

Capstone Project for the Degree of Bachelor of Science (B.Sc.)

Real Time Dual-Mode Fish Disease Detection Based on Hybrid Deep Learning Framework

By

Tamanna Akter Mim

Student ID: 2014751050

Azharul Islam

Student ID: 2014751037

Fahmida Nasrin

Student ID: 2014751069

Sanzida Akter Chadne

Student ID: 2014751147

Department of CSE



University of Information Technology and Sciences (UITs)

Baridhara J Block, Dhaka-1212

[November 2023]

Real Time Dual-Mode Fish Disease Detection Based on Hybrid Deep Learning Framework

Prepared By

Tamanna Akter Mim	Student ID: 2014751050
Azharul Islam	Student ID: 2014751037
Fahmida Nasrin	Student ID: 2014751069
Sanzida Akter Chadne	Student ID: 2014751147

Supervised by

Ferdaus Anam Jibon

A capstone project submitted to the Department of Computer Science and Engineering of University of Information Technology & Sciences (UITS), Dhaka in Partial Fulfilment of the Requirements for the Degree of B.Sc. in Computer Science and Engineering

The Capstone Project entitled “Real Time Dual-Mode Fish Disease Detection Based on Hybrid Deep Learning Framework” submitted by Tamanna Akter Mim, Student ID: 2014751050; Azharul Islam, Student ID: 2014751037, Fahmida Nasrin , Student ID: 2014751069 & Sanzida Akter Chadne, Student ID: 2014751147 has been accepted as satisfactory in partial fulfilment of the requirements for the Degree of Bachelor of Science in Computer Science and Engineering on, 2023

Board of Examiners

- | | |
|--------------------------------------------------------------|------------|
| <hr/> | |
| (Al Imtiaz) | |
| 1. Department of Head | Chairman |
| Department of Computer Science and Engineering University of | |
| Information Technology & Sciences (UITs) | |
| Dhaka, Bangladesh | |
| <hr/> | |
| (Ferdaus Anam Jibon) | |
| 2. Assistant Professor | Supervisor |
| Department of Computer Science and Engineering | (Member) |
| University of Information Technology & Sciences | |
| (UITs) Dhaka, Bangladesh | |

DECLARATION

It is hereby declared that the work presented in this project or any part of this project has not been submitted elsewhere for the award of any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Signature

(Tamanna Akter Mim)

Department of CSE, UITS

Signature

(Azharul Islam)

Department of CSE, UITS

Signature

(Fahmida Nasrin)

Department of CSE, UITS

Signature

(Sanzida Akter Chadne)

Department of CSE, UITS

Signature of Project Supervisor

.....

(Ferdaus Anam Jibon)

Assistant Professor

Department of Computer Science and Engineering

University of Information Technology & Sciences (UITs)

Baridhara J Block, Dhaka-1212

Acknowledgement

In this acknowledgement, We express our genuine appreciation to all those who have contributed to and supported me in the creation of this Capstone Project. Firstly, we would like to thank our supervisor, Ferdaus Anam Jibon, Assistant Professor, Department of Computer Science and Engineering, University of Technology & Sciences, Dhaka, for his priceless help, support, and guidance. We would also like to express our gratitude to one of our contributor MD.Rubel Islam who helped us a lot by providing fish images that we used in this research.

We are truly grateful to our family for their understanding and endless love throughout our academic journey. Additionally, we want to thank our teachers, friends, and department staff members for sharing their knowledge and experiences, which have been invaluable in completing our project. Lastly, we offer our deepest appreciation to the Almighty Allah, without whose blessings this project would not have been possible.

Signature

(Tamanna Akter Mim)

Department of CSE, UITS

Signature

(Azharul Islam)

Department of CSE, UITS

Signature

(Fahmida Nasrin)

Department of CSE, UITS

Signature

(Sanzida Akter Chadne)

Department of CSE, UITS

Table of Contents

	Page
Acknowledgement.....	5
Table of Contents.....	6
List of Tables	9
List of Figures.....	10
Abstract.....	13
CHAPTER 1 Introduction	14
1.1 Introduction.....	17
1.2 Motivation.....	17
1.3 Objective.....	17
1.4 Contribution	17
1.5 Project Organization	17
CHAPTER 2 Literature Review.....	18
CHAPTER 3 Deep Learning for Fish Disease Detection.....	25
3.1 Introduction.....	25
3.2 Epizootic Ulcerative Syndrome (EUS) & Fin and Tail Rot detection using deep learning.....	25
3.3 Convolutional Neural Networks (CNNs).....	26
3.4 DenseNet169.....	26
3.5 VGG16.....	27
3.6 Xception.....	28
3.7 Ensemble With Voting.....	28
CHAPTER 4 Methodology.....	29
4.1 Data Preparation.....	29
4.2 Feature Extraction.....	29
4.3 Feature Selection.....	31
4.4 Proposed Method-I.....	32
4.4.1 Preprocessing.....	32

