CASCADING STYLE SHEET
3	DNNs	DEEP LEARNING NEURAL NETWORKS
4	ANNs	ARTIFICIAL NEURAL NETWORKS
5	CNNs	CONVOLUTIONAL NEURAL NETWORKS

CHAPTER 1

INTRODUCTION

1.1 MARINE BIODIVERSITY

Indian peninsular is surrounded by huge water body like Bay of Bengal, Arabian Sea and Indian Ocean which brings India with vast coast line. It has a sea border of 7516 kms. The Indian Ocean is the third-largest of the world's five oceanic divisions, covering 70,560,000 km^2 (27,240,000 sq mi) or ~20% of the water on Earth's surface India is surrounded by water bodies on three sides and land on the other; therefore, it becomes a Peninsula.

Generally, the world is now experiencing the 4th industrial revolution. Globally technology is improvising every day. The technology is also used to understand the causes of global warming and climate change, which are also closely related to the vast ocean and water bodies. The co- relation of global warming can also bring concepts like sustainable living, zero- waste and protecting endanger species. One of Earth's most important natural resources is the ocean. Marine resources are economically vital and provides a lot of categories like goods, services, jobs opportunities etc. Food, fuel, renewable energy, minerals, sand, gravel and tourism are just a few of the resources that we tend to use and consume to bring living out of it. The blue economy is concerned with the sustainable use of marine resources and the health of marine ecosystems for economic development to improve livelihoods and securing the environment as same. This requires marine researchers to have appropriate training and practical knowledge. Apart from researchers the awareness of a common man will also be a great part of upbring a sustainable environment. Following this trend, these innovative technologies enable marine researchers to use monitoring systems to prioritize biodiversity and to restore the coastal area properties

which soon refers for the future sustainable development projects. They also measure sea surface temperature (SST), salinity, pH, dissolved oxygen (DO), electrical conductivity, total dissolved solids (TDS), and suitability of a site for aquaculture, erosion (cloudiness) monitoring, sea level rise analysis and identification of local native species of aqua culture.

1.2 META DATA

Metadata, structured information describing data characteristics and context, is pivotal in oceanography. It expedites data discovery and access by providing standardized descriptions, particularly valuable in multidisciplinary studies. Moreover, metadata ensures data interoperability, allowing integration across diverse datasets. With details on data quality, such as accuracy and precision, metadata aids in quality assessment and control during analysis. Its transparency fosters reproducibility, vital for scientific integrity. Notably, metadata documentation supports long-term data preservation efforts. Adherence to standards like CF conventions and ISO 19115, alongside domain-specific vocabularies, underscores consistency and interoperability. In essence, metadata serves as a cornerstone, facilitating comprehensive understanding and utilization of oceanographic data for scientific inquiry and environmental stewardship

1.3 WEB TECHNOLOGIES

Integrating web technologies into oceanography offers a powerful framework for effective metadata implementation. Through user-friendly interfaces and standardized templates, researchers can create and annotate metadata seamlessly, ensuring crucial details about datasets are captured. Centralized repositories and catalogs, powered by web platforms, enhance data accessibility by providing standardized metadata storage and search functionalities. Interactive visualization interfaces, enabled by web technologies, dynamically integrate metadata with oceanographic datasets,

facilitating comprehensive data exploration and interpretation. Additionally, web services and APIs promote data interoperability and integration, enabling seamless exchange of information across platforms. Automated quality assurance tools further enhance metadata reliability, ensuring adherence to established standards. In essence, leveraging web technologies in oceanography empowers researchers to maximize the utility and impact of their data, advancing scientific understanding and collaboration within the field.

1.4 CHATBOT

Rule-based chatbots operate on predefined rules, matching user inputs with predetermined patterns to generate responses. Their simplicity and transparency make them easy to develop and understand, with customization options tailored to specific use cases. They offer organizations valuable automation and assistance tools in various applications, such as customer support, information retrieval, and appointment scheduling. Due to their straightforward design, rule-based chatbots can often be deployed with minimal resources and are particularly well-suited for scenarios where quick setup and customization are priorities.

1.5 DEEP LEARNING NEURAL NETWORKS

Deep learning neural networks, or artificial neural networks, attempts to mimic thehuman brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.

Deep learning is a branch of machine learning which is based on artificial neural networks. It is capable of learning complex patterns and relationships within data. In deep learning, we don’t need to explicitly program everything. It has become increasingly popular in recent years due to the

advances in processing power and the availability of large datasets. Because it is based on artificial neural networks (ANNs) also known as deep neural networks (DNNs). These neural networks are inspired by the structure and function of the human brain's biological neurons, andthey are designed to learn from large amounts of data.

Deep Learning is a subfield of Machine Learning that involves the use of neural networks to model and solve complex problems. Neural networks are modeled afterthe structure and function of the human brain and consist of layers of interconnected nodes that process and transform data.The key characteristic of Deep Learning is the use of deep neural networks, which have multiple layers of interconnected nodes. These networks can learn complex representations of data by discovering hierarchical patterns and features in the data.Deep Learning algorithms can automatically learn and improve from data withoutthe need for manual feature engineering.

Deep Learning has achieved significant success in various fields, including image recognition, natural language processing, speech recognition, and recommendationsystems. Some of the popular Deep Learning architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Belief Networks (DBNs).

Training deep neural networks typically requires a large amount of data and computational resources. However, the availability of cloud computing and the development of specialized hardware, such as Graphics Processing Units (GPUs),has made it easier to train deep neural networks.

Neural Networks are used in various datasets like images, audio, and text. Differenttypes of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely, similarly for image classification we use Convolution Neural Networks.

1.6 CONVOLUTIONAL NEURAL NETWORK (CNN)

A Convolutional Neural Network (CNN) is a type of Deep Learning neural networkarchitecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the imageor visual data.

Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominantly used to extract the feature from the grid like matrix dataset.

For example visual datasets like images or videos where datapatterns play an extensive role.

1.7 CNN ARCHITECTURE

Convolutional Neural Network consists of multiple layers like the input layer,Convolutional layer, Pooling layer, and fully connected layers [8].

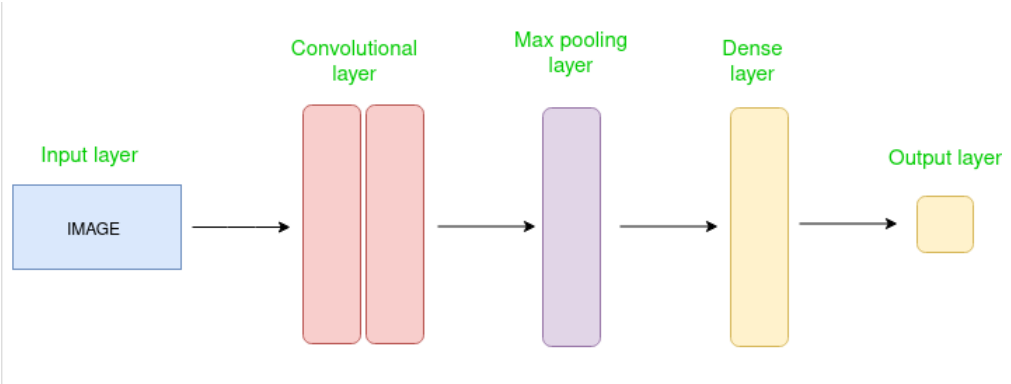


Figure 1.1 CNN ARCHITECTURE

The Convolutional layer applies filters to the input image to extract features, the Pooling layer down samples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

Compared with traditional machine learning methods, adding convolutional

layerscan automatically extract features at different levels in the image, and has translation invariance to the input image. In addition, the convolution kernel in theconvolution layer is parameters of the shared, which greatly reduces the size of theparameters. The pooling layer in the convolution process can reduce the image dimension, retain the ability of key information, and speed up the network training process. Common pool methods include maximum pool, average pool, and randompool. No matter which pooling method is used, its main purpose is to reduce them spatial feature dimension, reduce the system load, and speed up the network training speed. At the end of the neural network, there is usually one, or more fullyconnected layer. Its function is to be extended to a one-dimensional feature map, use the extracted high-dimensional feature information to classify the image, and use the last fully connected layer as output layer, and then network outputs classification result. In addition, the classification activation function can convert the feature information of the image into the (0, 1) interval, which reduces the computer performance consumed during the training process.

1.8 SUMMARY

Brief information about the meta data, web technologies, neural networks and CNN algorithm is explained.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

This chapter provides an extensive review of literature in oceanography, fish prediction, and chatbot implementation, exploring methodologies and advancements in each domain. By synthesizing existing research, this survey lays the groundwork for developing an integrated system for predicting fish behavior, leveraging oceanographic data, and deploying a user-friendly chatbot interface on the web.

2.2 LITERATURE SURVEY

The following are some of the existing works relevant to the collaborative Oceanography.

[1] Singh, A., & Ghosh, S. (2020). Applications of Artificial Intelligence in Marine Science:

A Review. This paper explores the potential applications of artificial intelligence (AI) in marine science, which aligns with our proposed system's integration of AI technologies. It discusses how AI techniques such as machine learning and deep learning can be leveraged to analyze marine ecosystems and biodiversity, providing insights that can inform decision-making processes within our platform.

[2] Sahu, S. K., & Satapathy, S. C. (2019). A Comprehensive Review on Marine Biodiversity Conservation Techniques:

This review examines various techniques for marine biodiversity conservation, which is a key aspect of our proposed system. By understanding different conservation strategies, we can better inform users

about the importance of marine conservation efforts and potentially incorporate features within our platform to promote and support such initiatives.

[3] Das, S., & Das, S. K. (2018). Sustainable Development of Marine Resources:

A Review of Challenges and Opportunities in India. This paper discusses the challenges and opportunities for sustainable development of marine resources, which directly relates to the goals of our proposed system. By addressing these challenges and leveraging opportunities, our platform can contribute to promoting sustainable practices and informed decision-making in marine resource management.

[4] Chakraborty, A., & Banerjee, S. (2019). Integration of AI and GIS for Marine Ecosystem Analysis:

A Review. This review explores the integration of artificial intelligence (AI) and geographic information systems (GIS) for analyzing marine ecosystems, which aligns with the data analysis and visualization aspects of our proposed system. By integrating AI and GIS techniques, we can provide users with valuable insights into marine ecosystems through interactive maps and data-driven analysis tools.

[5] Thakur, N., & Jain, S. K. (2021). Web-based Technologies for Marine Resource Management:

A Review. This paper reviews web-based technologies for marine resource management, which is central to our proposed system. By leveraging web-based platforms, we can provide users with access to valuable information and tools for exploring and understanding marine ecosystems, fostering informed decision-making and stakeholder engagement.

[6] Roy, S., & Das, D. (2020). Role of Web-based Platforms in Marine Conservation:

A Review. This review examines the role of web-based platforms in marine conservation efforts, which aligns with the outreach and community-building aspects of our proposed system. By utilizing web-based platforms, we can raise awareness about marine conservation issues, promote community participation, and facilitate data sharing and collaboration among stakeholders.

[7] Mishra, S., & Mohanty, S. (2018). Role of Artificial Intelligence in Marine Data Analysis:

A Review. This paper explores the role of artificial intelligence (AI) in analyzing marine data, which directly relates to the data analysis capabilities of our proposed system. By leveraging AI techniques such as machine learning and data mining, we can process and analyze vast amounts of marine data to extract valuable insights and inform decision-making processes.

[8] Behera, P., & Mohapatra, S. (2019). Real-time Weather Monitoring Systems for Marine Applications:

A Review. This review examines real-time weather monitoring systems for marine applications, which aligns with the integration of real-time weather information in our proposed system. By incorporating real-time weather data, we can provide users with valuable insights into current marine conditions, enhancing the usability and utility of our platform.

[9] Li, Y., Liu, X., & Zhang, J. (2020). Deep Learning Approaches for Fish Species Classification:

A Review. This paper provides a comprehensive overview of deep learning

approaches for fish species classification, aligning closely with our proposed system's focus on utilizing convolutional neural networks (CNNs) for this task. The review discusses various CNN architectures, such as AlexNet, ResNet, and VGG, and explores their application in accurately identifying different fish species from images. By examining the strengths and limitations of different deep learning techniques, we can effectively design and implement a robust fish species classification model within our system.

[10] Wang, C., Liu, Y., & Zhang, H. (2019). Challenges and Advances in Fish Species Classification Using Deep Learning Techniques:

This paper explores the challenges and recent advances in fish species classification using deep learning techniques, which directly relates to the goals of our proposed system. The review addresses issues such as data scarcity, class imbalance, and model interpretability, while also discussing emerging trends such as transfer learning and data augmentation. By understanding these challenges and advances, we can develop innovative solutions to enhance the accuracy and efficiency of fish species classification within our system.

