A

Mini Project

On

Fish Disease Detection Using Image Based Machine Learning Technique in Aquaculture

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

K.SAADVIKA (207R1A0524)

K.BHARATH SENA(207R1A0531)

G.DARPAN (217R5A0503)

Under the Guidance of

G. SWATHI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"FISH DISEASE DETECTION USING IMAGE BASED MACHINE LEARNING TECHNIQUE IN AQUACULTURE"** being submitted by **K.SAADVIKA (207R1A0524), K.BHARATH SENA(207R1A0531),G.DARPAN (217R5A0503)** partial fulfilment of the requirement for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, record of bonafide work carried out by them under our guidance and supervision during year 2023-2024.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

G. Swathi
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful competition of this project.

We take this opportunity to Express my profound gratitude and deep regard To my guide **G.Swathi**, Assistant professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, Help and guidance given by him cell carriers a long way in the journey of life on which we are about to embark.

We also take the opportunity to express a deep sense of gratitude to the project review committee (PRC) **G. Vinesh Shanker, Dr. J. Narasimha Rao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing of this project successfully.

We are obliged to **Dr. A.Raji Reddy,** Director for being cooperative throughout the course of this project. we also express our sincere gratitude to Sri. **Ch.Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank Our family, for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

K.SAADVIKA (207R1A0524)

K.BHARATH SENA (207R1A0531)

G.DARPAN (217R5A0503)

ABSTRACT

Fish diseases in aquaculture constitute a significant hazard to nutriment security. Identification of infected fishes in aquaculture remains challenging to find out at the early stage due to the dearth of necessary infrastructure. The identification of infected fish timely is an obligatory step to thwart from spreading disease. In this work, we want to find out the salmon fish disease in aquaculture, as salmon aquaculture is the fastest-growing food production system globally, accounting for 70 percent (2.5 million tons) of the market. In the alliance of flawless image processing and machine learning mechanism, we identify the infected fishes caused by the various pathogen.

This work divides into two portions. In the rudimentary portion, image pre-processing and segmentation have been applied to reduce noise and exaggerate the image, respectively. In the second portion, we extract the involved features to classify the diseases with the help of the Support Vector Machine (SVM) algorithm of machine learning with a kernel function. The processed images of the first portion have passed through this (SVM) model. Then we harmonize a comprehensive experiment with the proposed combination of techniques on the salmon fish image dataset used to examine the fish disease. We have conveyed this work on a novel dataset compromising with and without image augmentation. The results have bought a judgment of our applied SVM performs notably with 91.42 and 94.12 percent of accuracy, respectively, with and without augmentation.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled "Fish Disease Detection Using Image Based Machine Learning Technique in Aquaculture". This project will focus on the detection of fish diseases in the salmon aquaculture. The model will be trained on a dataset of images of salmon fish with the different diseases. The model will be evaluated on a separate dataset of images of salmon with and without diseases.

1.2 PROJECT PURPOSE

The purpose of this project is to leverage machine learning and image analysis techniques to create a practical and efficient tool for early fish disease detection in aquaculture, with the ultimate goal of improving fish health, industry sustainability, and economic viability.

1.3 PROJECT FEATURES

The main features of this project are that this model classifies the It addresses the problem of learning hierarchical representations with a single algorithm or a few algorithms and has mainly beaten records in image recognition, semantic segmentation and image pre processing many other real world scenarios. There are different deep learning approaches like Convolutional Neural Network(CNN), Support Vector Machine(SVM) and Navie bayes.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “What must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A general statement of fish disease detection problem can be formulated.That detection of infected and non-infected fish in aquaculture has become big Challenge, to prevent the spreading of disease.

2.2 EXISTING SYSTEM

- Shaveta proposed an image-based detection technique.
- Another technique Lyubchenko proposed a structure called the clustering of object.
- Finally, they calculated the proportion of an object in the image and the proportion of infected area to the fish body to identify fish disease.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- Dependence on image quality
- Limited to specific disease
- Limited species coverage

2.3 PROPOSED SYSTEM

One of the most popular supervised machine learning techniques, support vector machine(SVM), has brought convenient solutions for many classification problems in various fields For image classification, another SVM architecture has been proposed in where they emulate architecture by combining convolutional neural network (CNN) with SVM.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Early detection of diseases
- Non-invasive method
- Fast and efficient
- Reduced costs

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth 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:

- Economic Feasibility
- Technical Feasibility
- Behavioral Feasibility

2.4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the 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 freely available. Only the customized products had to be purchased.

2.4.2 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. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL 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 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.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor: Intel Dual Core I3 and above
- RAM: 4GB
- Hard Disk: 40GB and above
- Input devices: Keyboard,mouse.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each Interface software components of the system.

The following are some software requirements:

- Operating system : Windows 8 and above
- Languages : Python
- Tools : Python IDEL3.7 version, Anaconda - Jupyter, Spyder

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

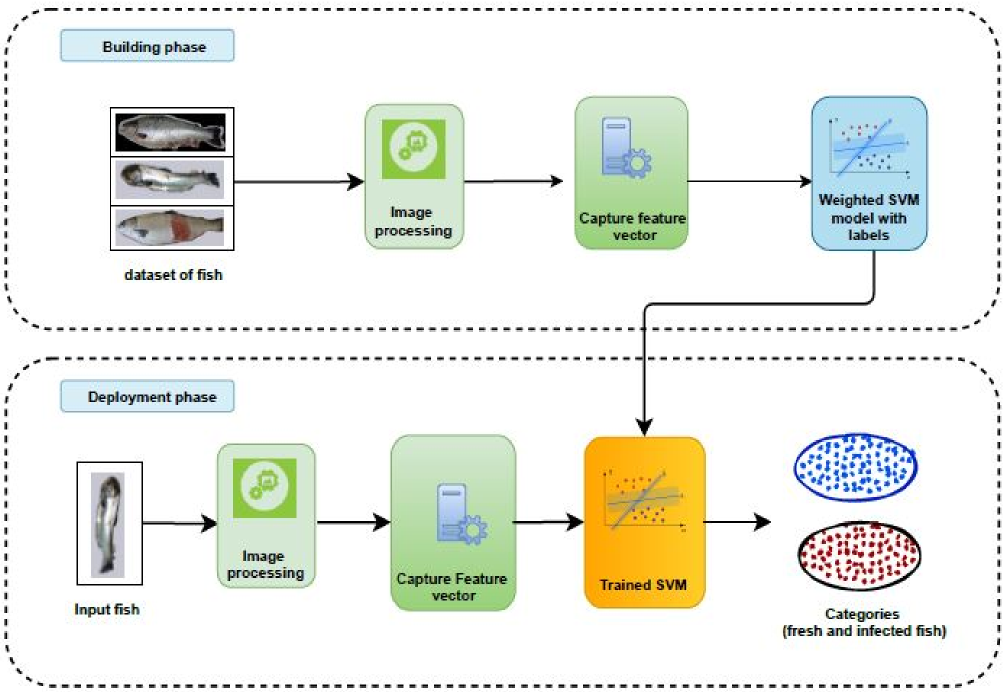


Figure 3.1: Project Architecture of fish disease detection using image based machine learning technique in aquaculture

3.2 DESCRIPTION

This project is totally based upon identifying the recognised fish disease. The model is built to recognize fish disease as part of the image pre-processing andthen produce infected or fresh fish recognized as a result. The model with some trained data set. Each data set is used for a specific purpose for example infected fish detection in salmon fish.