4.5 Proposed Method II.....	34
4.5.1 Preprocessing.....	34
4.6 Proposed Method III.....	35
4.6.1 Preprocessing.....	35
4.6.2 VGG16.....	35
4.6.3 Densnet169.....	37
4.6.4 Xception.....	39
CHAPTER 5 Dataset.....	41
CHAPTER 6 Experiment Analysis.....	43
6.1 Productive Setup.....	42
6.2 Experiment Analysis-I.....	44
6.2.1 Procedure	47
6.3 Experiment Analysis-II.....	47
6.3.1 Procedure of DensNet169.....	47
6.4 Experiment Analysis-III.....	49
6.4.1 Procedure of VGG16.....	49
6.5 Experiment Analysis-IV.....	51
6.5.1 Procedure of Xception.....	51
6.6 Ensemble of DenseNet169, VGG16 And Xception.....	53
6.7 Comparison of the Models Accuracy with Graph.....	54
6.7.1 Pre-trained Models.....	54
6.7.2 Basic CNN Model.....	54
6.8 Edge Detection.....	55
6.8.1 Canny Edge Detection.....	55
6.8.2 Prewitt Edge Detection.....	56
6.8.3 Robert Edge Detection.....	57
6.8.4 Sobel Edge Detection.....	57
6.8.5 Marr-Hildreth Edge Detection.....	58
CHAPTER 7 Result Analysis.....	60

7.1	Evaluation Metrics.....	60
7.2	Result Analysis for Experiment-I.....	61
7.2.1	Explanation.....	61
7.3	Result Analysis for Experiment-II.....	63
7.3.1	Explanation.....	63
7.4	Result Analysis for Experiment-III.....	65
7.4.1	Explanation.....	65
7.5	Result Analysis for Experiment-IV.....	66
7.5.1	Explanation.....	66
7.6	Result Analysis for Experiment-V.....	68
7.6.1	Explanation.....	68
7.7	Result Analysis for Experiment-VI.....	69
7.7.1	Explanation	69
7.8	Discussion.....	71
7.9	Limitations.....	73
CHAPTER 8	Future Work & Conclusion.....	74
8.1	Future Work.....	74
8.2	Conclusion.....	74
REFERENCES.....		75

List of Tables

	Page
Table 1: Third Dataset Information.....	39
Table 2: Dataset Information.....	40
Table 3: Number of images per classification type.....	40
Table 4: Hardware Setup.....	41
Table 5: Hyperparameter Setup.....	41
Table 6: Results for 10 different images using Basic CNN Model.....	44
Table 7: The Configuration of DenseNet.....	45
Table 8: Results for 10 different images using DenseNet169.....	46
Table 9: Results for 10 different images using VGG16	48
Table 10: Results for 10 different images using Xception.....	50
Table 11: CNN classification report.....	59
Table 12: CNN average matrices.....	60
Table 13: DenseNet169 classification report.....	61
Table 14: DenseNet169 average metrics.....	62
Table 15: VGG16 classification report.....	63
Table 16: VGG16 average matrices.....	64
Table 17: Xception classification report.....	65
Table 18: Xception average metrics.....	65
Table 19: Ensemble with voting average metrics.....	66
Table 20: Comparison of the Proposed Method with Other Related Work.....	69

List of Figures

	Page
Figure 1: Healthy Fish.....	12
Figure 2: Epizootic Ulcerative Syndrome (EUS).....	13
Figure 3: Fin and Tail Rot.....	14
Figure 4: Architecture of CNN.....	24
Figure 5: The architecture of Densnet169.....	25
Figure 6: The architecture of VGG16.....	25
Figure 7: The architecture of Xception.....	26
Figure 8: Feature extraction for fish disease detection.....	27
Figure 9: Histogram of Normal Image VS Contrast Enhanced Image.....	28
Figure 10: Extracted the feature for 10 Fish Image.....	29
Figure 11: Feature Selection Techniques of Image.....	29
Figure 12: Canny Edge detection.....	30
Figure 13: Prewitt Edge detection.....	30
Figure 14: Robert edge detection.....	31
Figure 15: Sobel edge detection.....	31
Figure 16: Marr-Hildreth edge detection.....	32
Figure 17: Classification with CNN.....	32
Figure 18: CNN Model.....	33
Figure 19: VGG-16 model.....	34
Figure 20:DenseNet-169 model.....	36
Figure 21: Xception Model.....	38
Figure 22: Sample images from our dataset.....	40
Figure 23: Basic CNN Architecture.....	43