2.3 SUMMARY

The related works correlated with the current project and the detailed literature review is explained.

CHAPTER 3

DESIGN CONCEPTS AND METHODOLOGIES

3.1 INTRODUCTION

This chapter gives an overall description of existing and proposed concepts and methodologies.

3.2 EXISTING SYSTEM

I. CENTRALIZED PLATFORM

A centralized web platform acts as a single hub for managing diverse online services, resources, and data. It streamlines operations and enhances user experience by providing a unified interface for accessing various functionalities. Through its integrated structure, users can interact seamlessly with different features without switching between multiple platforms. Centralized web platforms often incorporate robust security measures to protect sensitive data and ensure compliance with regulations. Overall, they offer an efficient solution for organizations and users to manage and interact with digital resources online.

II. MANUAL DATA LABELLING

The most common way to label images is manual annotation. This is the process of manually defining labels for an entire image or drawing regions in an image andadding textual descriptions of each region. Image annotation sets a standard, whicha computer vision algorithm tries to learn from.

Manual data labelling involves human annotators identifying objects in images or video frames. These annotators utilize thousands of images (or more) and tag themto gather comprehensive, high-quality data for training AI. Annotators assign labelsbased on the needs of the project.

III. SHAPE DETECTION

Shape detection is an important part of Image Processing referring to modules thatdeal with identifying and detecting shapes of parts of image which differ in brightness, color, or texture.

The borders, the edges of the shapes, are detected by the pattern extractor.

IV. COLOR RECOGNITION

A color detection algorithm identifies pixels in an image that match a specified color or color range. The color of detected pixels can then be changed to distinguishthem from the rest of the image.

3.2.1 LIMITATIONS

- Currently, there is no centralized web platform catering specifically to the field of oceanography in India.
- Manual labeling increases the workload, reduces speed and accuracy rate.
- Artificial feature learning methods are not easy to recognize for images withblurred feature boundaries.
- Most of the existing system works well only with the limited data set. Theyare not capable of handling large datasets.

3.3 PROPOSED SYSTEM

Our project proposes a comprehensive solution to address the critical need for efficient utilization of marine resources by integrating Artificial Intelligence (AI) and web-based technologies. Given India's extensive coastal regions, there is a pressing need to explore and analyze oceanic environments for sustainable development across various sectors, including marine activities. Through the incorporation of moderate AI interacting tools within a web-based platform, our system facilitates the thorough

analysis of marine ecosystems. Central to our approach is the implementation of a rule-based chatbot specifically designed to assist users navigating the website's features and functionalities, providing tailored assistance and information as needed.

The website comprises several meticulously designed sections to cater to diverse user interests. In the "Ecosystem Fishes in Bay of Bengal" section, users gain insights into the fish species found in the Bay of Bengal region, fostering a comprehensive understanding of local marine biodiversity. Additionally, the "Endangered Species List" provides crucial information about endangered marine species, raising awareness and encouraging conservation efforts.

Furthermore, our platform's maps section offers a range of interactive maps seamlessly integrating data such as wind speed sourced from external websites. Users can explore points of interest within sea areas, gaining detailed information through interactive features. Coastal region maps highlighted within the platform serve to emphasize ports, harbors, and India's sea borders, offering users valuable geographical context. Moreover, real-time weather information is seamlessly integrated into the platform using APIs, enriching the user experience by providing updates on marine conditions.

This integrated approach empowers users to comprehensively explore and understand marine ecosystems, thereby promoting sustainable development and conservation efforts. By leveraging AI and web technologies, our platform serves as a valuable tool for stakeholders across various sectors, facilitating informed decision-making and contributing to the preservation of marine biodiversity and ecosystem health.

3.3.1 ADVANTAGES

- The absence of a dedicated centralized web platform for oceanography in India provides a strategic opening for our project to pioneer a tailored solution for marine research and exploration.
- Oceanography websites currently lack chatbot assistance, depriving users of immediate and personalized support during navigation. This absence limits efficient exploration of marine resources and ecosystems.
- The independence from prior knowledge and human effort in feature designis a major advantage in image classification.
- Includes wide variety of datasets and will overcome blurred images.

3.4 SYSTEM ARCHITECTURE OF PREDICTION

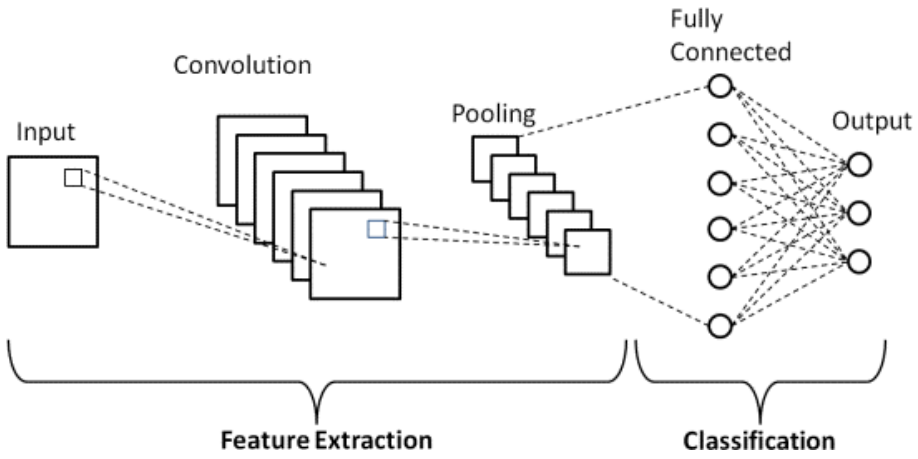


FIGURE 3.1 CLASSIFICATION ARCHITECTURE

The above figures show the classification of the image using Deep Learning Neural Networks. The given input image passes through the series of Convolution and Pooling layer. Convolution layer is used to extract the feature from the input dataset. By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. it will apply an element-wise activation function to the output of the convolution layer. Pooling layer is periodically inserted in the covnets, and

its main function is to reduce the size of volume which makes the computation fast reduces memory and prevents overfitting.

The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layerfor categorization or regression.

After the CNN Layers, the image is passed on to the fully connected neural networkand assigned the classification label as the output

3.5 UML DIAGRAM

3.5.1 WORK FLOW DIAGRAM

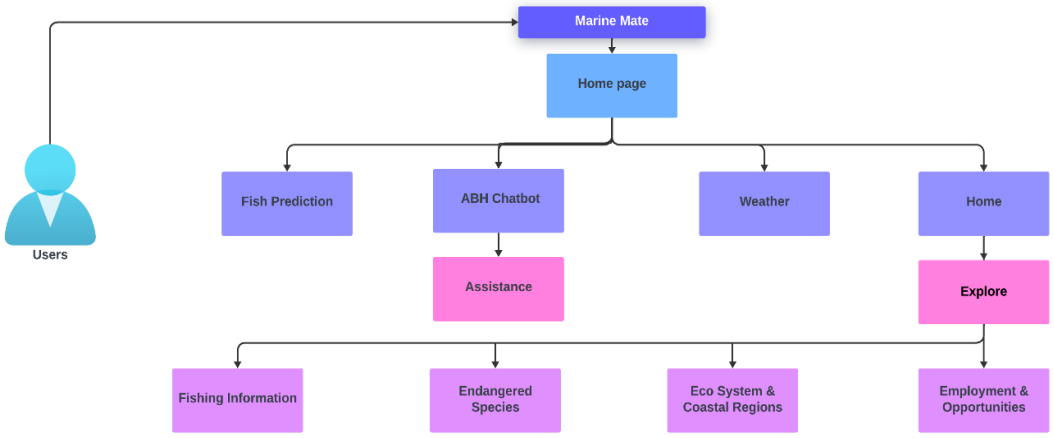


FIGURE 3.2 USECASE DIAGRAM

3.5.2 DATA FLOW DIAGRAM

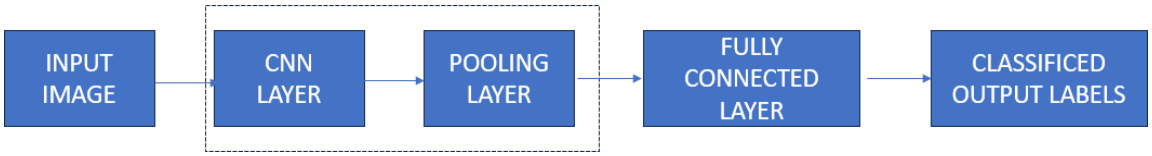


FIGURE 3.3 IMAGE CLASSIFICATION ARCHITECTURE

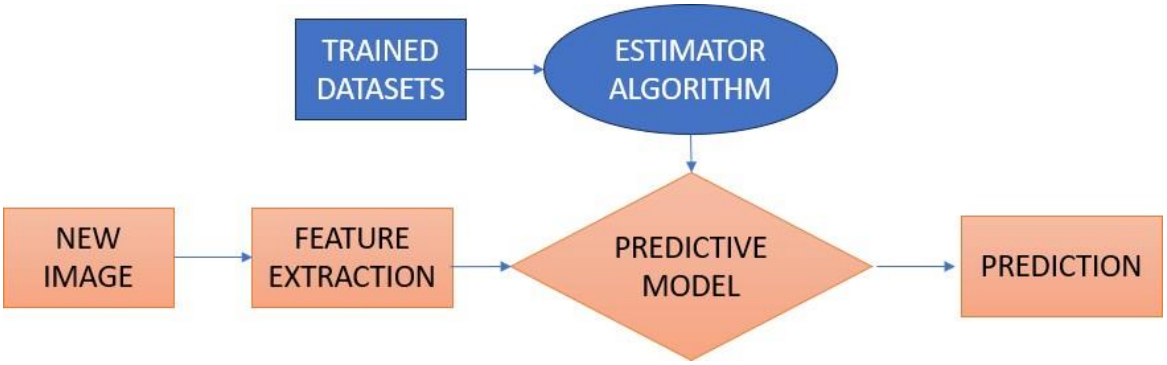


FIGURE 3.4 IMAGE PREDICTION ARCHITECTURE

3.6 SUMMARY

Detailed information about the existing methods and proposed architecture isclearly explained.

CHAPTER 4

MODULES

4.1 INTRODUCTION

This chapter gives an overall description of modules which and all used incomputational view of data formation.

4.2 MODULES

1. Home page
2. Chatbot - Assistant
3. Important Species
4. Endangered Species
5. Weather
6. Maps
7. Fish Prediction
    i. Feature Learning and Classification
        1. Input Layers
        2. Convolutional Layers
        3. Activation Layer
        4. Pooling layer
        5. Flattening
        6. Fully Connected Layers
    ii. Data Preprocessing
    iii. Experimental Setup for CNN
        1. Dropout
        2. Activation Function
    iv. Train and Validate the Model

4.2.1 HOME PAGE

Serving as the main entry point to the website, the Home Page provides users with a centralized hub for accessing key information and navigating different sections. It offers a snapshot of the website's content, highlighting featured topics, news updates, and important announcements. Through intuitive layout and design, the Home Page ensures a user-friendly browsing experience, guiding visitors to explore further and discover relevant resources easily.

4.2.2 CHATBOT – ASSITANT

Module features a rule-based chatbot designed to assist users in navigating the website and accessing relevant information about oceanography. The chatbot engages users in conversation, responding to their queries and providing guidance on various topics such as site navigation, frequently asked questions, and resources available. By offering immediate assistance and personalized responses, the chatbot enhances user experience and facilitates seamless interaction with the website's content and features.

4.2.3 MPORTANT SPECIES

Module focuses on highlighting significant marine species found specifically in the Tamil Nadu region. It provides a curated list of important species native to the area, along with detailed information about their characteristics, habitats, ecological roles, and conservation status. By focusing on the local marine biodiversity, this section aims to raise awareness about the rich and diverse ecosystems present in Tamil Nadu's coastal waters, fostering appreciation and stewardship of the region's marine environment.

4.2.4 ENDANGERED SPECIES

This module is dedicated to raising awareness about endangered marine species that inhabit the waters of Tamil Nadu. It presents a comprehensive

list of species facing threats to their survival in the region, along with detailed information on their population status, habitat degradation, and conservation efforts. By highlighting the plight of these endangered species, the module aims to mobilize support for conservation initiatives and foster a sense of urgency in protecting the unique marine biodiversity of Tamil Nadu's coastal areas.

4.2.5 WEATHER

This module integrates dynamic maps displaying real-time weather data such as temperature, wind speed, and other relevant meteorological information for various regions. Users can interact with these maps to visualize current weather conditions and forecasts. Additionally, the module includes a feature where users can input coordinates to obtain specific weather data for a particular location, providing tailored information on temperature, wind speed, and other relevant weather parameters. This functionality enhances users' ability to plan and make informed decisions related to oceanographic activities and coastal exploration.