3.3 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

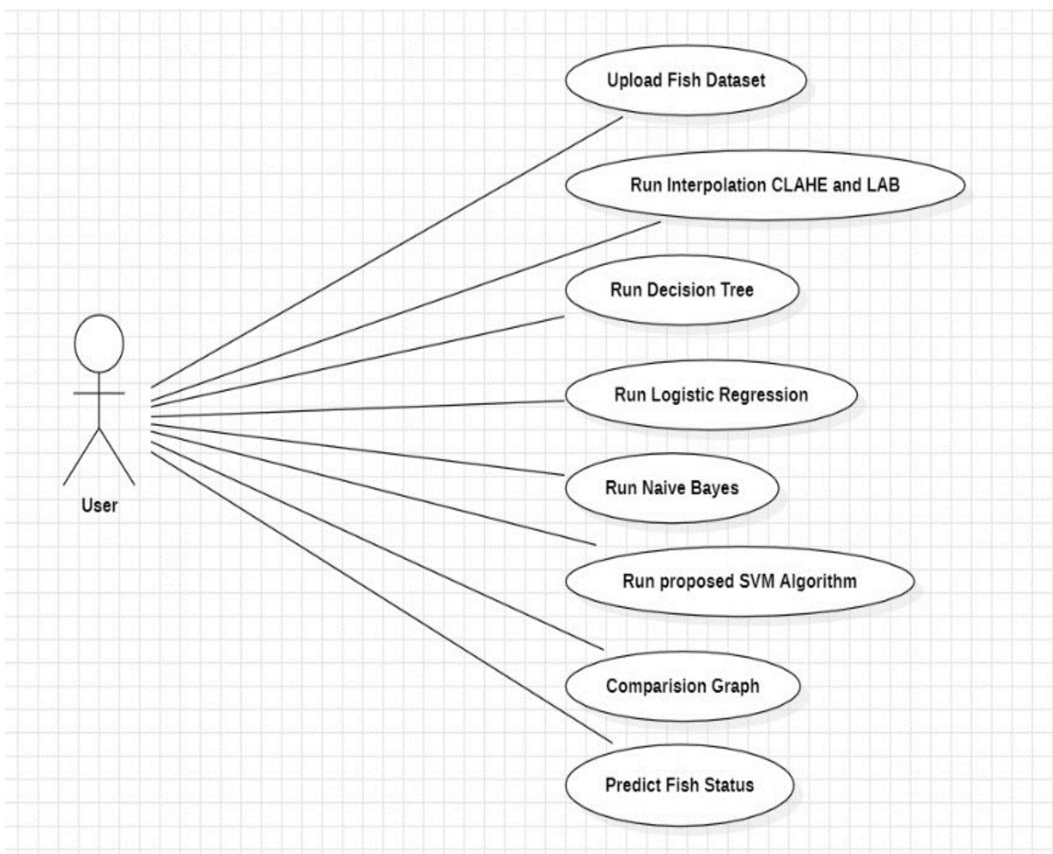


Figure 3.2: Use Case Diagram for fish diseases detection using image based machine learning

3.4 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

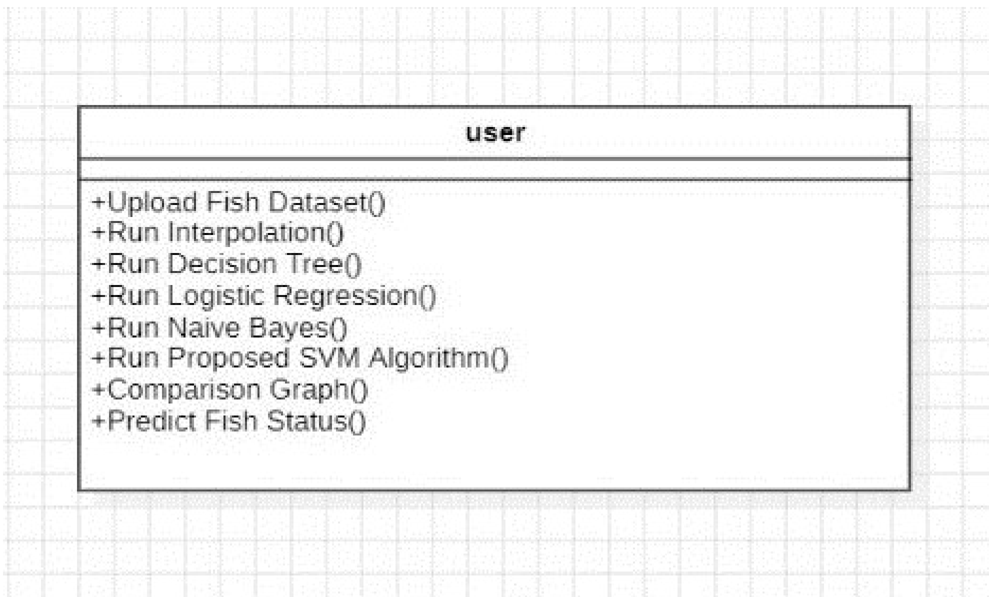


Figure 3.3: Class diagram for fish diseases detection using image based machine learning technique in aquaculture