Figure 24: The Configuration of Basic CNN Model.....	44
Figure 25: Basic CNN Result Prediction.....	44
Figure 26: Result Prediction of DenseNet169.....	46
Figure 27: The Configuration of VGG16.....	47
Figure 28: Detailed Information of the Proposed Model VGG16.....	48
Figure 29: Result prediction of VGG16.....	49
Figure 30: Detailed Information of the Proposed Model Xception.....	49
Figure 31: The Configuration of Xception.....	50
Figure 32: Result prediction of Xception.....	51
Figure 33: Model evaluation using graph.....	52
Figure 34: CNN Model.....	52
Figure 35: Process steps of canny edge detection.....	53
Figure 36: Canny Edge Detection.....	53
Figure 37: Process steps of Prewitt Edge Detection.....	54
Figure 38: Prewitt Edge Detection.....	54
Figure 39: Robert Edge Detection.....	55
Figure 40: Process steps of Sobel Edge Detection.....	56
Figure 41: Sobel Edge Detection.....	56
Figure 42: Process steps of Marr-Hildreth Edge Detection.....	57
Figure 43: Marr-Hildreth Edge Detection.....	57
Figure 44: CNN Confusion Matrix.....	59
Figure 45: Result for Basic CNN Model.....	60
Figure 46: DenseNet169 Confusion Matrix.....	61
Figure 47: Result for DenseNet169 Model.....	62
Figure 48: VGG16 Confusion Matrix.....	63
Figure 49: Result for VGG16 Model.....	64

Figure 50: Xception Confusion Matrix	65
Figure 51: Result for Xception Model.....	66
Figure 52: Matrix Representation.....	68

Abstract

This study proposes an innovative approach to fish disease detection through the utilization of hybrid deep learning models, emphasizing both accuracy and interpretability. The first model, based on Convolutional Neural Networks (CNN), achieves an outstanding accuracy rate of 98.99%. It successfully differentiates between healthy fish, those with EUS illness, and those exhibiting fin and tail rot. This high precision in classification demonstrates the effectiveness of the proposed model in capturing nuanced health states among fish populations. In addition to the primary CNN-based model, a second hybrid deep learning model is introduced. This model combines the strengths of Xception (86% accuracy), Vgg16 (60%), and Densnet169 (96.5%) to discern between diseased and healthy fish. This ensemble approach leverages the unique features of each model, contributing to a more comprehensive and adaptable diagnostic system for fish health. The integration of diverse edge detection techniques, including Canny, Prewitt, Robert, Sobel, and Marr-Hildreth, further enhances the research. These techniques provide valuable insights into the edges and structural details of fish images, offering researchers a nuanced understanding of disease-related features. This aspect contributes to the interpretability of the models, making the diagnostic process not only accurate but also transparent and informative. The comprehensive nature of this study is underlined by the thoughtful combination of advanced deep learning architectures, ensemble strategies, and edge detection techniques. This multifaceted approach not only attains high accuracy rates but also enriches the interpretability and explainability of the models, crucial for practical implementation in real-world aquatic environments. Furthermore, the proposed models and techniques hold promise for practical applications in aquaculture and fishery management. The ability to accurately identify and differentiate between various health states in fish populations can significantly contribute to early disease detection and timely interventions, promoting the overall health and sustainability of aquatic ecosystems. In conclusion, this research provides a robust framework for fish disease detection, demonstrating the efficacy of hybrid deep learning models, ensemble techniques, and edge detection strategies. The achieved high accuracy rates, coupled with enhanced interpretability, position this study at the forefront of advancements in automated aquatic health assessment, with implications for improved disease management and ecological conservation.

CHAPTER 1

Introduction

1.1 Introduction

Fish is a vital source of protein for billions of people worldwide, with over 3 billion relying on it as a crucial part of their diet. It's especially important for coastal communities and developing countries. The global fish industry is a significant contributor to the economy, providing jobs for millions in fishing, processing, and selling fish. The world economy is deeply intertwined with the fish industry, valued at over \$150 billion annually. However, challenges like overfishing and illegal practices can harm fish populations and impact the economy. Fish diseases also have a big effect. They can spread quickly in crowded fish farms, leading to losses for fish farmers. In some cases, diseases can even affect wild fish populations. This highlights the importance of proper management and health measures in aquaculture. Overall, fish is not just a tasty meal, but a cornerstone of food security, livelihoods, and the world economy. It's crucial to protect this valuable resource for current and future generations.



Figure 1: Healthy Fish

Fish is a crucial source of protein in Bangladesh, providing nutrition for millions of people. Around 60% of animal protein in the average Bangladeshi diet comes from fish. This makes it an essential part of their meals and diets. Moreover, nearly 1.5 million people are directly employed in the fishing industry, supporting their livelihoods. The economy of Bangladesh greatly benefits

from the fish sector. It contributes to about 4% of the country's total GDP. However, fish diseases can have a harmful impact. Outbreaks of diseases in fish farms can lead to significant economic losses, affecting both farmers and the overall industry. This emphasizes the need for measures to prevent and control such diseases, ensuring a stable and thriving fish sector in Bangladesh.

Fish diseases in Bangladesh have significant repercussions on the aquaculture industry, which is a vital component of the country's economy. Outbreaks of diseases can lead to reduced fish yields, lower quality of produce, and even mass mortality in fish farms. This results in substantial economic losses for fish farmers and the industry as a whole.



Figure 2: Epizootic Ulcerative Syndrome (EUS)

Various types of diseases afflict Bangladeshi fish, including bacterial, viral, and parasitic infections. For instance, diseases like Aeromoniasis, Motile Aeromonad Septicemia (MAS), and White Spot Syndrome Virus (WSSV) have been known to affect fish populations. To mitigate these challenges, efforts are underway to improve disease prevention and control measures, including better farm management practices, improved water quality, and the use of vaccines and antibiotics where necessary. This is crucial to sustain the growth and stability of the fish sector in Bangladesh.