4.2.6 MAPS

This module offers a comprehensive visual representation of geographic features and maritime infrastructure, utilizing interactive maps to showcase various elements such as minor ports, major ports, sea borders, and sea routes. Users can explore these maps to gain insights into the layout of coastal regions, shipping lanes, and international boundaries. Additionally, the module includes a scrolling news area adjacent to each map section, providing users with timely updates and relevant information related to maritime activities, port operations, regulatory changes, and other developments impacting oceanography and marine industries.

4.2.7 FISH PREDICTION

4.2.7.1 FEATURE LEARNING AND CLASSIFICATION

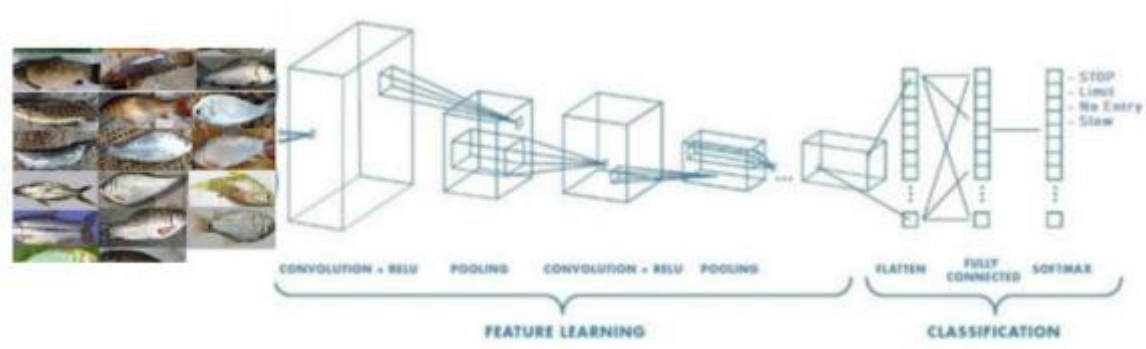


FIGURE 4.1 FEATURE LEARNING AND CLASSIFICATION

1 .INPUT LAYERS

It's the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image with width 32, height 32, and depth 3.

2. CONVOLUTIONAL LAYERS

This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices usually 2×2, 3×3, or 5×5 shape. it slides over the input image data and computes the dot product between kernel weight and thecorresponding input image patch. The output of this layer is referred ad feature maps. Suppose we use a total of 12 filters for this layer we'll get an output volumeof dimension 32 x 32 x 12.

3. ACTIVATION LAYER

By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. it will apply an element-wise activation

function to the output of the convolution layer. Some common activation functions are RELU: max (0, x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimensions 32 x 32 x 12.

4. POOLING LAYER

This layer is periodically inserted in the covnets, and its main function is to reducethe size of volume which makes the computation fast reduces memory and prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.

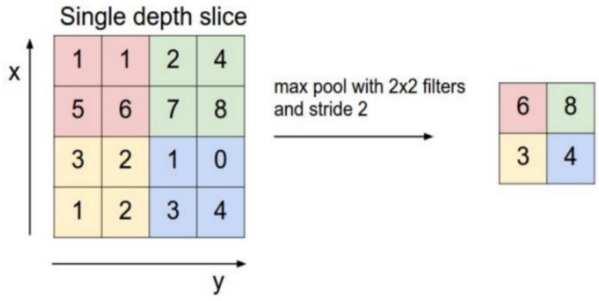


FIGURE 4.2 MAX POOLING

5. FLATTENING

The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layerfor categorization or regression.

6. FULLY CONNECTED LAYERS

It takes the input from the previous layer and computes the final classification or regression task.

4.3 DATA PREPROCESSING

For this project, we have selected the publicly available dataset from

Kaggle. Thedataset contains more than 18,000 images of different fish species. It is further classified into 21 different classes.

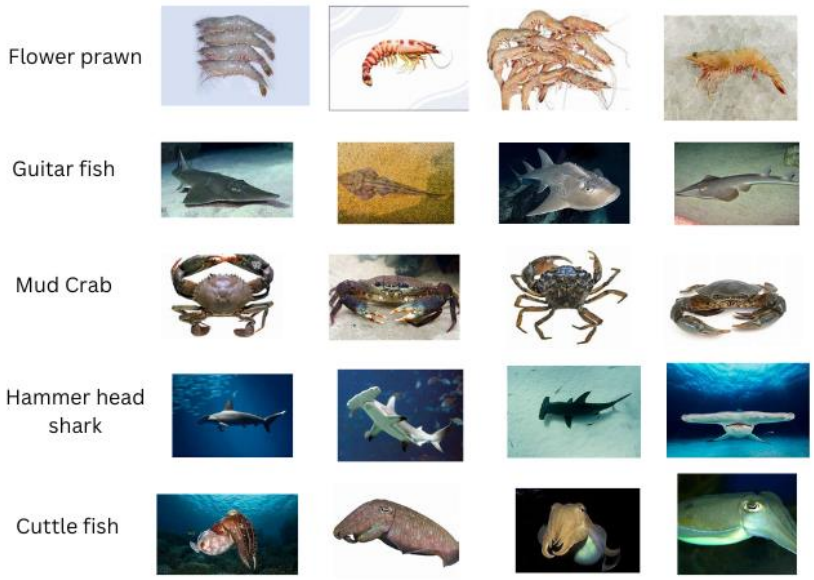


FIGURE 4.3 DATA PREPROCESSING

The ‘train’ folder contains 21 folders each representing a different class. The rangeof the folder is from 0-21. With the help of OS module, we must iterate overall theclasses and append images and their respective labels in the data and label list.

An image is made up of pixels and each pixel has 3 values to specify its color i.e. RGB (Red, Green, Blue). For machines to understand the image, we converted theimage into numbers. For this purpose, we use the PIL (Python Imaging Library) that can perform many image manipulations tasks. The subsequent step was of resizing the images into some uniform criteria. Therefore, we resize all the images

to a fixed size, such as, 30x30.Let’s traverse through all the classes, open the imageusing PIL and resize the image to 30x30 dimensions. Then we will append the dataand label in the X and Y list respectively. After storing all the images and their labels into lists (data and labels), the list was further

transformed into NumPy arrays for feeding the model. Nevertheless, finally the shape of data is (17,854, 30,30, 3) which means that there are 17,854 images of size 30x30 pixels and the last 3 means the data contains coloured images (RGB Value).

4.4 EXPERIMENTAL SETUP FOR CNN

To classify the images into their respective categories, we developed and trained. The CNN have proved the state of the art in image classification tasks, and this is what we will be using for our model. A CNN is made up of convolutional and pooling layers. At each layer, the features from the image are extracted that helps in classifying the image.

The Table shows layers in details of CNN model along with parameters.

Layer Number	Layer Number
L1	Conv2D (32x5x5), ReLU
L2	Conv2D (32x5x5), ReLU
L3	MaxPool2Dlayer (pool_size= (2,2))
L4	Dropout layer (rate=0.25)
L5	Conv2D (64x5x5), ReLU
L6	Conv2D (64x5x5), ReLU
L7	Flatten Layer (1 Dimension)
L8	Dense Fully connected layer (256 nodes, ReLU)
L9	Dropout layer (rate=0.5)
L10	Dense Layer (43 nodes, SoftMax)

TABLE 4.1 EXPERIMENTAL SETPU FOR CNN

Apart from this, a dropout layer is also added, which is used to handle the over fitting of the model. The dropout layer drops some of the neurons while training. Finally, the model is compiled with cross entropy measures because

the dataset hasmulti classes to be classified.

4.4.1 DROPOUT

We apply Dropout technology to the construction of the network [6]. In the processof forward propagation, this method randomly inactivates neurons with a certain probability P to reduce the scale of parameters, improve the generalization ability of the model. Below figure is a before and after diagram.

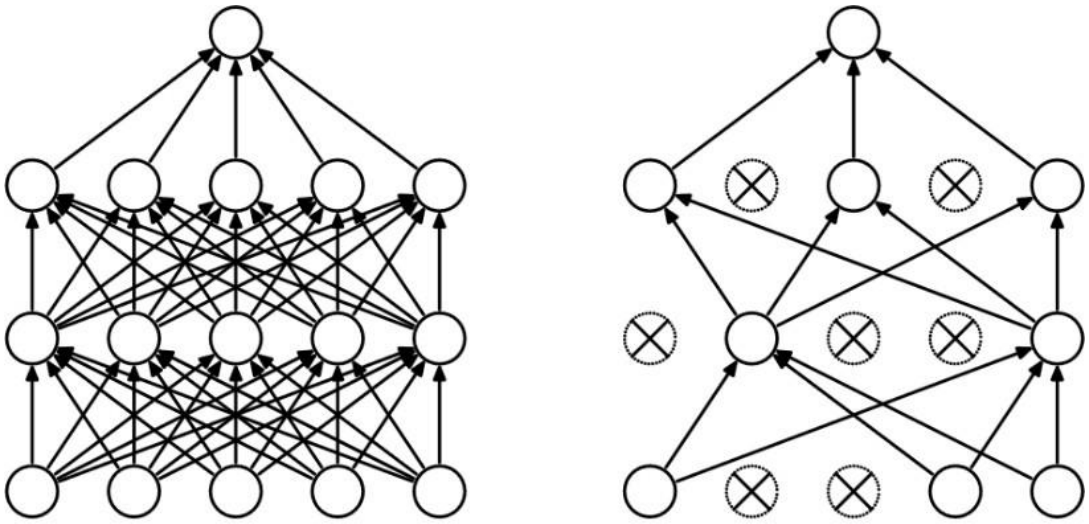


FIGURE 4.4 BEFORE AND AFTER USING DROPOUT

The left picture in the above picture is not using Dropout, the right picture is usingDropout, it can be clearly seen that the complexity of the network structure after using Dropout is reduced, which is helpful to improve the network trainingefficiency and generalization ability.

4.4.2 METHODOLOGY

The target neural network constructed in this paper is trained on the training set toverify the recognition accuracy of the network on the validation set. According to the results on the validation set, the training is continued on the

training set. Finally,the accuracy of the network on the test set is tested.

Technically, deep learning CNN [9], [10], [11], models to train and test, each inputimage will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply SoftMax function to classify an object with probabilistic values between 0 and 1. The calculation of the convolutional layer can be simplified as shown in Equation (1):

$$a^k = f\left(\sum_{i,j=1}^{3} w_{ij}^k \times x_{ij} + b^k\right) \qquad (1)$$

Where, w^k_{ij} is the weight value of the convolution kernel; x_{ij} the input pixel value corresponding to the weight of the convolution kernel; the offset corresponding tothe first convolution kernel; b^k is the bias; f (x) is the activation function. CNN commonly used activation functions as an RLU (Rectified Linear Unit).

4.5 TRAIN AND VALIDATE THE MODEL

After building the architecture of model. The model is defined, and the data is ready. To start the training of our model we initialize the training set, validation set,batch size and no of epochs.

CNN Model No.	Batch Size	Epochs	Test Accuracy (%)
1	32	100	82
2	32	110	91
3	64	100	94
4	64	110	95

TABLE 4.2 CNN TRAINED MODEL SUMMARY

So, for classifier, we opted for multiple architectures of CNN model. We experimented with various number of batch sizes and activation functions. The Table 2 shows the details of all our implemented models.

However, as we find out that CNN4 has outperformed to the rest of other models. Therefore, we used this model to further demonstrations associated with it. Therefore, this model is trained with batch size 64. And after 110 epochs, the accuracy was stable. Our model got 95% accuracy on the training dataset.

4.6 HARDWARE REQUIREMENTS

4.6.1	Processor	:	Intel Core i3 Processor
4.6.2	Speed	:	2.5 GHz
4.6.3	RAM	:	2GB (min)
4.6.4	Hard Disk	:	500MB
4.6.5	Keyboard	:	Standard Windows Keyboard
4.6.6	Mouse	:	Two or Three Button Mouse
4.6.7	Monitor	:	LCD

4.7 SOFTWARE REQUIREMENTS

4.7.1	Operating System	:	Windows 10 or Higher
4.7.2	Language	:	Python
4.7.3	Modules	:	TensorFlow, karas, sklearn, flask.

4.8 SUMMARY

Detailed information about the various modules used in the proposed system is very neatly explained.

CHAPTER 5 RESULTS AND DISCUSSION

5.1 INTRODUCTION

This chapter describes the Language, System and Hardware Specifications of the project to be established.

5.2 HTML5

1.Semantic Elements:

HTML5 introduces semantic elements like `<header>`, `<footer>`, `<nav>`, and `<article>`, which provide more meaningful structure to web documents. Semantic elements describe the purpose of the content they contain, making it easier for search engines to index and understand the information on a web page. For example, the `<header>` element typically contains introductory content or navigation links at the top of a page, while the `<footer>` element typically contains information such as copyright notices or contact details at the bottom of a page. By using semantic elements, developers can create cleaner and more accessible HTML code, improving both usability and search engine optimization (SEO).