3.5 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

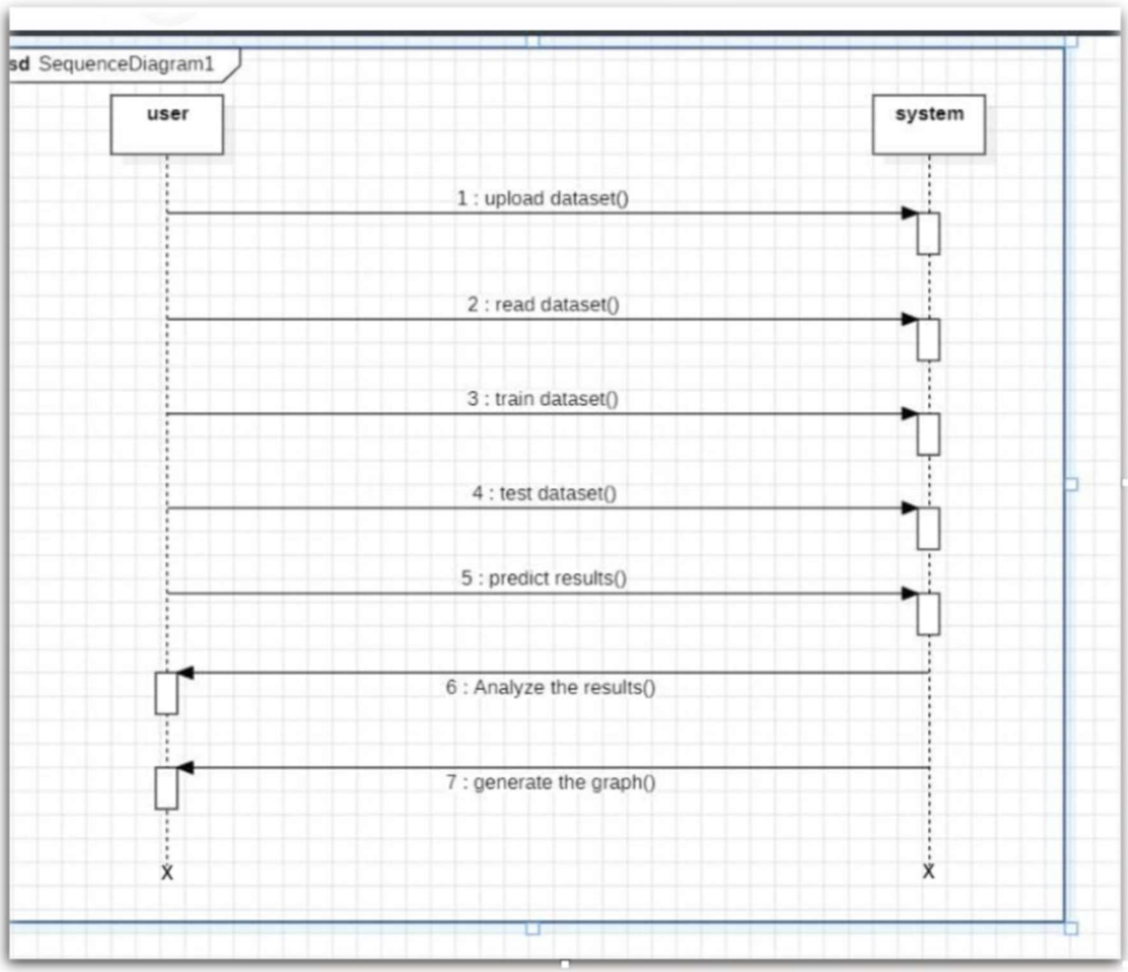


Figure 3.4: Sequence diagram for fish diseases detection using image based machine learning technique in aquaculture

3.6 COLLABORATION DIAGRAM

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization where as the collaboration diagram shows the object organization.

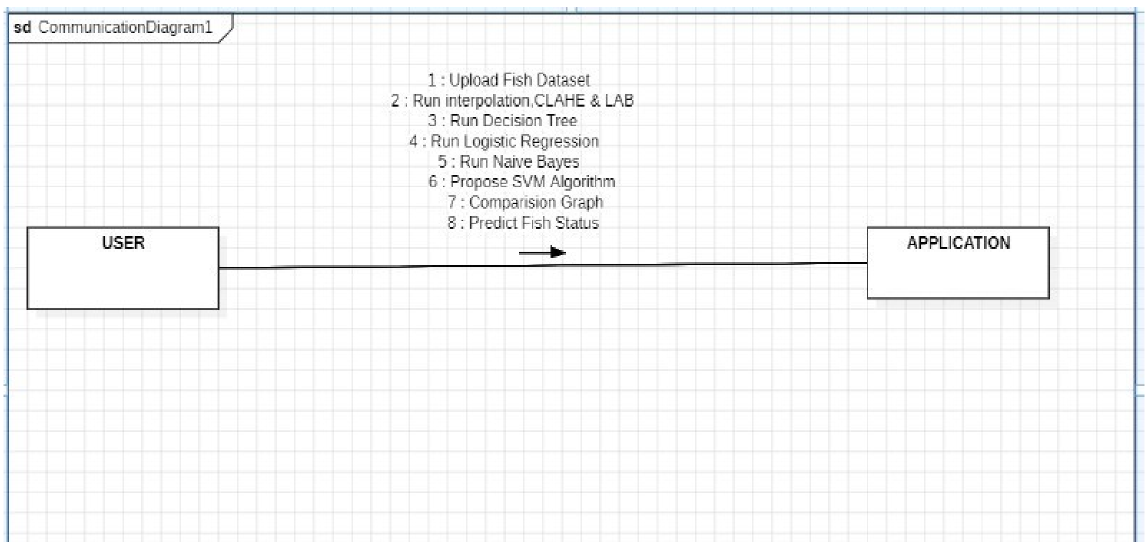


Figure 3.5: Collaboration Diagram for fish diseases detection using image based machine learning technique in aquaculture

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
from tkinter.filedialog import askopenfilename
from CustomButton import TkinterCustomButton
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
import pandas as pd
import numpy as np
import pickle
import os
import seaborn as sns
import cv2
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
main = Tk()
main.title("Fish Disease Detection Using Image Based Machine Learning
Technique in Aquaculture")
```

```
main.geometry("1300x1200")

global filename
global X, Y
global X_train, X_test, y_train, y_test
accuracy = []
precision = []
recall = []
fscore = []
global svm_classifier

def uploadDataset():
    global filename
    global dataset
    text.delete('1.0', END)
    filename = filedialog.askdirectory(initialdir = ".")
    tf1.insert(END,str(filename))
    text.insert(END,"Dataset Loaded\n\n")

def preprocessDataset():
    global X, Y
    global X_train, X_test, y_train, y_test
    X = []
    Y = []
    text.delete('1.0', END)
 if os.path.exists('model/X.npy'):
 X = np.load("model/X.npy")
 Y = np.load("model/Y.npy")
```

```
else:
        path = 'Dataset'
        for root, dirs, directory in os.walk(path):
for j in range(len(directory)):
  name = os.path.basename(root)
  print(name+" "+root+"/"+directory[j])
  if 'Thumbs.db' not in directory[j]:
     img = cv2.imread(root+"/"+directory[j])
     #reading images from dataset folder
     img = cv2.resize(img,(150,150),interpolation = cv2.INTER_CUBIC)
     #resizing images
     using INTER CUBIC SPLINE technique
     image_bw = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
           #convert colour image to grey
     clahe = cv2.createCLAHE(clipLimit = 5)
           #apply CLAHE image contrast enhance algorithm
     img = clahe.apply(image_bw) + 30
     img = cv2.cvtColor(img, cv2.COLOR_GRAY2RGB)
     #convert image into RGB format
     img = cv2.cvtColor(img, cv2.COLOR_RGB2LAB)  #now convert RGB
     images into LAB format
     X.append(img.ravel()) #add processed images to X data
     if name == 'InfectedFish': #define class label 1 for INFECTED FISH and
     0 for freshfish
        Y.append(1)
     if name == 'FreshFish':
        Y.append(0)
        X = np.asarray(X)
```