Fish disease detection using images is like using a special kind of "eye" to see if a fish is sick. This special eye is actually a computer program that looks at pictures of fish and checks for signs of illness. This process is called "image processing." "Image processing works by using computer algorithms to analyse pictures. These algorithms are like sets of instructions that tell the computer how to look for specific things in the images. For example, they can be trained to recognize changes in the colour, shape, or texture of a fish, which might indicate a disease. Scientists and researchers take pictures of fish and feed them into the computer program. The program then follows the instructions in the algorithms to study the pictures. If it finds any signs

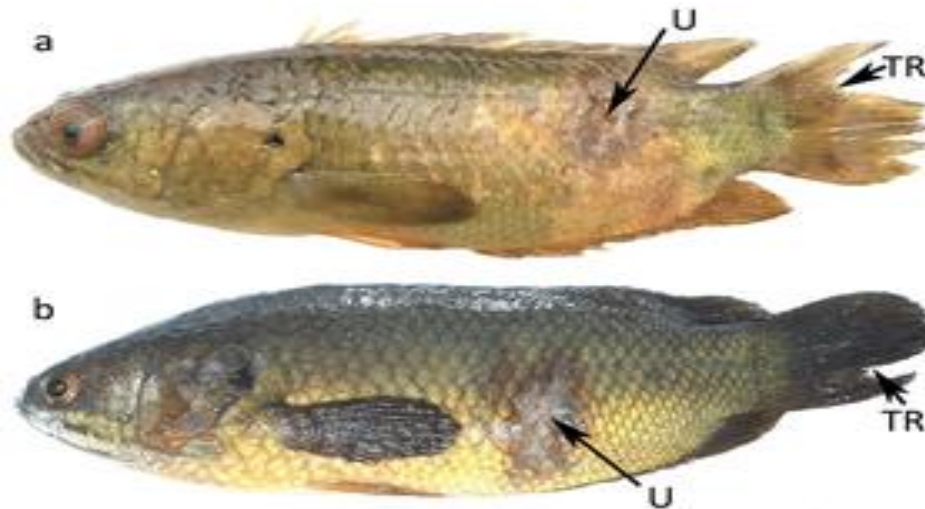


Figure 3: Fin and Tail Rot

of sickness, it will alert the experts. This method helps in quickly and accurately identifying fish diseases, which is important for taking care of the fish and preventing the spread of illnesses in farms and natural habitats. It's like having a smart helper that can spot when a fish isn't feeling well just by looking at a picture!

Fish disease detection using deep learning is a powerful technology that uses advanced computer algorithms to identify illnesses in fish. Deep learning is like giving the computer a super smart brain that can learn and recognize patterns from lots of data. The impact of deep learning on fish disease detection is significant. It allows for more accurate and efficient identification of diseases compared to traditional methods. Deep learning models are trained on vast amounts of fish images, learning to recognize subtle signs of illness that might be hard for humans to notice. Because of this, deep learning can quickly analyse large batches of fish images, making it a valuable tool for aquaculture farms and researchers. It helps in detecting diseases early, which is crucial for keeping fish healthy and preventing outbreaks that can harm fish populations and the industry. Overall, deep learning revolutionizes fish disease detection, making it faster, more precise, and ultimately contributing to the health and sustainability of fish populations. It's like having a highly trained expert that can examine lots of fish pictures in no time.

Fish disease detection using deep learning offers several key benefits. Firstly, it allows for early and accurate identification of illnesses, enabling prompt treatment and prevention of outbreaks. This helps to maintain healthy fish populations, which is crucial for the aquaculture industry. Additionally, deep learning can process large volumes of images quickly, saving time and labor costs. As a result, the efficiency and productivity of fish farms are enhanced, contributing to economic growth in the aquaculture sector. Moreover, the technology promotes sustainability by minimizing the use of antibiotics and chemicals, which can be costly and have environmental impacts. Overall, the adoption of deep learning in fish disease detection leads to improved fish health, increased productivity, and a more sustainable industry.

1.2 Motivation

The motivation behind working on fish disease detection is to ensure the health and well-being of fish populations. Fish are an important source of food and livelihoods for many people around the world. Detecting diseases early can help prevent outbreaks and ensure that the fish stay healthy. This not only benefits the fish themselves, but also supports the livelihoods of those who rely on fishing and aquaculture. Additionally, healthy fish populations contribute to a sustainable and thriving ecosystem. By developing effective methods for fish disease detection, we aim to protect this valuable resource and the communities that depend on it.