2.Multimedia Support:

HTML5 provides native support for multimedia elements such as `<audio>` and `<video>`, allowing developers to embed audio and video content directly into web pages without relying on third-party plugins like Adobe Flash. This native support enhances performance, accessibility, and cross-browser compatibility, as modern web browsers are equipped to handle multimedia content natively. Developers can customize the appearance and behavior of multimedia elements using HTML attributes and JavaScript APIs, enabling a wide range of interactive and engaging multimedia

experiences on the web.

3.Form Enhancements:

HTML5 introduces several enhancements to web forms, including new input types (`<input type="email">`, `<input type="date">`) and attributes (`required`, `placeholder`). These enhancements improve user experience by providing better input validation and feedback, reducing reliance on JavaScript for form validation. For example, the `email` input type ensures that user-entered email addresses are correctly formatted, while the `required` attribute specifies that a form field must be filled out before submission. Additionally, the `placeholder` attribute allows developers to provide hints or examples of expected input directly within form fields, improving usability and accessibility.

4.Responsive Web Design:

HTML5 facilitates responsive web design, allowing websites to adapt dynamically to different screen sizes and devices. Media queries enable developers to apply different stylesheets based on device characteristics such as screen width, height, and orientation, ensuring that content is displayed optimally on smartphones, tablets, and desktops. The `<picture>` element provides a flexible solution for serving responsive images, allowing developers to specify multiple image sources based on device capabilities and screen resolutions. By embracing responsive web design principles, developers can create websites that provide a consistent and user-friendly experience across a variety of devices and screen sizes.

5.Geolocation:

HTML5's Geolocation API enables web applications to access the user's geographical location through the browser. This allows developers to create location-aware services and applications, such as mapping, local

search, and location-based notifications. The Geolocation API provides methods for obtaining the user's latitude and longitude coordinates, as well as their altitude, speed, and heading, if available. With user consent, web applications can access this location information to provide personalized and relevant content based on the user's current location, enhancing the overall user experience.

6.Web Accessibility:

HTML5 includes features designed to improve web accessibility, making it easier for people with disabilities to access and interact with web content. Semantic elements such as `<header>`, `<footer>`, and `<nav>` provide structural landmarks that assistive technologies can use to navigate web pages more effectively. Attributes like `aria-label` and `aria-describedby` allow developers to provide additional context and descriptions for elements, making them more understandable to screen readers and other assistive technologies. By following HTML5 best practices for accessibility, developers can ensure that their websites are inclusive and usable by people of all abilities.

7.Browser Compatibility:

HTML5 enjoys broad support across modern web browsers, including Chrome, Firefox, Safari, Edge, and Opera, as well as mobile browsers on iOS and Android devices. This widespread adoption ensures that HTML5 features and capabilities are consistently available to users across different platforms and devices, reducing compatibility issues for developers. While older versions of Internet Explorer may have limited support for certain HTML5 features, the overall compatibility of HTML5 is continually improving as browsers evolve and standards are implemented more consistently.

8.Future Directions:

HTML5 continues to evolve, with ongoing development and enhancement of standards to address emerging trends and technologies in web development. Future directions for HTML5 may include advancements in areas such as virtual reality (VR), augmented reality (AR), artificial intelligence (AI), and the Internet of Things (IoT), as well as improvements in performance, security, and accessibility. By staying informed about the latest developments in HTML5 and related web technologies, developers can leverage new features and capabilities to create innovative and engaging web experiences for users around the world.

5.3 CSS

1.Selectors and Specificity:

Selectors in CSS are patterns used to target HTML elements for styling. Specificity refers to the hierarchy of selectors and how styles are applied when multiple selectors match the same element. Understanding selectors and specificity is crucial for writing efficient and maintainable CSS code.

2. Layout Techniques:

CSS offers various layout techniques for positioning and arranging elements on a web page. These techniques include float-based layouts, flexbox, and CSS Grid. Each layout method has its advantages and use cases, allowing developers to create responsive and visually appealing designs.

3. Responsive Design and Media Queries:

Responsive design is an approach to web design that ensures web pages adapt gracefully to different screen sizes and devices. Media queries are CSS rules that allow developers to apply different styles based on the characteristics of the device or viewport, enabling responsive layouts and optimized user

experiences across a range of devices.

4. Transitions and Animations:

CSS transitions and animations add interactivity and visual appeal to web pages. Transitions allow smooth changes in property values over a specified duration, while animations provide more complex motion effects. By incorporating transitions and animations, developers can create engaging user experiences and improve usability.

5. Performance Optimization:

Optimizing CSS performance is essential for improving website loading times and user experience. Techniques such as minification, concatenation, and tree shaking reduce the size of CSS files, while browser caching and CDN usage help deliver stylesheets efficiently to users. By optimizing CSS performance, developers can ensure fast and responsive web pages for visitors.

5.4 JAVASCRIPT

1. JavaScript Basics:

JavaScript is a versatile programming language primarily used for adding interactivity and dynamic behavior to web pages. It's an essential part of web development alongside HTML and CSS. JavaScript can manipulate the content and structure of web pages, respond to user actions, and interact with web browsers.

2. Data Types and Variables:

JavaScript supports various data types, including numbers, strings, booleans, arrays, objects, and functions. Variables are used to store and manipulate data in JavaScript. They can be declared using `var`, `let`, or `const` keywords. Understanding data types and variables is fundamental

for writing JavaScript code.

3. Control Flow and Functions:

Control flow structures like `if` statements, `for` loops, and `while` loops allow developers to control the execution flow of JavaScript code based on conditions and iterations. Functions are reusable blocks of code that perform a specific task when called. They help organize code and promote code reusability.

4. DOM Manipulation:

The Document Object Model (DOM) represents the structure of HTML documents as a tree-like hierarchy of nodes. JavaScript can manipulate the DOM dynamically, allowing developers to add, remove, or modify elements and attributes on web pages. DOM manipulation is essential for creating interactive and dynamic user interfaces.

5. Asynchronous JavaScript:

Asynchronous JavaScript allows code to execute non-blocking operations, such as fetching data from servers or handling user input, without freezing the browser's main thread. Techniques like callbacks, promises, and async/await enable asynchronous programming in JavaScript, improving responsiveness and performance in web applications. Understanding asynchronous JavaScript is crucial for building modern web applications with real-time updates and smooth user experiences.

5.5 PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale

projects. Python is dynamically typed, and garbage collected. Python consistently ranks as one of the most popular programming languages.

Syntax and Semantics

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation.

Indentation

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. The recommended indent size is four spaces.

Statements and control flow:

Python's statements include:

- The assignment statement, using a single equals sign =.
- The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
- for statement, which iterates over an iterator object, capturing each elementto a local variable for use by the attached block.
- The while statement, which executes a block of code if its condition is true.
- The Try statement, which allows exceptions raised in its attached code blockto be caught and handled by except clauses; it also ensures that clean-up codein a finally block will always be run regardless of how the block exits.
- The raise statement, used to raise a specified exception or re-raise a caught exception.
- The class statement, which executes a block of code and attaches its

local namespace to a class, for use in object-oriented programming.

- The def statement, which defines a function or method.
- The with statement, which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing resource-acquisition-is- initialization (RAII) - like behaviour and replaces a commontry/finally idiom.
- The break statement exits from a loop.
- The continue statement, skips this iteration, and continues with the next item.
- The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block.
- The del statement removes a variable, which means the reference from the name to the value is deleted and trying to use that variable will cause an error.A deleted variable can be reassigned.
- The assert statement, used during debugging to check for conditions that should apply.
- The yield statement, which returns a value from a generator function and yield is also an operator. This form is used to implement co-routines.
- The return statement, used to return a value from a function.
- The import statement, which is used to import modules whose functions or variables can be used in the current program.

Expressions:

Some Python expressions are like those found in languages such as C and Java,while some are not:

- Addition, subtraction, and multiplication are the same, but the behaviour ofdivision differs. There are two types of divisions in

Python. They are floor division (or integer division) // and floating-point / division. Python also usesthe ** operator for exponentiation.

- From Python 3.5, the new @ infix operator was introduced. It is intended to be used by libraries such as NumPy for matrix multiplication.
- From Python 3.8, the syntax: =, called the 'walrus operator' was introduced.It assigns values to variables as part of a larger expression.
- In Python, == compares by value, versus Java, which compares numeric byvalue and objects by reference. (Value comparisons in Java on objects can be performed with the equals () method.) Python's is operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example A<=B<=C.
- python uses the words and, or, not for or its Boolean operators rather than the symbolic &&, ||, ! used in Java and C.
- Python has a type of expression termed a list comprehension as well as a more general expression termed a generator expression.
- Anonymous functions are implemented using lambda expressions; however,these are limited in that the body can only be one expression.
- Conditional expressions in Python are written as x if c else y (different in order of operands from the c? x: y operator common to many other languages).
- Python makes a distinction between lists and tuples. Lists are written as [1, 2, 3], are mutable, and cannot be used as the keys of dictionaries (dictionarykeys must be immutable in Python). Tuples are written as (1, 2, 3), are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable. The + operator can be used to concatenate two tuples. Parentheses are optional for tuples in unambiguous contexts.

- Python features sequence unpacking wherein multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property,etc.), are associated in an identical manner to that forming tuple literals andare put on the left-hand side of the equal sign in an assignment statement.
- Python has a "string format" operator %. This functions analogously ton printf format strings in C, e.g. “spam=%s eggs=%d” % (“blah”,2) evaluates to “spam=blah eggs=2”. In Python 3 and 2.6+, this was supplemented by the format () method of the str class, e.g. “spam={0}eggs={1}”.format(“blah”,2). Python 3.6 added "f-strings": blah = “blah”; eggs = 2; f‘spam={blah} eggs={eggs}’
- Strings in Python can be concatenated, by "adding" them (same operator as for adding integers and floats). E.g. “spam” + “eggs”returns “spameggs”. Even if your strings contain numbers, they are still added as strings rather than integers. E.g. “2” + “2” returns “2”.Python has various kinds of string literals:
- Strings delimited by single or double quote marks. Unlike in Unix shells, Perl and Perl-influenced languages, single quote marks and double quote marks function identically. Both kinds of string use the backslash (\) as an escape character. String interpolation became available in Python 3.6 as "formatted string literals".
- Triple-quoted strings, which begin and end with a series of three single or double quote marks. They may span multiple lines and function like here documents in shells, Perl and Ruby.
- Raw string varieties, denoted by prefixing the string literal with an r. Escape sequence are not interpreted; hence raw strings are useful where literal backslashes are common, such as regular expressions and Windows-style paths. Compare "@- quoting" in C#.
- Python has array index and array slicing expressions on lists, denoted as a[Key], a[start:stop] or a[start:stop:step]. Indexes are zero-based,

and negative indexes are relative to the end. Slices take elements from the start index up to, but not including the stop index. The third slice parameter, calledstep or stride, allows elements to be skipped and reversed. Slice indexes maybe omitted, for example a [:] returns a copy of the entire list. Each element of a slice is a shallow copy.

Method

Methods on objects are functions attached to the object's class, the syntax instance.Method(argument) is, for normal methods and functions, syntactic sugar for Class. Method (instance, argument). Python methods have an explicit self-parameter access instance data, in contrast to the implicit self (or this) in some other object- oriented programming languages (e.g., C++, Java, Objective-C, or Ruby). Apart from this Python also provides methods, sometimes called d-under methods due totheir names beginning and ending with double-underscores, to extend the functionality of custom class to support native functions such as print, length, comparison, support for arithmetic operations, type conversion, and many more.

Typing

Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being dynamically typed, Python is strongly typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silentlyattempting to make sense of them. Python allows programmers to define their owntypes using classes, which are most often used for object-oriented programming.

5.6 TENSORFLOW

TensorFlow is an open-source machine learning library developed by Google. TensorFlow is used to build and train deep learning models as it facilitates the creation of computational graphs and efficient execution on various hardware platforms.

TensorFlow is basically a software library for numerical computation using data flow graphs where:

- nodes in the graph represent mathematical operations.
- edges in the graph represent the multidimensional data arrays (called tensors)communicated between them. (Please note that tensor is the central unit of data in TensorFlow).

Computational Graph

Any TensorFlow Core program can be divided into two discrete sections:

- Building the computational graph. A computational graph is nothing but a series of TensorFlow operations arranged into a graph of nodes.
- Running the computational graph. To evaluate the nodes, we must run the computational graph within a session. A session encapsulates the control andstate of the TensorFlow runtime.