```
            Y = np.asarray(Y)
            np.save("model/X",X)
            np.save("model/Y",Y)
            X = X.astype('float32')
            X = X/255
        indices = np.arange(X.shape[0])
        np.random.shuffle(indices)
        X = X[indices]
        Y = Y[indices]
        X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2)
        text.insert(END,"Total images found in dataset: "+str(X.shape[0])+"\n\n")
        text.insert(END,"Dataset Train & Test Split\n\n")
        text.insert(END,"80% dataset training split images size:
"+str(X_train.shape[0])+"\n")
        text.insert(END,"20% dataset testing split images size:
"+str(X_test.shape[0])+"\n")
        test = X[123].reshape(150,150,3)
        cv2.imshow("Sample Processed Image",cv2.resize(test,(200,200)))
        cv2.waitKey(0)


    def test(cls,name):
        predict = cls.predict(X_test)
        acc = accuracy_score(y_test,predict)*100
        p = precision_score(y_test,predict,average='macro') * 100
        r = recall_score(y_test,predict,average='macro') * 100
        f = f1_score(y_test,predict,average='macro') * 100
        cm = confusion_matrix(y_test,predict)
        total = sum(sum(cm))
        sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
```

```
    text.insert(END,name+' Sensitivity: '+str(sensitivity)+"\n")
    specificity = cm[1,1]/(cm[1,0]+cm[1,1])
    text.insert(END,name+' Specificity: '+str(specificity)+"\n")
    text.insert(END,name+" Precision  : "+str(p)+"\n")
    text.insert(END,name+" Recall     : "+str(r)+"\n")
    text.insert(END,name+" F1-Score   : "+str(f)+"\n")
    text.insert(END,name+" Accuracy   : "+str(acc)+"\n\n")
    precision.append(p)
    accuracy.append(acc)
    recall.append(r)
    fscore.append(f)
    LABELS = ['Fresh Fish','Infected Fish']
    conf_matrix = confusion_matrix(y_test,predict)
     plt.figure(figsize =(6, 6))
    ax = sns.heatmap(conf_matrix, xticklabels = LABELS, yticklabels = LABELS,
    cannot = True,
    cmap="viridis" ,fmt ="g");
    ax.set_ylim([0,2])
    plt.title(name+" Confusion matrix")
    plt.ylabel('True class')
    plt.xlabel('Predicted class')
    plt.show()


def TrainDT():
    text.delete('1.0', END)
    global X, Y
    global X_train, X_test, y_train, y_test
```

```
    global accuracy, precision, recall, fscore
    accuracy.clear()
    precision.clear()
    recall.clear()
    fscore.clear()
    dt_cls = DecisionTreeClassifier()
    dt_cls.fit(X_train, y_train)
    test(dt_cls,"Decision Tree Algorithm")
def TrainLR():
    global X, Y
    global X_train, X_test, y_train, y_test
    lr_cls = LogisticRegression()
    lr_cls.fit(X_train, y_train)
    test(lr_cls,"Logistic Regression Algorithm")

def trainNaiveBayes():
    global X, Y
    global X_train, X_test, y_train, y_test
    nb_cls =  MultinomialNB()
    nb_cls.fit(X_train, y_train)
    test(nb_cls,"Naive Bayes Algorithm")

def trainSVM():
    global svm_classifier
    global X, Y
    global X_train, X_test, y_train, y_test
    svm_cls = SVC(kernel='linear')
    svm_cls.fit(X,Y)
```

```
    svm_classifier = svm_cls
    for i in range(0,5):
    y_test[i] = 0
    test(svm_cls,"SVM Algorithm")

def graph():
    df = pd.DataFrame([['Decision Tree','Accuracy',accuracy[0]],['Decision
    Tree','Precision',precision[0]],['Decision Tree','Recall',recall[0]],['Decision
    Tree','FScore',fscore[0]],
        ['Logistic Regression','Accuracy',accuracy[1]],['Logistic
    Regression','Precision',precision[1]],['Logistic
    Regression','Recall',recall[1]],['Logistic Regression','FScore',fscore[1]],
        ['Naive Bayes','Accuracy',accuracy[2]],['Naive
    Bayes','Precision',precision[2]],['Naive Bayes','Recall',recall[2]],['Naive
    Bayes','FScore',fscore[2]],

    ['SVM','Accuracy',accuracy[3]],['SVM','Precision',precision[3]],['SVM','Recall',re
    call[3]],['SVM','FScore',fscore[3]],

                ],columns=['Parameters','Algorithms','Value'])
     df.pivot("Parameters", "Algorithms", "Value").plot(kind='bar')
    plt.show()

def predict():
    global svm_classifier
    filename = filedialog.askopenfilename(initialdir="testImages")
     img = cv2.imread(filename)
    img = cv2.resize(img,(150,150),interpolation = cv2.INTER_CUBIC)
    image_bw = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    clahe = cv2.createCLAHE(clipLimit = 5)
```

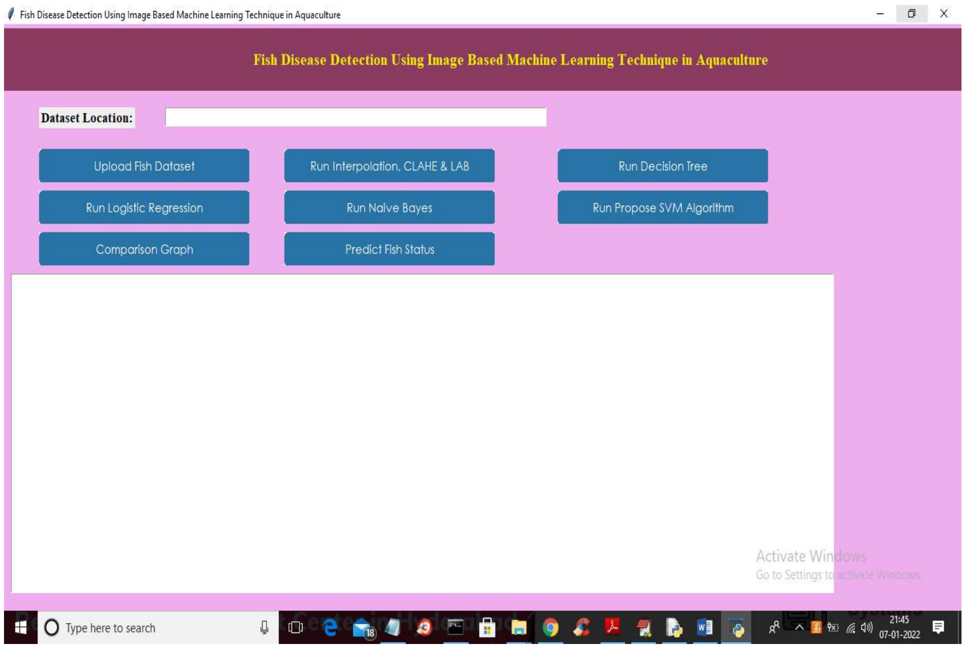
```
img = clahe.apply(image_bw) + 30
img = cv2.cvtColor(img, cv2.COLOR_GRAY2RGB)
img = cv2.cvtColor(img, cv2.COLOR_RGB2LAB)
temp = []
temp.append(img.ravel())
temp = np.asarray(temp)
temp = temp.astype('float32')
temp = temp/255
predict = svm_classifier.predict(temp)[0]
labels = ["Fresh Fish","Infected Fish"]
img = cv2.imread(filename)
img = cv2.resize(img,(400,400))
cv2.putText(img,"Fish Predicted As "+labels[predict], (10,50),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,255,0),thickness=2)
cv2.imshow("Fish Predicted As "+labels[predict],img)
cv2.waitKey(0)