1.3 Objective:

This thesis aims to create a smart system for identifying diseases in fish. The dual-mode approach ensures effectiveness in different situations. The goal is to provide a reliable and intelligent tool for quickly identifying and managing diseases in fish using a combination of powerful deep learning methods, CNN (Convolutional Neural Network) and an ensemble of DenseNet169, VGG16, and Xception deep learning models. The thesis's particular aims are to design and construct the model architecture, pick appropriate hyperparameters, evaluate the model's performance on benchmark datasets, compare it to current approaches, and analyze the results and potential future directions for the study

1.4 Contribution

- i. Proposed the use of CNN in a hybrid deep learning model to distinguish between fish that are healthy, fish with EUS illness, and fish with fin and tail rot.
- ii. Designed a second hybrid deep learning model to distinguish between disease- and healthy-fish by combining Xception, Vgg16, and Densnet169.
- iii. Using edge detection techniques like Canny Edge Detection, Prewitt Edge Detection, Robert Edge Detection, Sobel Edge Detection, and Marr-Hildreth Edge Detection to find the edges and give researchers fresh insights.

1.5 Project Organization

In this section, we will provide an overview of the chapters. Chapter 2 explores the related work, offering insights into existing research and frameworks relevant to fish disease detection. Moving forward, Chapter 3 delves into the application of deep learning specifically for fish disease identification. Chapter 4 provides a concise discussion of the methodology employed in the study. The dataset used for experimentation takes center stage in Chapter 5, detailing its composition and significance. Chapter 6 delves into the analysis of experiments conducted, shedding light on the outcomes and insights gained from implementing the proposed models. Following this, Chapter 7 presents a succinct description of the results analysis, summarizing the key findings and implications of the study.

Chapter 2

Literature Review

Fish disease detection can be achieved through various methods, including visual inspection, molecular techniques (e.g., PCR), and advanced imaging technologies. Visual inspection involves identifying external symptoms like lesions or discoloration. Molecular techniques analyse fish tissues for specific pathogens. Advanced imaging, such as computer vision and machine learning, utilizes cameras to capture and analyses fish behavior, body features, or histopathological images, enabling automated and rapid detection of diseases. We have used machine learning techniques. Here we used binary class classification to detect if the fishes are infected by EUS and Fin & Tail rot or not.

Authors of [1] present a valuable system for automatic fish disease detection, vital for fisheries management. They used image processing techniques, including K-means and C-means fuzzy logic clustering, along with feature extraction and Multi-SVM classification. But the paper lacks dataset details, a comprehensive evaluation, and a thorough comparison with existing methods. To improve the paper, provide dataset specifics, include additional evaluation metrics, offer technical implementation details, and explore the potential of deep learning methods for fish disease detection.

Authors of [2] introduce a machine learning-based SVM model to identify infected fishes in aquaculture, particularly salmon. It offers a valuable solution for early disease detection in fish farming, crucial for preventing the spread of diseases. The study combines image processing techniques and machine learning, including SVM, to identify fish diseases. It uses a dataset with and without image augmentation and achieves high accuracy rates of 91.42% and 94.12% for infected fish detection, respectively. But potential limitations might include the need for further validation with larger and more diverse datasets, and potential challenges in real-world implementation. The authors could improve the research by conducting validation on a broader range of fish species, expanding the dataset, and addressing potential real-world implementation challenges. Additionally, they should consider exploring the generalizability of their model to different aquaculture settings.

Authors of [3] address a critical issue in the fishing industry by proposing a Convolutional Neural Network (CNN) for automated fish disease detection, offering a fast and non-invasive solution to improve accuracy. The study employed a CNN with a three-layer process (convolutional, pooling, and fully connected) and tested it on a dataset of 90 fish disease images, achieving an overall mean accuracy of 94.44%. The paper lacks a thorough discussion of the limitations and challenges faced during implementation, and there is room for improvement in terms of dataset size and diversity. Future work should include a more extensive dataset, detailed evaluation metrics, and a discussion on the generalizability of the proposed CNN model to different fish species and diseases for enhanced practicality in real-world scenarios.

Authors of [4] Addresses the crucial need for precise and timely fish disease diagnosis in aquaculture, emphasizing the integration of image-processing and computer vision technologies. It highlights the potential ecological and economic benefits of accurate disease prevention, making it a valuable contribution to the field. The study comprehensively reviews image-processing technology and image-based fish disease detection methods, analyzing their benefits and drawbacks. It emphasizes the application of computer vision for real-time and automatic diagnosis, providing insights into the challenges faced in the current methods. The paper acknowledges the generally low accuracy of disease diagnosis due to the diversity of fish species and challenges in obtaining high-definition images of diseased fish. However, it lacks specific details on quantitative metrics and specific limitations encountered in the reviewed methods. Future research should focus on improving image quality, constructing 3D models of fish diseases, establishing standard and shared datasets, and enhancing deep learning methods for more accurate and reliable fish disease diagnosis. Additionally, addressing the challenges of image acquisition in complex water environments could further improve the practicality of the proposed approaches.

Authors of [5] present a practical methodology for detecting and diagnosing fish diseases through colour image segmentation, offering an automated and time-efficient approach. It provides a valuable contribution to the field by addressing the need for effective and quick methods to prevent the spread of infectious fish diseases. The study employs color image segmentation and markers to detect infected areas on fish bodies, demonstrating the feasibility of digital image processing for studying fish diseases. The approach is illustrated using public images of fish diseases, showcasing its application in practical scenarios. But a potential drawback is the appearance of false points marked as infected areas due to automatic colour allocation. This could affect the accuracy of the calculations and the reliability of disease detection. To enhance the methodology, the authors could explore methods to reduce false positives in colour segmentation. Additionally, validating the approach with a larger dataset and quantitative metrics would strengthen the reliability and applicability of the proposed method.