Variables

TensorFlow has Variable nodes too which can hold variable data. They are mainlyused to hold and update parameters of a training model. Variables are in-memory buffers containing tensors. They must be explicitly initialized and can be saved todisk during and after training. You can later restore saved values to exercise or analyse the model.

Placeholders

A graph can be parameterized to accept external inputs, known as placeholders. Aplaceholder is a promise to provide a value later. While evaluating the graph involving placeholder nodes, a feed_dict parameter is passed to the session's run method to specify Tensors that provide concrete values to these placeholders.

Prepare and load data for successful ML outcomes.

Data can be the most important factor in the success of your ML endeavours. TensorFlow offers multiple data tools to help you consolidate, clean and preprocessdata at scale:

- Standard datasets for initial training and validation.
- Highly scalable data pipelines for loading data.
- Preprocessing layers for common input transformations.
- Tools to validate and transform large datasets.

Additionally, responsible AI tools help you uncover and eliminate bias in your datato produce fair, ethical outcomes from your models.

Build and fine-tune models with the TensorFlow ecosystem.

Explore an entire ecosystem built on the Core framework that streamlines model construction, training, and export. TensorFlow supports distributed training, immediate model iteration and easy debugging with Keras, and much more. Toolslike Model Analysis and Tensor Board help you track development and improvement through your model's lifecycle.

5.7 KERAS

Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers, Francois Chollet. It is made user- friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

Keras is the high-level API of the TensorFlow platform. It provides an approachable, highly productive interface for solving machine learning (ML)problems, with a focus on modern deep learning. Keras covers every step of the machine learning workflow, from data processing to hyperparameter tuning to deployment. It was developed with a focus on enabling fast experimentation.

With Keras, you have full access to the scalability and cross-platform capabilities of TensorFlow. You can run Keras on a TPU Pod or large clusters of GPUs, and you can export Keras models to run in the browser or on mobile devices. You canalso serve Keras models via a web API.

Keras is designed to reduce cognitive load by achieving the following goals:

- Offer simple, consistent interfaces.
- Minimize the number of actions required for common use cases.
- Provide clear, actionable error messages.
- Follow the principle of progressive disclosure of complexity: It's easy to getstarted, and you can complete advanced workflows by learning as you go.
- Help you write concise, readable code.

Who should use Keras?

The short answer is that every TensorFlow user should use the Keras APIs by default. Whether you're an engineer, a researcher, or an ML practitioner, you shouldstart with Keras.

There are a few use cases (for example, building tools on top of TensorFlow or developing your own high-performance platform) that require the low-level TensorFlow Core APIs. But if your use case doesn't fall into one of the Core API applications, you should prefer Keras.

Keras API components

The core data structures of Keras are layers and models. A layer is a simple input/output transformation, and a model is a directed acyclic graph (DAG) of layers.

Layers

The tf.keras.layers.Layer class is the fundamental abstraction in Keras. A Layer encapsulates a state (weights) and some computation (defined in the tf.keras.layers.Layer.call method).

Weights created by layers can be trainable or non-trainable. Layers are recursively composable: If you assign a layer instance as an attribute of another layer, the outerlayer will start tracking the weights created by the inner layer.

You can also use layers to handle data preprocessing tasks like normalization and text vectorization. Preprocessing layers can be included directly into a model, eitherduring or after training, which makes the model portable.

Models

A model is an object that groups layers together and that can be trained on data.

The simplest type of model is the Sequential model, which is a linear stack of layers. For more complex architectures, you can either use the Keras functional API, which lets you build arbitrary graphs of layers, or use subclassing to write models from scratch.

The tf.keras.Model class features built-in training and evaluation methods:

- tf.keras.Model.fit: Trains the model for a fixed number of epochs.
- tf.keras.Model.predict: Generates output predictions for the input samples.
- tf.keras.Model.evaluate: Returns the loss and metrics values for the model;configured via the tf.keras.Model.compile method.

These methods give you access to the following built-in training features:

Callbacks: You can leverage built-in callbacks for early stopping, model checkpointing, and TensorBoard monitoring. You can also implement custom callbacks.

Distributed training: You can easily scale up the training to multiple GPUs, TPUs,or devices.

Step fusing. With the steps_per_execution argument in tf.keras.Model.compile, you can process multiple batches in a single tf.function call, which greatly improves device utilization on TPUs.

5.8 MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on

NumPy arrays anddesigned to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

Types of Matplotlib

Matplotlib comes with a wide variety of plots. Plots help to understand trends, andpatterns, and to make correlations. They're typically instruments for reasoning about quantitative information. Some of the sample plots are listed here.

- Matplotlib Line Plot
- Matplotlib Bar Plot
- Matplotlib Histograms Plot
- Matplotlib Scatter Plot
- Matplotlib Pie Charts
- Matplotlib Area Plot

5.9 SYSTEM STUDY

5.9.1 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is putforth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are.

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

5.9.2 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have onthe organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified.

Thus, the developed system as well within the budget and this was achieved because most of the technologies used are open source and are freely available.

5.9.3 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.

This will lead to high demands on the available technical resources. This

will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementingthis system.

5.9.4 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The usermust not feel threatened by the system, instead must accept it as a necessity.

The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5.10 OUTPUT

1.The picture shows the training of the Neural Network.

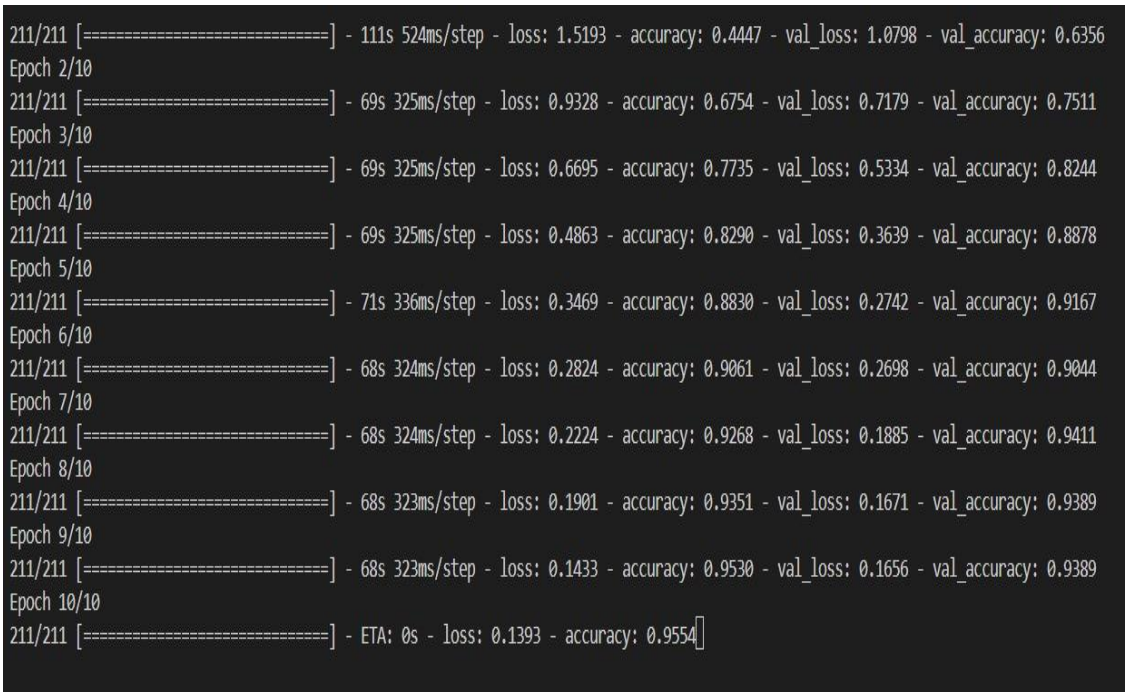
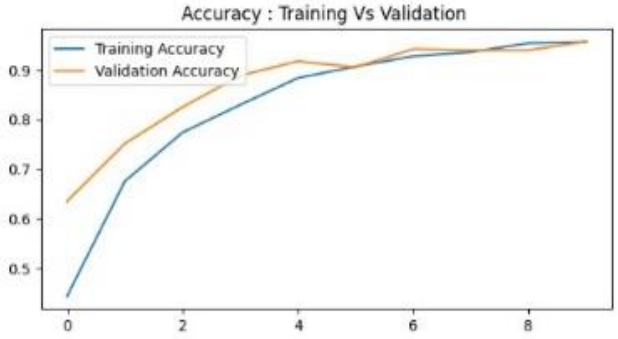


FIGURE 5.1 TRAINING OF NEURAL NETWORKS

2.The picture of the accuracy graph of the Classifier.

FIGURE 5.2 ACCURACY GRAPH



3.The picture of the loss graph of the Classifier.

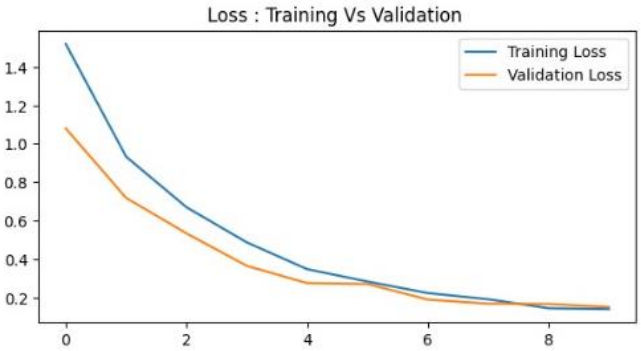


FIGURE 5.3 LOSS GRAPH

4.The screenshot of the initial user interface of home page.

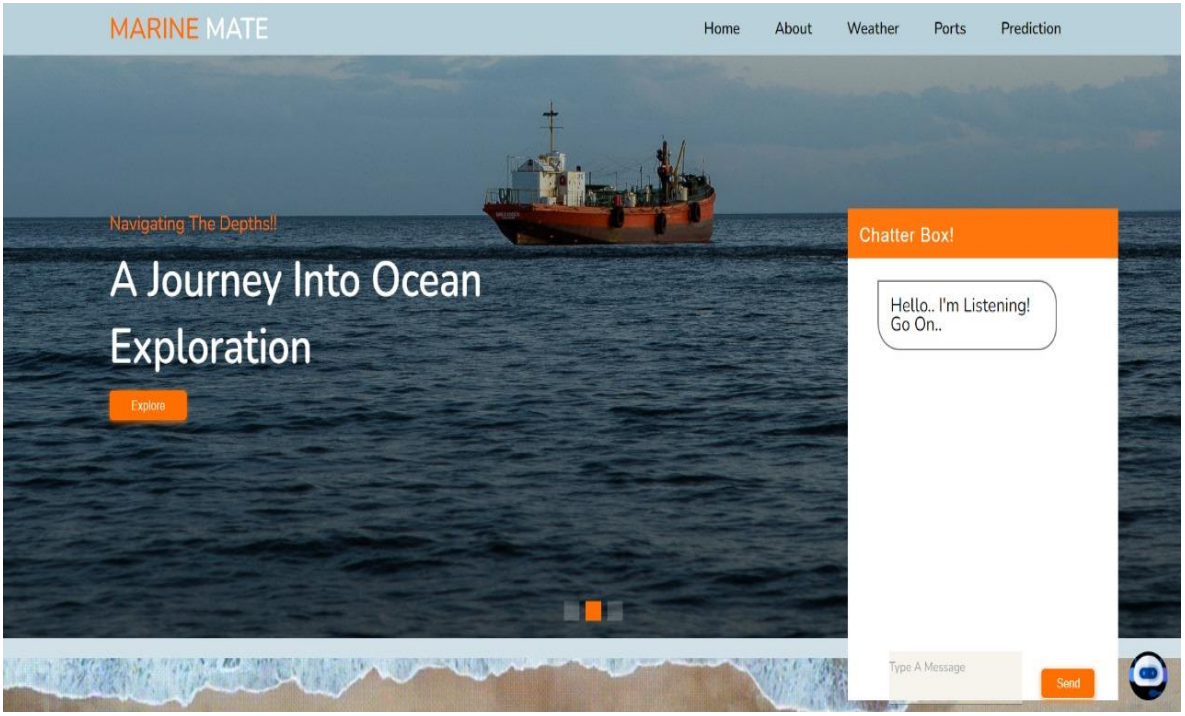


FIGURE 5.4 INITIAL UI OF THE HOME PAGE & CHATBOT.

5.The screenshot of the initial user interface of Important species.