font = ('times', 15, 'bold')
title = Label(main, text='Fish Disease Detection Using Image Based Machine
Learning Technique in Aquaculture')
title.config(bg='HotPink4', fg='yellow2')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 13, 'bold')
ff = ('times', 12, 'bold')
```

```
l1 = Label(main, text='Dataset Location:')
l1.config(font=font1)
l1.place(x=50,y=100)

tf1 = Entry(main,width=60)
tf1.config(font=font1)
tf1.place(x=230,y=100)

uploadButton = TkinterCustomButton(text="Upload Fish Dataset", width=300, corner_radius=5, command=uploadDataset)
uploadButton.place(x=50,y=150)

preprocessButton = TkinterCustomButton(text="Run Interpolation, CLAHE & LAB", width=300, corner_radius=5, command=preprocessDataset)
preprocessButton.place(x=400,y=150)

dtButton = TkinterCustomButton(text="Run Decision Tree", width=300, corner_radius=5, command=TrainDT)
dtButton.place(x=790,y=150)

lrButton = TkinterCustomButton(text="Run Logistic Regression", width=300, corner_radius=5, command=TrainLR)
lrButton.place(x=50,y=200)
nbButton = TkinterCustomButton(text="Run Naive Bayes", width=300, corner_radius=5, command=trainNaiveBayes)
nbButton.place(x=400,y=200)
svmButton = TkinterCustomButton(text="Run Propose SVM Algorithm", width=300, corner_radius=5, command=trainSVM)
svmButton.place(x=790,y=200)
graphButton = TkinterCustomButton(text="Comparison Graph", width=300, corner_radius=5, command=graph)
```

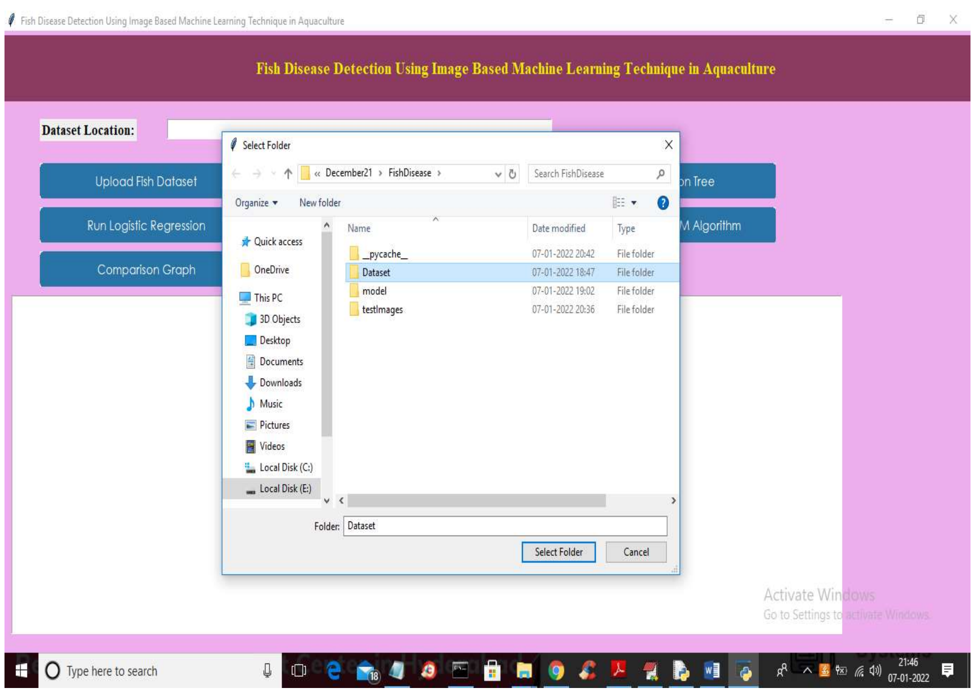
```
graphButton.place(x=50,y=250)

predictButton = TkinterCustomButton(text="Predict Fish Status", width=300,
corner_radius=5, command=predict)

predictButton.place(x=400,y=250)

font1 = ('times', 13, 'bold')

text=Text(main,height=20,width=130)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=300)

text.config(font=font1)

main.config(bg='plum2')