The authors of [6] address the challenging task of fish disease diagnosis, focusing on Epizootic Ulcerative Syndrome (EUS). It combines image processing techniques, feature extraction, and machine learning algorithms to achieve accurate and automated identification of EUS-infected fish, offering a valuable contribution to early disease detection. The paper employs segmentation, edge detection, and morphological operations in the first part to enhance EUS-infected fish images. In the second part, it utilizes feature extraction techniques like HOG and FAST, coupled with PCA for dimensionality reduction, and employs machine learning algorithms, such as Neural Network and K-NN, for classification. The paper mentions the challenge of manual diagnosis, but it does not explicitly address potential limitations or drawbacks of the proposed methodology, such as sensitivity to variations in image quality or conditions. The authors could enhance the paper by providing a more detailed discussion of the limitations or challenges encountered during experimentation. Additionally, exploring methods

to improve robustness against variations in image quality and conditions would strengthen the proposed methodology.

The authors of [7] address the critical issue of early detection of fish diseases by proposing a machine learning-based solution that predicts water quality degradation, a key factor influencing fish health. The approach aims to empower fish farmers with timely information for preemptive actions, contributing to economic stability in the aquaculture sector. The authors utilize a state-of-the-art machine learning algorithm, Gradient Boosting, to analyse water quality parameters and predict potential fish diseases. They emphasize the importance of a hierarchical architecture and decision-tree-based regression to enhance the accuracy of classification and decision-making. The paper lacks explicit discussion on potential limitations or challenges associated with the proposed model. Addressing factors such as data variability or the need for real-time data acquisition could strengthen the paper. The authors could enhance the paper by providing a more detailed discussion of the dataset's representativeness and potential biases. Additionally, considering the integration of IoT-based implementations to enable on-site usage would be a valuable improvement for practical application.

The authors of [8] offer fish producers an automated solution to the pressing problem of early fish disease identification using computer vision techniques. The suggested method, which makes use of CNN architectures, shows excellent accuracy in classifying three different fish diseases, offering aquaculture professionals a useful tool for disease control. The authors used CNN designs like Alexnet, ResNet18, ResNet50, and ResNet101 as well as other computer vision approaches. They used a dataset made up of fish photos with three different sorts of diseases to conduct studies in three distinct colour spaces: RGB, YCBCR, and XYZ. The study's main objective was to automatically identify and categories fish diseases using anomalous behaviors and visual cues. The difficulties or restrictions of the suggested system are not thoroughly discussed in the paper. Furthermore, practical problems are left unanswered as the hardware circuit and notification system's real-world implementation is still planned for the future. By outlining potential solutions and offering insights into the difficulties encountered during the trials, the authors can improve the paper. A more thorough explanation of the hardware implementation, together with experimental settings and outcomes, would increase the suggested system's practical usefulness.

The authors of [9] a valuable approach to automatically identify and segment diseased areas in fish images using PCA, K-Means clustering, HSV, and morphological operations. The proposed method can significantly contribute to the early detection of fish diseases, aiding in preventive measures. The authors employed PCA for feature extraction, K-Means clustering for colour segmentation, and HSV with morphological operations for precise diseased area detection. The combination of these techniques provides a comprehensive solution for automated fish disease identification. While the paper demonstrates promising results, the drawback lies in the limited scope of evaluation with only four case study images of Epizootic Ulcerative Syndrome (EUS) diseased fish. Further validation on a larger dataset with diverse fish diseases is essential for

generalizability. To enhance the paper, future work could focus on expanding the dataset to include various fish diseases, evaluating the method's robustness. Additionally, comparing the proposed approach with other state-of-the-art methods would strengthen the validation and provide insights into its competitiveness in the field.

The authors of [10] proposed a system that provides an automated method for the detection and classification of fish diseases, addressing a critical issue in the fisheries sector. It employs image processing techniques, including K-means and C-means clustering, Gabor filters, GLCM, and Multi-SVM, achieving high accuracy rates. The authors collected data from the Rangamati Kaptai Lake and Sunamganj Hoar area, implemented preprocessing, segmentation, feature extraction, and classification stages using various techniques. The experimental results demonstrate the effectiveness of the proposed approach. The study may lack a comprehensive discussion on the limitations and challenges of the proposed system, potentially hindering a deeper understanding of its applicability in diverse settings. Future work could focus on addressing the system's limitations, exploring real-world implementation challenges, and considering advancements in deep learning for enhanced accuracy and adaptability to evolving fish disease scenarios. Additionally, the paper could benefit from a more extensive literature review and comparison with existing methodologies.