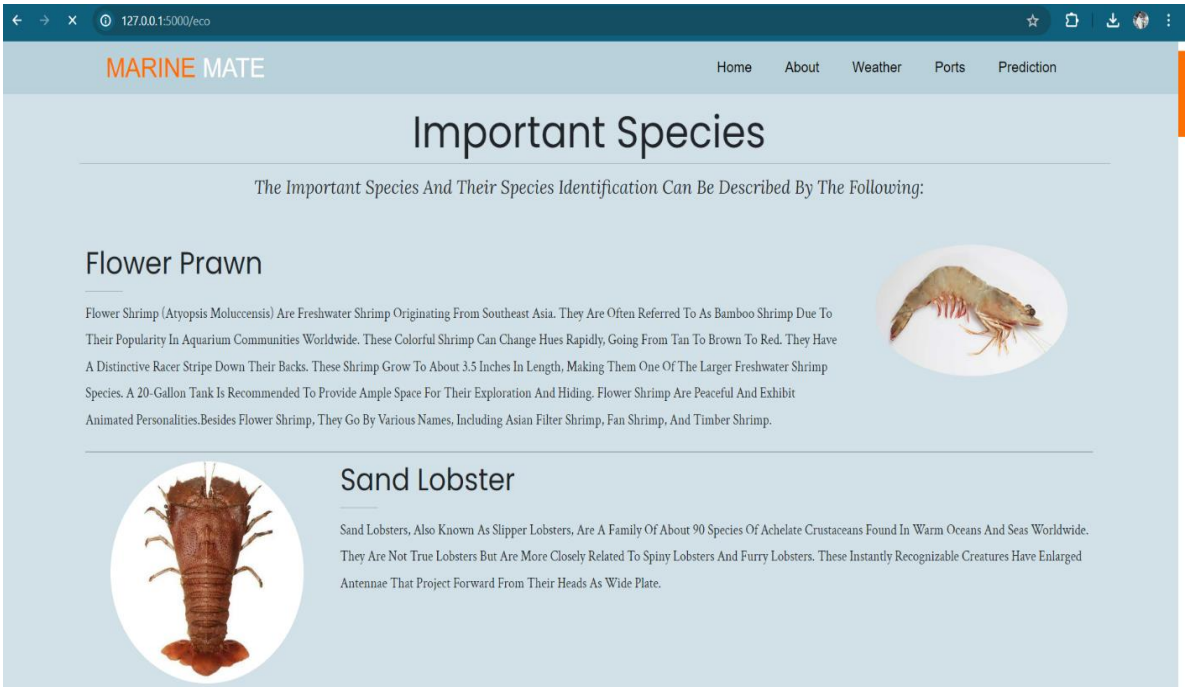


FIGURE 5.5 INITIAL UI OF THE IMPORTANT SPECIES.

6.The screenshot of the initial user interface of endangered species.

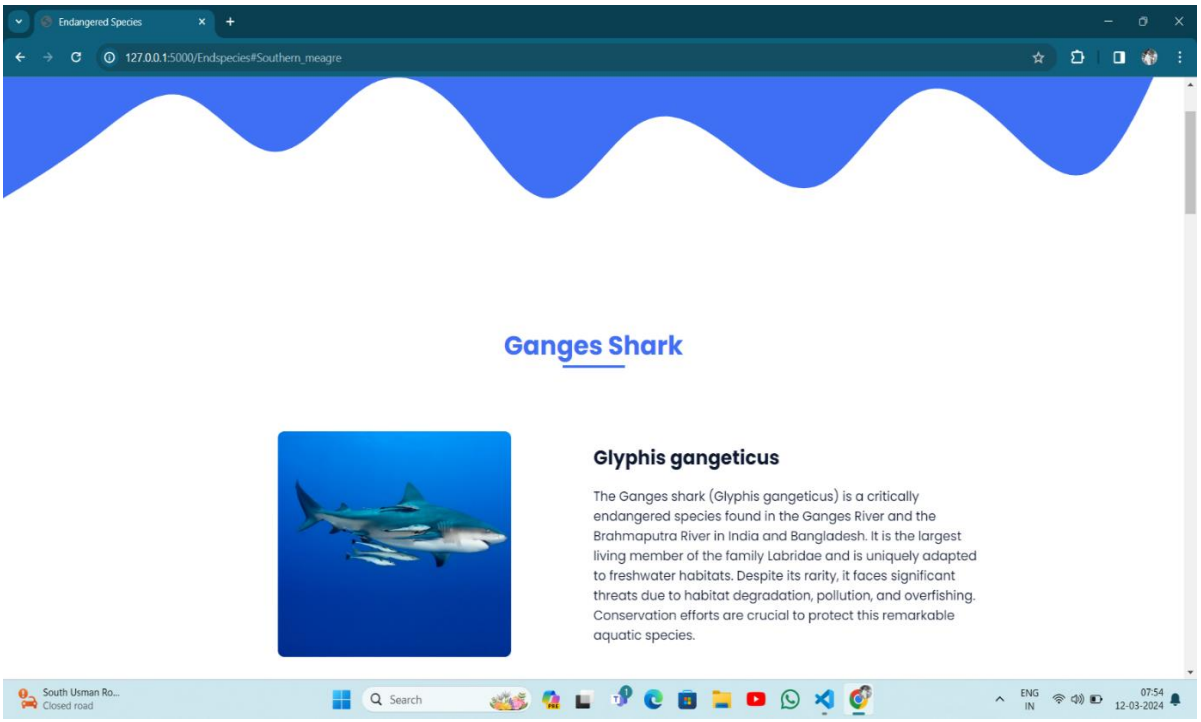


FIGURE 5.6 INITIAL UI OF THE ENDANGERED SPECIES.

7.The screenshot of the initial user interface of fishing information.

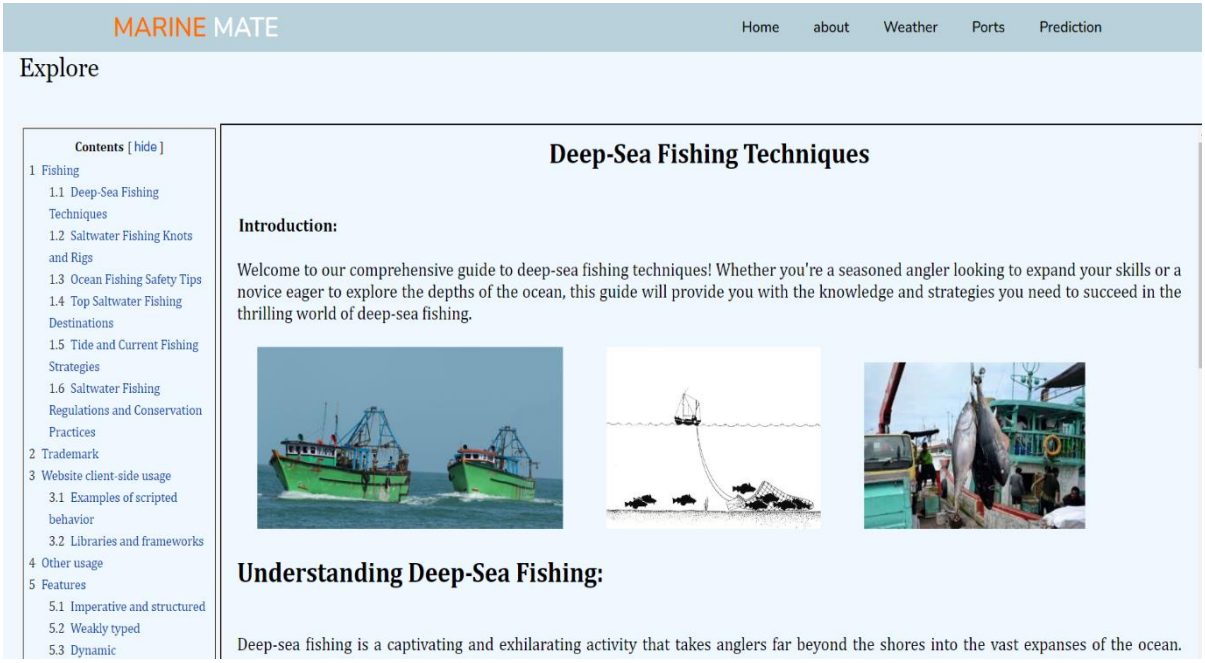


FIGURE 5.7 INITIAL UI OF THE FISHING INFORMATION.

8.The screenshot of the initial user interface of ports

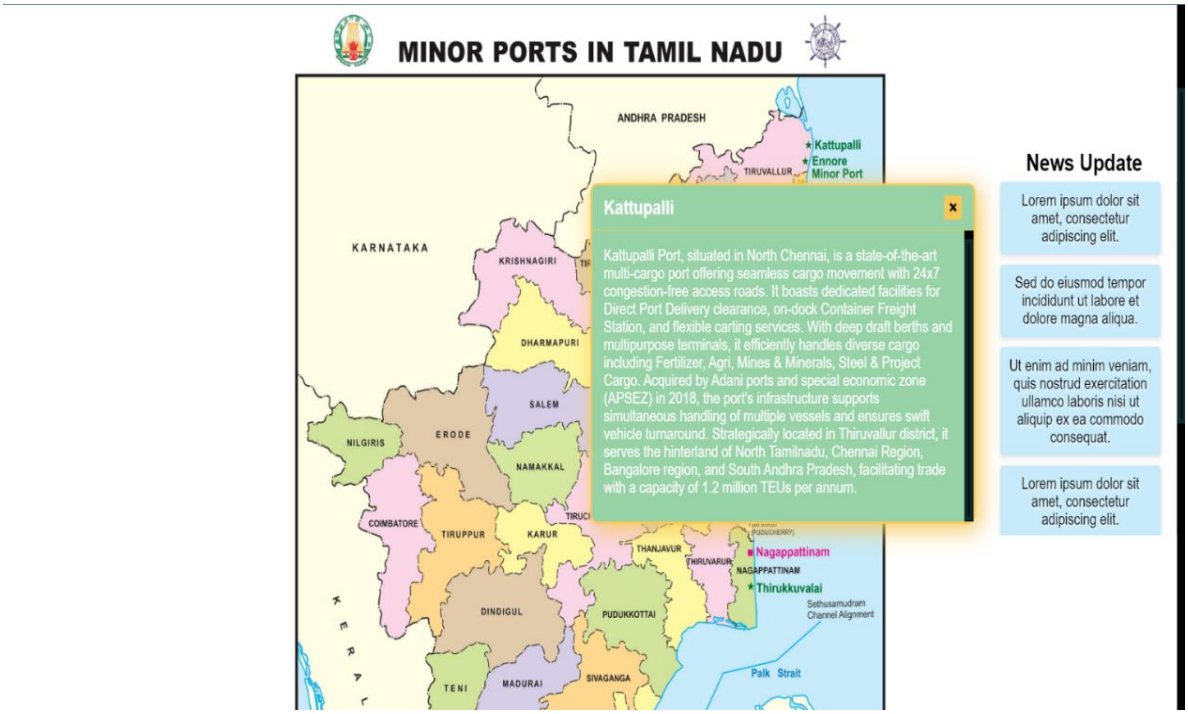


FIGURE 5.8 INITIAL UI OF THE PORT INFORMATION.

9.The screenshot of the user interface for weather forecast

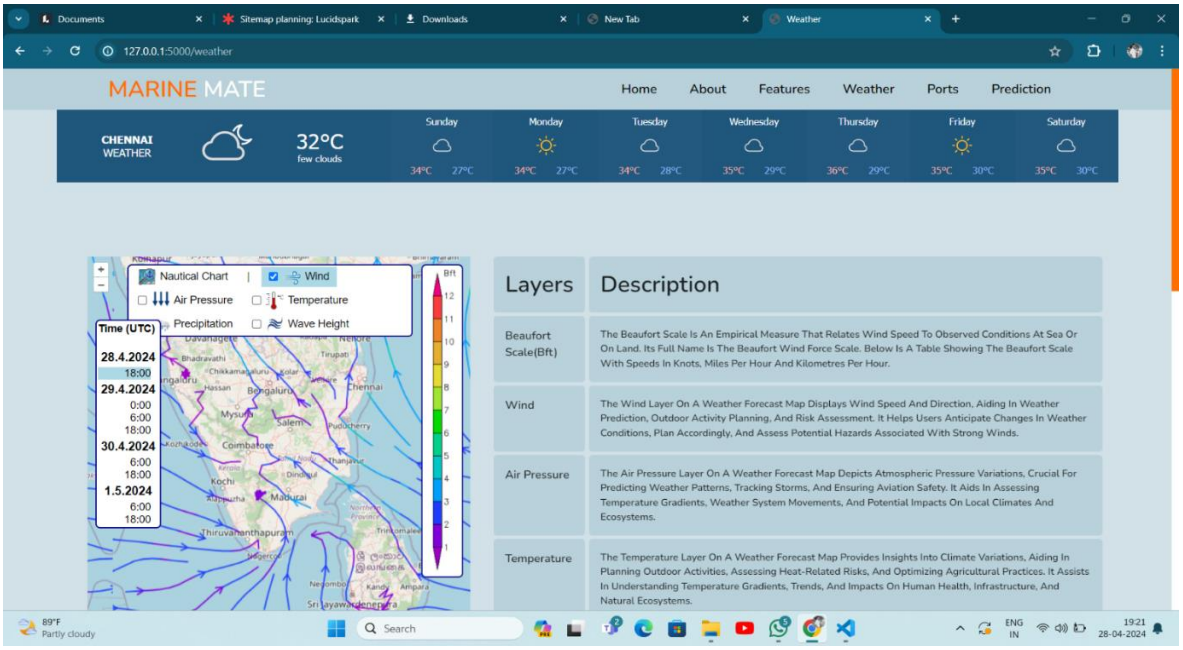


FIGURE 5.9 INITIAL UI OF THE WEATHER FORECAST.