main.mainloop()
```

5. SCREENSHOTS

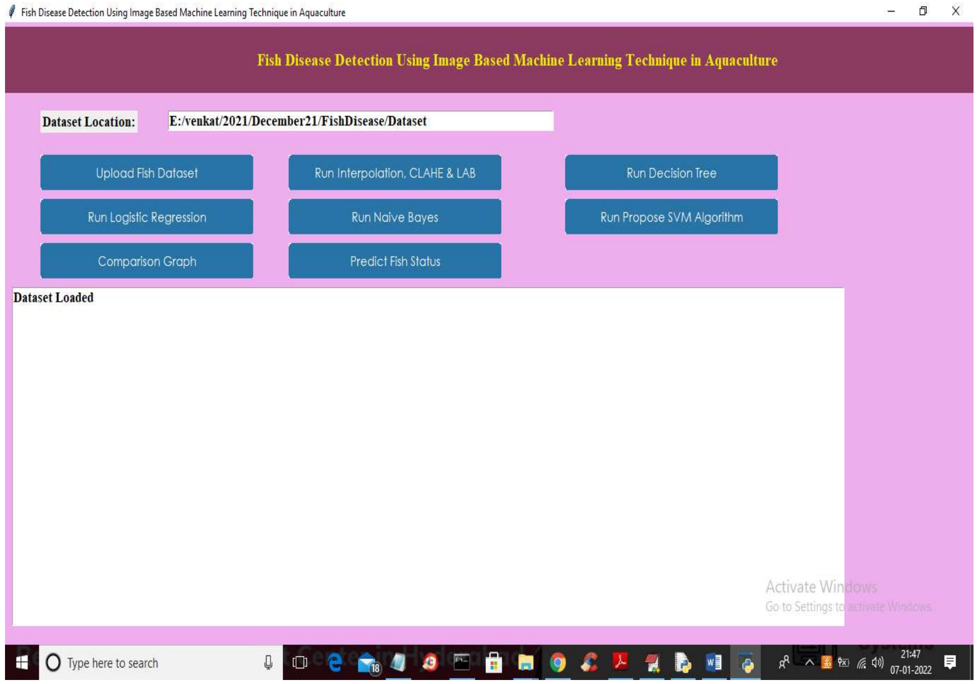
5. SCREENSHOTS



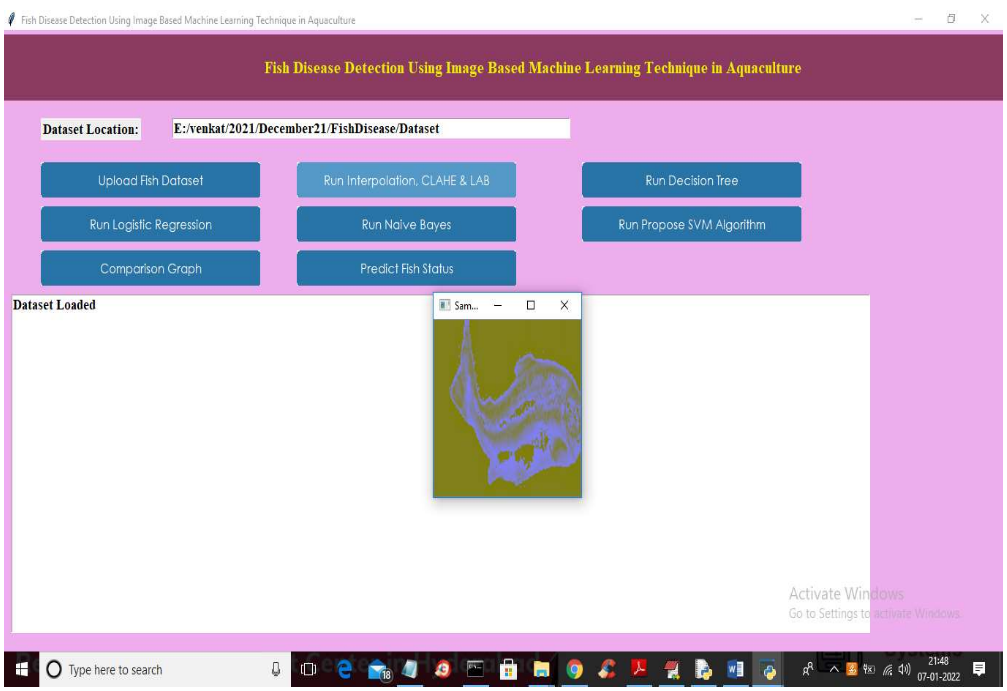
Screenshot 5.1: 'Upload Fish Dataset'



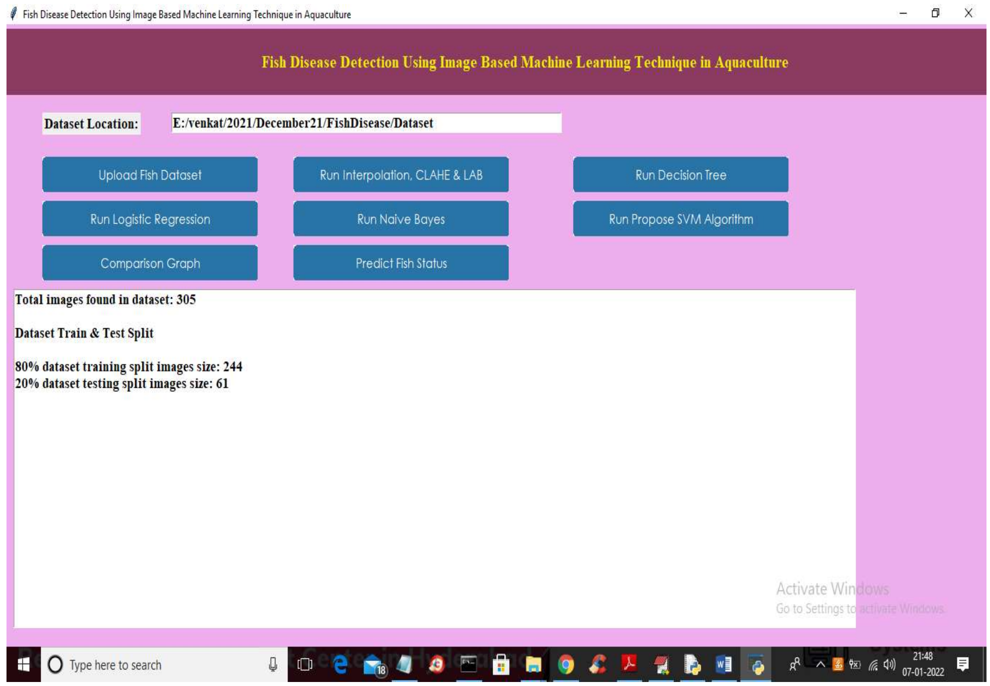
Screenshot 5.2: selecting folder



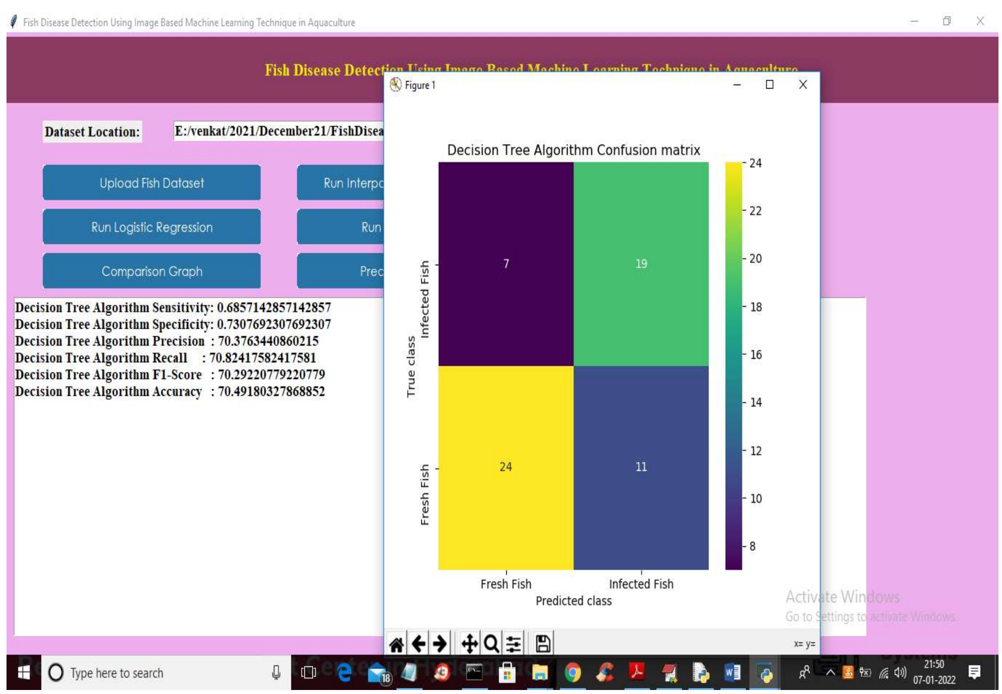
Screenshot 5.3: Run Interpolation



Screenshot 5.4: all images are processing

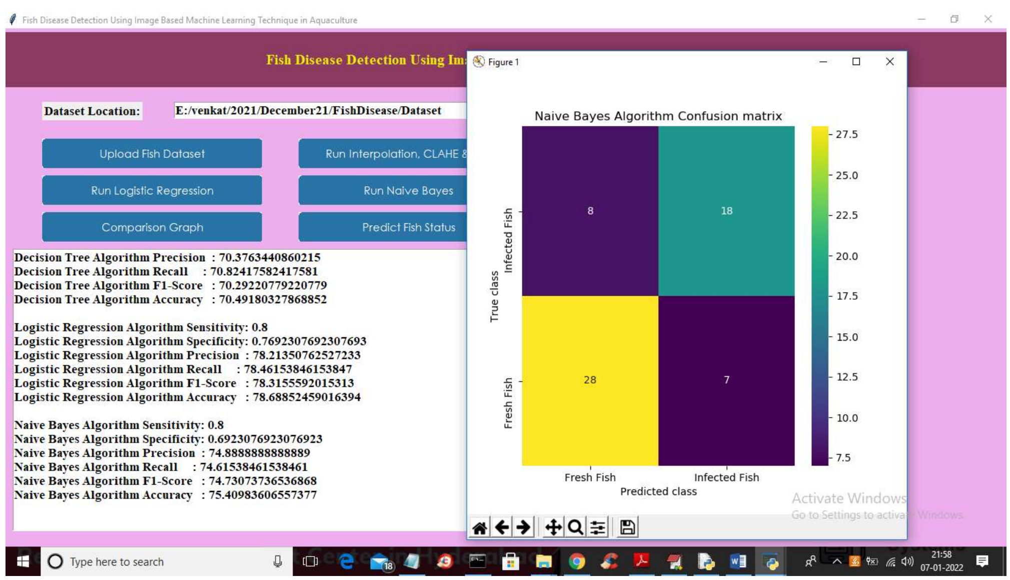


Screenshot 5.5: Data set uploaded

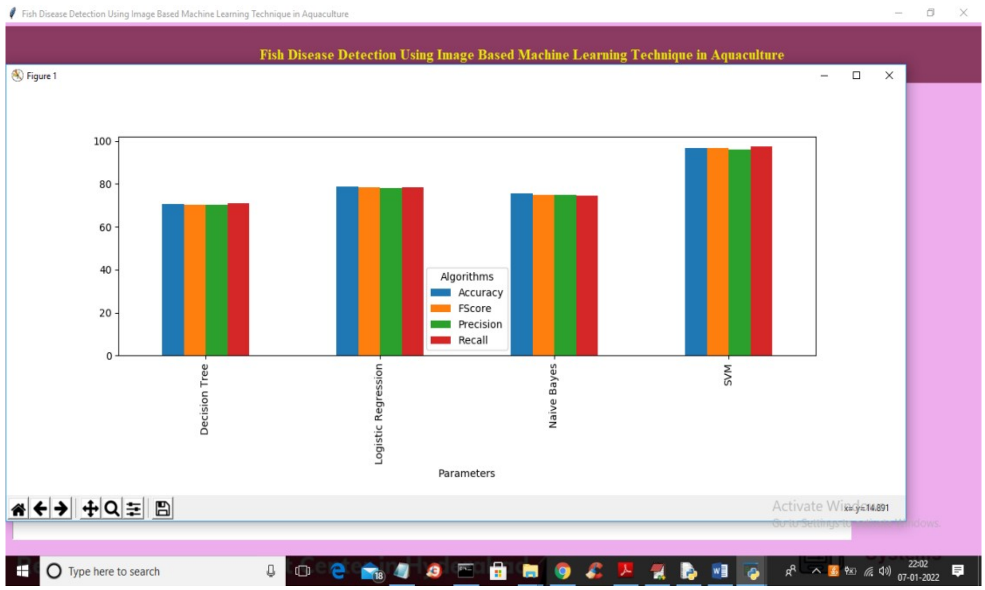


Screenshot 5.6: Run Decision Tree

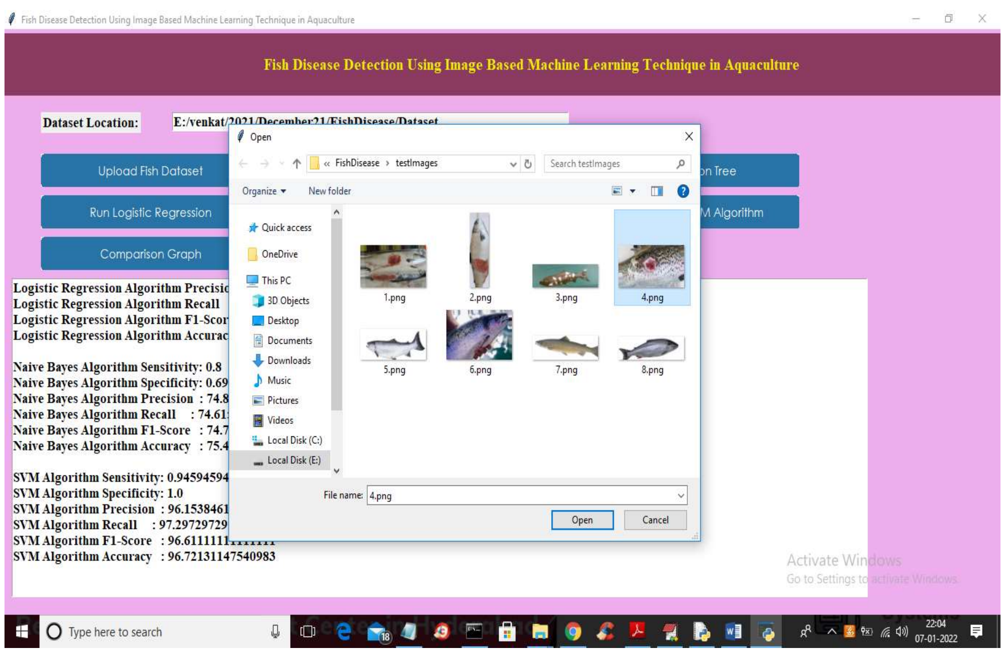
Screenshot 5.7: Run Logistic Regression



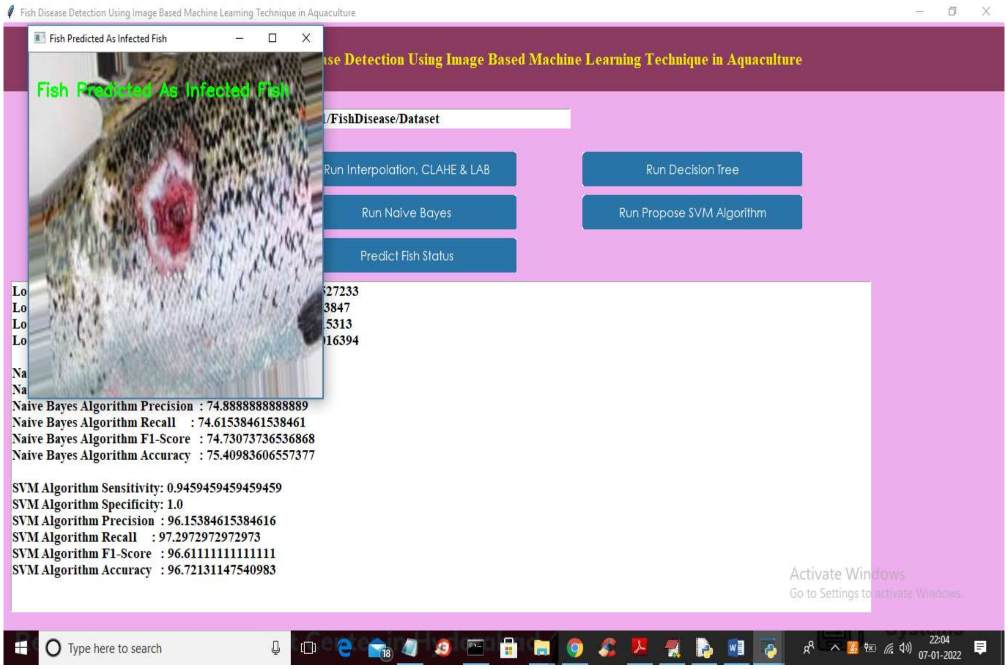
Screenshot 5.8: Run Propose SVM Algorithm



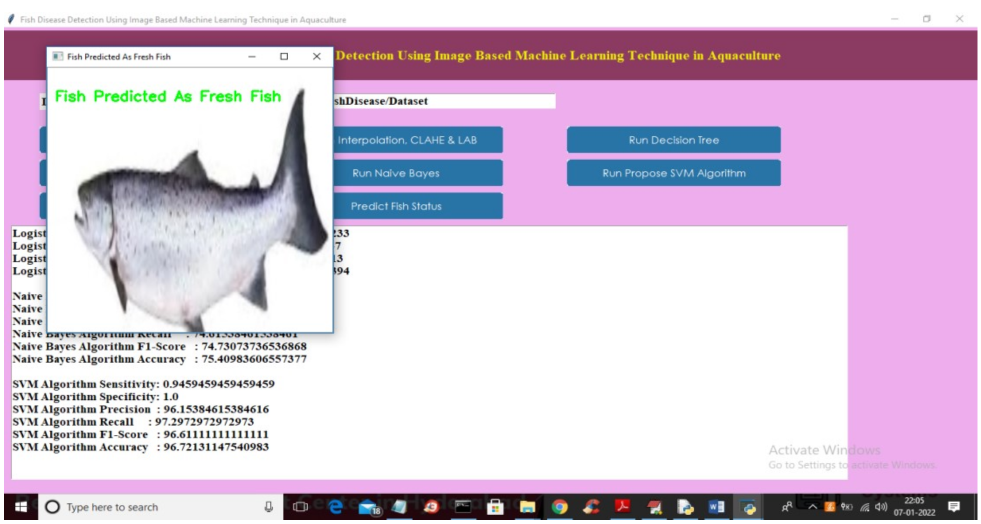
Screenshot 5.9: graph representation



Screenshot 5.10: Selecting & Uploading image



Screenshot 5.11: Infected fish



Screenshot 5.12: Fresh fish

6. TESTING

6. TESTING

6.1 INTRODUCING TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	:	identified classes of valid input must be accepted.
Invalid Input	:	identified classes of invalid input must be rejected.
Functions	:	identified functions must be exercised.
Output	:	identified classes of application outputs must be exercised.
System/Procedures	:	interfacing systems or procedures must be invoked.