The authors of [11] address the critical issue of detecting salmon fish diseases in aquaculture using a combination of image processing and machine learning, providing a valuable contribution to fish health management. The authors employ image pre-processing, segmentation, and SVM-based classification on a novel dataset to achieve disease detection. They also compare results with and without image augmentation, adding depth to their analysis. The paper lacks specific details on the image augmentation techniques used, and the comparison with other models could be more comprehensive. The authors could enhance the paper by providing a more detailed explanation of the image augmentation methods employed and conducting a more extensive comparison with a variety of existing models to strengthen the paper's validity.

The authors of [12] address the crucial issue of fish disease recognition using a computer vision-based expert system, offering a promising solution for early detection and prevention in aquaculture, thereby contributing to food security. The authors employ image processing, including the K-means clustering algorithm for feature extraction, and evaluate eight classifiers, with Random Forest achieving an accuracy of 88.87%, showcasing a comprehensive approach to fish disease recognition. The paper lacks detailed information on the size and diversity of the dataset used, and there is a need for more extensive testing on a broader range of fish diseases to enhance generalizability. The authors could improve the paper by providing more transparency on the dataset size and diversity. Additionally, expanding the study to include a more comprehensive set of fish diseases would strengthen the validity and applicability of the proposed system.

The authors of [13] introduce a novel approach by combining Augmented Reality (AR) and Image Processing for early detection of fish diseases, specifically focusing on Epizootic Ulcerative Syndrome (EUS). This innovative method provides real-time, on-the-spot identification, contributing to improved fish health and production. The authors create a 3D model of diseased fish using Blender, develop an AR app with Unity and Vuforia, and implement HSV-based image processing for disease detection. The combination of AR visualization and image processing enhances the accuracy of identifying diseased areas. The study is limited to a specific disease (EUS) and relies on mobile devices with certain hardware capabilities. The paper lacks a comprehensive evaluation of the proposed method across various fish diseases and devices. Future work could expand the research to encompass a broader range of fish diseases, ensuring the applicability of the AR and Image Processing technique. Additionally, addressing limitations related to hardware dependency would enhance the method's accessibility and usability across different devices.

The authors of [14] present an innovative approach to detect Epizootic Ulcerative Syndrome (EUS) in fish using FAST and HOG feature descriptors, aiming at accurate and rapid diagnosis. The application of PCA for dimensionality reduction enhances the classification accuracy, demonstrating the potential for improved disease identification in the aquaculture industry. The authors conducted experiments on real images of EUS-infected fish in MATLAB, employing Neural Network algorithms for classification. The combination of FAST and HOG feature descriptors, along with PCA, contributed to better accuracy in identifying infected fish. The study is limited to a specific disease (EUS) and lacks a comprehensive comparison with other existing methods. Additionally, the evaluation could benefit from a larger dataset and consideration of diverse fish diseases. Future work could involve expanding the research to cover a broader spectrum of fish diseases, ensuring the generalizability of the proposed method. Including a comparative analysis with other state-of-the-art techniques and addressing limitations related to dataset size would enhance the paper's overall contribution to the field.

The authors of [15] demonstrate the application of deep learning, specifically the YOLO v3 algorithm, for real-time monitoring of fish behavior in aquaculture. The proposed system successfully distinguishes normal and abnormal swimming patterns in rock bream, offering potential for automated bio-management in the aquaculture industry. The researchers utilised a video recorder to capture swimming patterns of rock bream under normal and abnormal conditions. The YOLO v3 algorithm, based on the Darknet-53 convolutional neural network, was employed for fish identification and behavior analysis, achieving a high accuracy of 98.1%. The study's limitation lies in the focus on a specific fish species (rock bream) and the induced abnormal conditions, potentially limiting the generalizability of the algorithm to diverse aquaculture scenarios and fish species. Future research could involve testing the algorithm across different fish species and environmental conditions, ensuring robustness and broader applicability. Additionally, incorporating a larger dataset and considering additional behavioral parameters could enhance the algorithm's accuracy and reliability.

The authors of [16] address a critical issue in aquaculture by introducing a fish disease detection and identification system based on the YOLOv4 model, enhancing real-time monitoring of skin health in deep-sea cage culture. The MobileNet3-GELU-YOLOv4 model demonstrates improved learning ability, reduced parameters, and increased detection speed, offering practical applications in disease prevention. The researchers employed an improved YOLOv4 model, integrating lightweight depthwise separable convolution and optimized feature extraction for efficient detection of four common fish diseases. The model's application in deep-sea cage culture showcases its utility in real-time monitoring. The study acknowledges limited availability of high-quality underwater images for training, potentially affecting the model's generalisation. This limitation underscores the need for comprehensive datasets to enhance the model's performance. Future research should focus on constructing larger, higher-quality datasets for improved generalization and incorporate underwater image processing algorithms to enhance real-time disease recognition. Continuous efforts in AI-based fish disease detection may revolutionize aquaculture monitoring practices.