10. The screenshot of the test image uploaded for prediction.

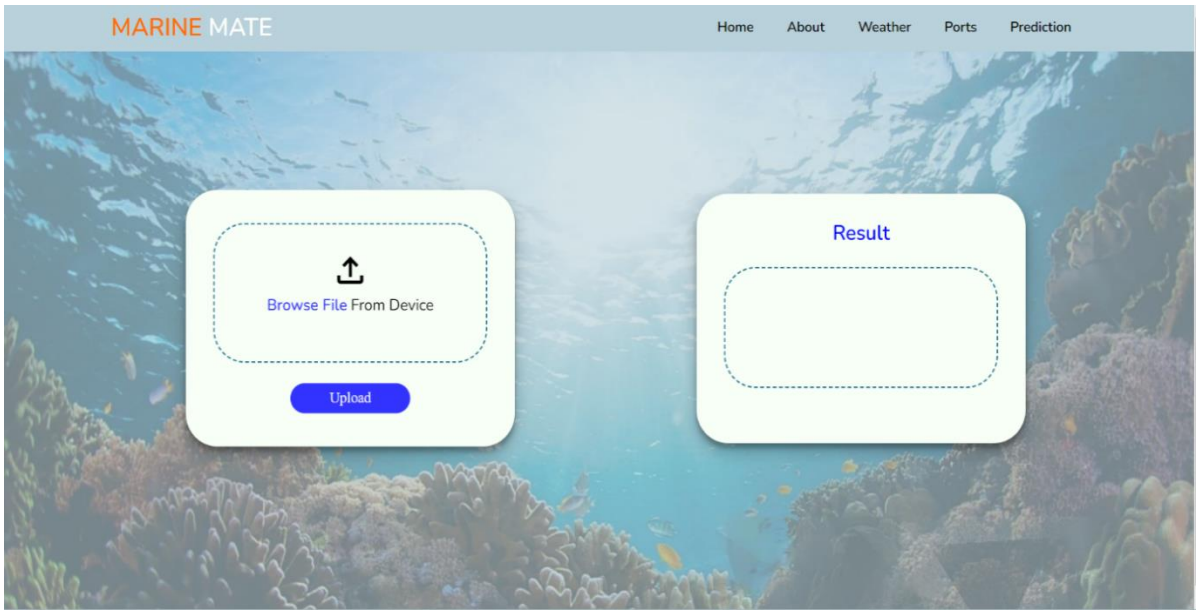


FIGURE 5.10 IMAGE UPLOADED FOR PREDICTION

11. The screenshot of the highly blurred test image predicted and classified to thecorresponding class.

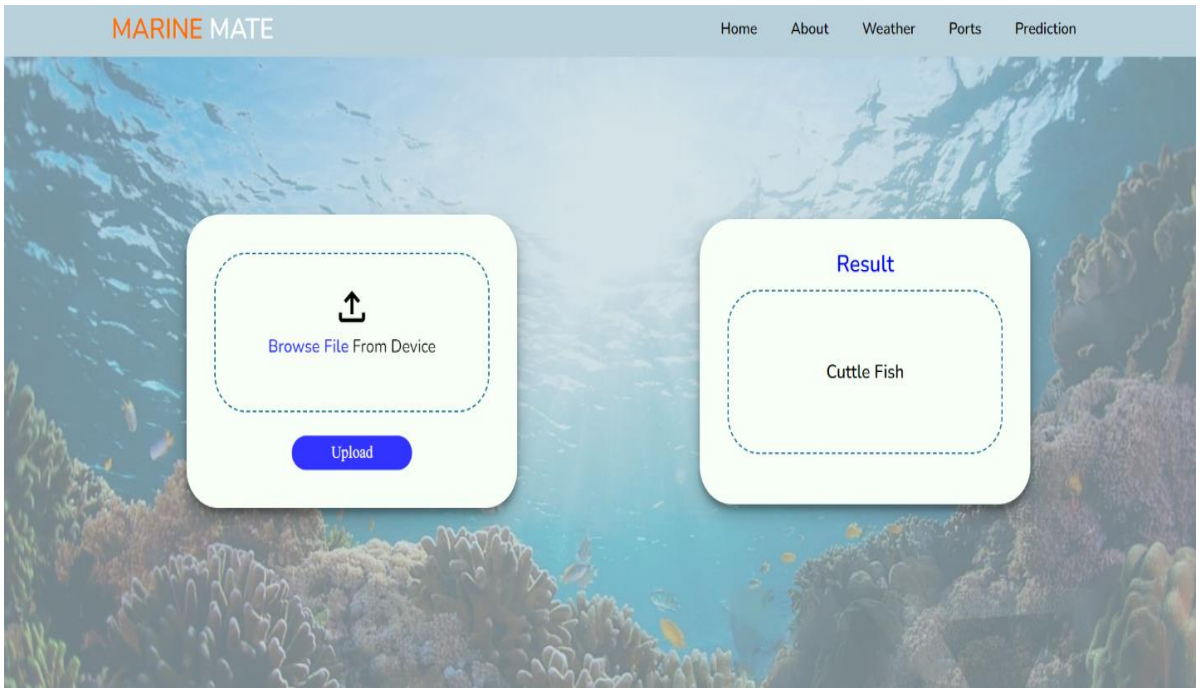


FIGURE 5.11 BLURRED IMAGE PREDICTED AND CLASSIFIE

5.11 SUMMARY

Detailed information about the various inbuilt python libraries for deep learning neural networks are given and the snapshots of the system developed is clearly mentioned.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In conclusion, our project has successfully proposed a comprehensive solution to address the critical need for efficient utilization of marine resources by integrating Artificial Intelligence (AI) and web-based technologies. Through the implementation of a rule-based chatbot and meticulously designed sections within the website, users can gain insights into marine ecosystems, fish species, and endangered species in the Bay of Bengal region.

Furthermore, the integration of interactive maps and real-time weather information enriches the user experience, allowing for exploration and analysis of marine environments with valuable geographical context and up-to-date conditions. Our platform serves as a valuable tool for stakeholders across various sectors, facilitating informed decision-making and contributing to the preservation of marine biodiversity and ecosystem health.

Moving forward, we aim to enhance the AI capabilities of our platform, expand and enrich our datasets, and develop a companion mobile application to further improve accessibility and usability. By fostering collaborative partnerships and engaging in educational outreach initiatives, we will continue to promote marine conservation and sustainable development, making strides towards a more resilient and thriving marine ecosystem.

6.2 FUTURE WORK

In future iterations, our project aims to further enhance the AI capabilities of our web-based platform, enabling more intuitive and personalized user interactions. We plan to expand and enrich our datasets, collaborating with marine research institutions to ensure the provision of the most accurate and up-to-date information on marine ecosystems.

Additionally, we will focus on developing a mobile application companion to offer users greater flexibility and accessibility. By integrating remote sensing technologies and IoT devices, we aim to provide real-time environmental data, enhancing our platform's utility for users and supporting informed decision-making.

Educational outreach and collaborative partnerships will remain central to our efforts, as we continue to promote marine conservation and sustainable development across diverse sectors. Through workshops, webinars, and joint initiatives, we will engage stakeholders and raise awareness about the importance of preserving marine biodiversity and ecosystem health.

APPENDICES

TRAIN

```
import pandas as pd

import os

from pathlib import Path

from sklearn.model_selection import train_test_split

from keras.preprocessing.image import ImageDataGenerator

from tensorflow import keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D,

MaxPooling2D

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.losses import SparseCategoricalCrossentropy

import tensorflow as tf


direc = Path('C:/Users/sivabala pc/Desktop/Prediction/dataset/Fish_Dataset')

filepaths = list(direc.glob(r'/*.png'))

Labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1],filepaths))
```

```
filepaths = pd.Series(filepaths, name='FilePaths').astype(str)

Labels = pd.Series(Labels, name='Labels').astype(str)


img_df = pd.merge(filepaths, Labels, right_index = True, left_index = True)

img_df = img_df[ img_df['Labels'].apply(lambda x: x[-2:]!='GT') ]
```

#Resampling it

```
img_df=img_df.sample(frac = 1).reset_index(drop=True)

img_df.head()

import matplotlib.pyplot as plt

f,a = plt.subplots(nrows=3, ncols=3,figsize=(13, 7),

                   subplot_kw={'xticks': [], 'yticks': []})


for i, ax in enumerate(a.flat):

    ax.imshow(plt.imread(img_df.FilePaths[i]))

    ax.set_title(img_df.Labels[i])


plt.tight_layout()

plt.show()
```

#shape of dataframe

```
print(f" Count of Rows : {img_df.shape[0]} \n Count of Columns :
{img_df.shape[1]} ")
#types of fishes
img_df['Labels'].value_counts(ascending=True)
#spiliting the data
train_ratio = 0.75
validation_ratio = 0.10
test_ratio = 0.15
x_train, x_test = train_test_split(img_df, test_size=1 - train_ratio)
x_val, x_test = train_test_split(x_test, test_size=test_ratio/(test_ratio +
validation_ratio))

print(f'Shape of Training Data : ',x_train.shape)
print(f'Shape of Testing Data : ',x_test.shape)
print(f'Shape of Validation Data : ',x_val.shape)
x_testt = x_test.copy()

#augmenting the data
img_datagen =
ImageDataGenerator(preprocessing_function=tf.keras.applications.mobilenet_v2.p
```

reprocess_input)

img_size=(224, 224)

x_train = img_datagen.flow_from_dataframe(dataframe = x_train,
x_col='FilePaths', y_col='Labels', target_size=img_size,
color_mode='rgb',class_mode='categorical',batch_size=32,seed=42)
x_test = img_datagen.flow_from_dataframe(dataframe = x_test, x_col='FilePaths',
y_col='Labels',
target_size=img_size,color_mode='rgb',class_mode='categorical',batch_size=32,se
ed=42)
x_val = img_datagen.flow_from_dataframe(dataframe = x_val, x_col='FilePaths',
y_col='Labels', target_size=img_size,
color_mode='rgb',class_mode='categorical',batch_size=32,seed=42)

#**creating the model**

```
fc_model = tf.keras.models.Sequential([
  tf.keras.layers.experimental.preprocessing.Rescaling(1./255),
  tf.keras.layers.Flatten(input_shape=(113,)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
```

```
    tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(9,activation="softmax")
])


fc_model.compile(optimizer="adam",
            loss="categorical_crossentropy",
            metrics=["accuracy"])
```

#training the model

```
Callback = tf.keras.callbacks.EarlyStopping(monitor='accuracy', patience=3)
mo_fit = fc_model.fit(x_train,
              validation_data = x_val,
               epochs = 10, callbacks=Callback)
fc_model.save("fish1.h5")
test_accuracy = fc_model.evaluate(x_test)[1] * 100
print('Test accuracy is : ',test_accuracy, '%' )
accuracy = mo_fit.history['accuracy']
loss = mo_fit.history['loss']
```

```
validation_loss = mo_fit.history['val_loss']

validation_accuracy = mo_fit.history['val_accuracy']

plt.figure(figsize=(17, 7))

plt.subplot(2, 2, 1)

plt.plot(range(10), accuracy, label='Training Accuracy')

plt.plot(range(10), validation_accuracy, label='Validation Accuracy')

plt.legend(loc='upper left')

plt.title('Accuracy : Training Vs Validation ')

plt.subplot(2, 2, 2)

plt.plot(range(10), loss, label='Training Loss')

plt.plot(range(10), validation_loss, label='Validation Loss')

plt.title('Loss : Training Vs Validation ')

plt.legend(loc='upper right')

plt.show()
```