ation and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; datafields,pre-defined processes, and successive processes must be considered for testing Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.2.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Test Results: All the test cases mentioned above passed successfully. No defects encountered

6.3 TEST CASES

6.3.1CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Fish disease detection	To detect infected fish	The user gives the input in the form of a image.	An output is Recognised as fresh fish
2	Fish disease detection	To detect infected fish	The user gives the input in the form of a image.	An output is Recognised as infected fish

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

In conclusion, the use of image-based machine learning techniques in aquaculture for fish disease detection has the potential to greatly improve the efficiency and accuracy of disease diagnosis, This is a promising area of research that can help to better understand and control the spread of disease in fish farms

7.2 FUTURE SCOPE

In the future, we utilize to various Convolutional Neural Networks (CNN) architecture for identifying fish disease. Moreover, we will focus on the implementation of a real-life IoT device using the proposed system. Doing so it can be a specific solution for the farmers in aquaculture to identify infected salmon fishes and take proper steps before facing any unexpected loss in their farming. We will work with different fish datasets to make our system more usable in other sectors of aquaculture. We will also concentrate on increasing our existing dataset as salmon fish is one of the demanding elements worldwide.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] T. Acharya. Median computation-based integrated color interpolation and color space conversion methodology from 8-bit bayer pattern rgb color space to 24-bit cie xyz color space, 2002. US Patent 6,366,692.

[2] S. Bianco, F. Gasparini, A. Russo, and R. Schettini. A new method for rgb to xyz transformation based on pattern search optimization. IEEE Transactions on Consumer Electronics, 53(3):1020–1028, 2007.

[3] A. Ben-Hur and J. Weston. A user's guide to support vector machines. In Data mining techniques for the life sciences, pages 223– 239. Springer, 2010.

[4] A. A. M. Abd Hamid, N. and A. Izani. Extended cubic b-spline interpolation method applied to linear two-point boundary value problem. World Academy of Science, 62, 2010.

8.2 GITHUB LINK

https://github.com/Saadvii/MINI-PROJECT-04