ROUTING AND PREDICTION:

```
import numpy as np

import os

from tensorflow.keras.preprocessing import image

from flask import Flask, render_template, request, flash, session

from keras.models import load_model
```

```
from werkzeug.utils import secure_filename

model1=load_model("C:/Users/sivabala pc/Desktop/Prediction/body1.h5")

print(model1.summary())

app = Flask(_name_)

app.secret_key = 'your_secret_key'

@app.route('/')

def index():

    if not session.get('logged_in'):

        return render_template('log.html')

    else:

        return render_template('index.html')

@app.route('/login', methods=['POST'])

def do_admin_login():

    if request.form['password'] == 'siva@123' and request.form['username'] ==

'admin':

        session['logged_in'] = True

    else:

        flash('wrong password!')

    return index()
```

```
@app.route('/weather')
def weather():
    return render_template('weather.html')
@app.route('/prediction')
def prediction()
    return render_template('prediction.html')
@app.route('/eco')
def eco():
     return render_template('eco.html')
@app.rote('/Endspecies')
def endspecies():
     return render_template('Endspecies.html'
@app.route('/ports')
def ports():
     return render_template('ports.html'
@app.route('/maps')
def maps():
     return render_template('maps.html')
@app.route('/fishing')
def fishing():
```

```
    return render_template('fishing.html')
@app.route('/result', methods=['post'])
def result():
    print('Received POST request')
    if request.method == 'POST':
        print('Processing request')
        f = request.files['image'
        basepath = os.path.dirname(_file_
        file_path =  os.path.join(basepath, 'uploads',
secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(224, 224))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        print('Shape of input image:', x.shape)
        print(img
        prediction1 = np.argmax(model1.predict(x))
      #    index1=['Black Sea Sprat', 'blood_red_snapper','blotched_croaker'
,'cuttle_fish' ,'flower_prawn' ,'Gilt-Head Bream' ,'guitar_fish',
'hammer_head_shark', 'Hourse Mackerel', 'malabar_reef_cod' ,'mud_crab' ,'Red
```

```
Mullet' ,'Red Sea Bream' ,'rock_lobster','sand_lobster','Sea Bass' ,'Shrimp','squid'
,'sting_ray' ,'Striped Red Mullet' ,'Trout',"]
        index1 =
['','blood_red_snapper','cuttle_fish','flower_prawn','guitar_fish','hammer_head_shar
k','malabar_reef_cod','mud_crab','rock_lobster','sand_lobster','squid','sting_ray']

        res1 = index1[prediction1]

        result1=format(str(res1))
        if result1 == "Black Sea Sprat":
            value = "Black Sea Sprat"
        elif result1 == "blood_red_snapper":
            value = "Blood red snapper"
        elif result1 == "blotched_croaker":
            value = "blotched_croaker"

        elif result1 == "cuttle_fish":
            value = "Cuttle fish"
        elif result1 == "flower_prawn":
            value = "Flower prawn"
```

```
elif result1 == "Gilt-Head Bream":
    value = "Gilt-Head Bream"


elif result1 == "guitar_fish":
    value = "Guitar fish"
elif result1 == "hammer_head_shark":
    value = "Hammer head shark"
elif result1 == "Hourse Mackerel":
    value = "Hourse Mackerel"
elif result1 == "malabar_reef_cod":
    value = "Malabar reef cod"
elif result1 == "mud_crab":
    value = "Mud crab"
elif result1 == "Red Mullet":
    value = "Red Mullet"
elif result1 == "Red Sea Bream":
    value = "Red Sea Bream"
elif result1 == "rock_lobster":
    value = "Rock lobster"
elif result1 == "sand_lobster":
```

```
        value = "Sand lobster"
    elif result1 == "Sea Bass":
        value = "Sea Bass"
    elif result1 == "Shrimp":
        value = "Shrimp"
    elif result1 == "squid":
        value = "Squid"
    elif result1 == "sting_ray":
        value = "Sting ray"
    elif result1 == "Striped Red Mullet":
        value = "Striped Red Mullet"
    elif result1 == "Trout":
        value = "Trout"
    else:
        value = "Not Found"
    print('prediction:',value)


    return render_template('prediction.html',prediction=value)
```

```
if _name_ == '_main_':
    app.secret_key = os.urandom(12)
    app.run(debug=True)
```

CHATBOT

```
let chatbotVisible = false;
        function toggleChatbot() {
            chatbotVisible = !chatbotVisible;
         const chatbotPopup = document.getElementById('chatbotPopup');
            if (chatbotVisible) {
                chatbotPopup.classList.add('show-chatbot');
            } else {
                chatbotPopup.classList.remove('show-chatbot');
            }
        }

        function showChatbot() {
            const chatbotPopup = document.getElementById('chatbotPopup');
            if (window.scrollY > 100) {
```

```
            chatbotPopup.classList.add('show-chatbot');
        } else {
            chatbotPopup.classList.remove('show-chatbot');
        }
    }
    // const userMessage = [
    // ["fish prediction accuracy"],
    // ["your fish prediction"]
    // ];
    // const botReply = [
    // ["accuracy of prediction is:87%"],
    // ["Our fish prediction provides name of species"]
    // ];
    const userMessage = [
// Fish Prediction Section
["information about fish prediction", "about fish prediction"],
["accuracy","fish prediction accuracy"],
//endangere species
["endangered species","about endangered species"],
["how to access endangered species","where is endangered species"],
```

```
    ["endangered species list","tamilNadu endangered species"],


    // Maps Section
    ["maps", "maps on your website"],
    ["What types of maps are available on your site?"],


    // Weather Section
    ["guide to weather","weather now"],
    ["weather", "weather on your website"],
    ["weather in your website", "weather section on your website."],
    ["How often is the weather data updated?"],
    //port qes
    ["ports","about ports","about ports section"],
    // General Questions
    ["topics does your website cover", "overview of your website"],
    ["How to access sections"],
    ["sections","section","list of sections"]
];


const botReply = [
```

// Fish Prediction Section Replies

["Our fish prediction section provides forecasts based on oceanographic data and predictive models."],

["The accuracy of fish prediction is: approxmately 87%"],

//endangered species

["Our website provides a list of endangered species.","Information about endangered species in Tamil Nadu is available in our website"],

["first click 'endangered species' on navigation bar , in that page list of endangered species is availabale"],

["sorry , i can't able give all the species list, go to endangered species section"],

// Maps Section Replies

["Yes, we offer a variety of ocean maps including "],//not finished

["You can find maps depicting ocean currents, bathymetry, and marine biodiversity on our website."],

// Weather Section Replies

["first click 'Weather' on navigation bar \n after the page was opened scroll down there will be some section ,enter cordinates or place name finally get weather report!! "],

["we provide information on ocean weather conditions, including temperature, currents, and wind patterns."],

["Our website provides information on ocean weather conditions, including temperature, currents, and wind patterns."],

["Weather data on our website is updated regularly, usually on a daily basis."],

//ports

["This section gives information about major ports ,minor ports and news of the ports."],

// General Questions Replies

["Our website covers various topics related to oceanography, including fish prediction, maps,ports,enadangered species, important species and weather information."],

["You can navigate through different sections of our website using the menu bar or navigation links provided."],

["The sections in this website are -Endangered species, -Important species, -Maps, -Ports, -Fish prediction"]

];

```
const alternative = [

"Same here, dude.",

"That's cool! Go on...",

"Dude...",

"Ask something else...",

"Hey, I'm listening..."

];


const synth = window.speechSynthesis;


function voiceControl(string) {

let u = new SpeechSynthesisUtterance(string);

u.text = string;

u.lang = "en-aus";

u.volume = 10;

u.rate = 1;

u.pitch = 1;

synth.speak(u);

}
```

```
function sendMessage() {
const inputField = document.getElementById("input");
let input = inputField.value.trim();
input != "" && output(input);
inputField.value = "";
}
document.addEventListener("DOMContentLoaded", () => {
const inputField = document.getElementById("input");
inputField.addEventListener("keydown", function (e) {
  if (e.code === "Enter") {
    let input = inputField.value.trim();
    input != "" && output(input);
    inputField.value = "";
  }
});
});

function output(input) {
let product;
```

```
let text = input.toLowerCase().replace(/[^\w\s\d]/gi, "");

text = text
  .replace(/[\W_]/g, " ")
  .replace(/ a /g, " ")
  .replace(/i feel /g, "")
  .replace(/whats/g, "what is")
  .replace(/please /g, "")
  .replace(/ please/g, "")
  .trim();

let comparedText = compare(userMessage, botReply, text);

product = comparedText
  ? comparedText
  : alternative[Math.floor(Math.random() * alternative.length)];
addChat(input, product);
}

function compare(triggerArray, replyArray, string) {
```

```
let item;
for (let x = 0; x < triggerArray.length; x++) {
  for (let y = 0; y < replyArray.length; y++) {
    if (triggerArray[x][y] == string) {
      items = replyArray[x];
      item = items[Math.floor(Math.random() * items.length)];
    }
  }
}
//containMessageCheck(string);
if (item) return item;
else return containMessageCheck(string);
}
function containMessageCheck(string) {
let expectedReply = [
  [
    "Good Bye, dude",
    "Bye, See you!",
    "Dude, Bye. Take care of your health in this situation."
  ],
```

```
  ["Good Night, dude", "Have a sound sleep", "Sweet dreams"],
  ["Have a pleasant evening!", "Good evening too", "Evening!"],
  ["Good morning, Have a great day!", "Morning, dude!"],
  ["Good Afternoon", "Noon, dude!", "Afternoon, dude!"]
];
let expectedMessage = [
  ["bye", "tc", "take care"],
  ["night", "good night"],
  ["evening", "good evening"],
  ["morning", "good morning"],
  ["noon"]
];
let item;
for (let x = 0; x < expectedMessage.length; x++) {
  if (expectedMessage[x].includes(string)) {
    items = expectedReply[x];
    item = items[Math.floor(Math.random() * items.length)];
  }
}
return item;
```

```
}

function addChat(input, product) {

const mainDiv = document.getElementById("message-section");

let userDiv = document.createElement("div");

userDiv.id = "user";

userDiv.classList.add("message");

userDiv.innerHTML = <span id="user-response">${input}</span>;

mainDiv.appendChild(userDiv);


let botDiv = document.createElement("div");

botDiv.id = "bot";

botDiv.classList.add("message");

botDiv.innerHTML = <span id="bot-response">${product}</span>;

mainDiv.appendChild(botDiv);

var scroll = document.getElementById("message-section");

scroll.scrollTop = scroll.scrollHeight;

voiceControl(product);

}
```

REFERENCES

[1] EMSA Annual Overview of marine casualties and incidents, Accessed 12 April 2022. available at: http://www.emsa.europa.eu/news-a-presscentre/external-news/item/3734-annual-overview-of-marinecasualties-and-incidents-2019.html.

[2] J. G. Clark and M. Kronengold, “Long-period fluctuations of CW signals in deep and shallow water,” J. Acoust. Soc. Amer., vol. 56, no. 4, pp. 1971-1983, Oct. 1974. C. Garrett and W. Munk, “Space-time scales of internal waves: A progress report,” J. Geophys. Res., vol. 30, no. 3, pp. 291-297, Jan. 1975

[3] Alagarswami, K. (1981). Prospects for coastal aquaculture in India. CMFRI Bulletin, 30, pp.83-86.

[4] Bostock, J., Mcandrew, B., Richards, R., Jauncey, K., Telfer, T., Lorenzen, K., Little, D., Ross, L., Handisyde, N., Gatward, I. and Corner, R. (2010). Aquaculture: global status and trends. Philosophical Transactions of the Royal Society B: Biological Sciences, 365(1554), pp.2897-2912.

[5] James, P.S.B.R., Jhingran, A.G. and Rao, K.M. (1986). Present status and future scope for fish production in cages and enclosures in India. Indian Journal of Animal Sciences, 56(3), pp.453-458.

[6] Sathiadhas, R., Panikkar, K.K.P., Satyavan, U.K. and Jacob, T. (1989). Economic Evaluation of Paddy-Prawn Integrated Farming in Kerala. Seafood Export Journal, 21(11), pp.9-21

[7] Thakur, N., & Jain, S. K. (2021). Web-based Technologies for Marine Resource Management

[8] Satyaveer, Ranjanna, K. B. and Verma, D. K. 2022. Present Trend and Future Scope of Coastal Aquaculture in India. Vigyan Varta 3(6): 121-124.

[9] https://www.nio.res.in/

[10] https://www.chennaiport.gov.in/home

[11] https://www.chatbase.co/

[12] https://mpeda.gov.in/fishers/ , First Record of Eleven Reef Inhabiting Fishes from Tamil Nadu Coast of India, Bay of Bengal | Proceedings of the Zoological Society

[13] Sahu, S. K., & Satapathy, S. C. (2019). A Comprehensive Review on Marine Biodiversity Conservation Techniques

[14] Singh, A., & Ghosh, S. (2020). Applications of Artificial Intelligence in Marine Science

[15] Cooley, James W. and Tukey, John W., “An algorithm for the machine calculation of complex Fourier series,” Math. Comput. 19, 297–301 (1965).