The authors of [17] address the crucial need for automated fish disease recognition in aquaculture to enhance disease control and optimize production. The proposed hybrid approach, integrating multilayer fusion, attention mechanism, and OSELM, achieves high accuracy and demonstrates potential for real-world applications. The authors developed a CNN-OSELM multilayer fusion network with attention mechanism, incorporating same level composition and background elimination to improve feature extraction in recognizing fish diseases. The model was compared with baseline, attention-based, ConvNeXt, and Swin transformer models, showcasing its efficacy. The study acknowledges limited availability of public fish disease datasets and challenges posed by low-quality underwater images. These limitations could affect model generalisation and performance. Future work should focus on expanding the dataset for improved generalisation and comparing the proposed model with fuzzy logic applied to segmented images. Addressing these aspects can enhance the model's robustness and applicability in smart aquaculture systems.

The authors of [18] introduce a novel method for real-time detection of anomalous behaviour in underwater fish, leveraging deep learning, object detection, directed cycle graph, fish tracking, and dynamic time warping. This contributes to early detection of issues in aquaculture, reducing losses, enabling precision breeding, and supporting ecological conservation. The authors utilize object detection and a directed cycle graph to encode fish postures, implement a self-recovery tracking algorithm, and calculate polar descriptors for dynamic time warping. The method shows robustness against untidy backgrounds and imperfect instance segmentation, offering stable performance. The method's performance depends on the accuracy of object detection, and occasional detection failures can occur. Additionally, the study acknowledges the need for further exploration of the relevance between fish behaviour and environmental factors. Future work could involve enhancing object detection accuracy, exploring more behaviour templates, and considering additional environmental parameters like temperature and salinity. Deployment

of edge computing in sea cages with compressed or lite versions of object detection models is suggested for practical implementation.

The authors of [19] The paper introduces DFYOLO, an improved YOLOV5-based model (DFYOLO) for real-time detection of diseased fish in intensive aquaculture. DFYOLO demonstrates enhanced accuracy (99.38% mAP50) compared to YOLOV5m, offering a valuable tool for timely disease identification and prevention in aquaculture. DFYOLO addresses challenges of low-quality underwater images by replacing the CSPNet structure, employing a convolution kernel group, and incorporating a convolutional block attention module (CBAM). These modifications result in improved detection accuracy and efficient inference speed, contributing to the applicability of the model in intensive aquaculture. The study acknowledges limitations in not supporting multi-target dynamic tracking, hindering comprehensive health assessment. The dataset is limited, and the method may benefit from further exploration of geometric features for a more accurate health estimation model. Future work could focus on enhancing the algorithm to support multi-target dynamic tracking, expanding the dataset to include more fish disease behaviours, and exploring additional features such as geometric parameters for a more comprehensive health estimation model. Integration of stereo and multispectral cameras may further improve performance in challenging environmental conditions.

The authors of [20] provide a comprehensive review of image processing techniques in fish disease detection, emphasising the importance of automation for rapid and reliable identification. It highlights the potential of image-based models to enhance accuracy and early detection in aquaculture. The authors systematically review various image processing steps, including acquisition, pre-processing, segmentation, and classification, offering insights into the diverse approaches used in fish disease detection. They discuss applications of statistical methods, rule-based expert systems, machine learning, deep learning, and hybrid methods. The review lacks specific quantitative comparisons or evaluations of the discussed methods, making it challenging to gauge the performance differences among the various image processing techniques. Additionally, there's a limited exploration of recent advancements or emerging technologies in the field. Future work could include more detailed performance evaluations and comparative analyses of image processing methods for fish disease detection. Incorporating recent advancements, emerging technologies, and addressing the limitations of existing approaches would enhance the paper's value for researchers and practitioners in the field.

Chapter 3

Deep Learning for Fish Disease Detection

3.1 Introduction:

With the evolution of machine learning, especially the emergence of deep learning, and the widespread use of graphics processing units in the 21st century, the application of artificial intelligence (AI) technology in big data analysis is becoming increasingly prevalent across various fields, including fisheries. In the realm of fish health, the utilization of AI technology, particularly in the analysis of data from methods like fish pictures, promises an objective and thorough exploration of complex biological processes. By employing AI-powered analysis, researchers can gain a holistic and unbiased understanding of factors influencing fish health, thus advancing the field significantly.

Deep learning, a pivotal development in machine learning since 2006, operates by leveraging multi-layer non-linear processing for feature extraction, transformation, pattern analysis, and classification. In the context of fish disease detection, deep learning models can unravel intricate associations between different data points, providing a powerful tool for comprehending and addressing challenges in aquatic health.

3.2 Epizootic Ulcerative Syndrome (EUS) & Fin and Tail Rot detection using deep learning

Deep learning algorithms play a crucial role in automated systems for detecting Epizootic Ulcerative Syndrome (EUS) and Fin and Tail Rot in fish, overcoming the limitations of traditional machine learning approaches. As illustrated in Figure 9, deep learning exhibits the capability to manage extensive and imbalanced datasets without bias. Unlike manual feature extraction methods, deep learning automates this process, enhancing the identification of diseases in aquatic organisms. The application of deep learning, particularly through the use of Python and open-source toolboxes, facilitated by cloud computing, has significantly advanced the development of unique automated systems. TensorFlow and Keras are commonly employed in the context of EUS and Fin and Tail Rot detection in fish, showcasing their adaptability in handling diverse datasets have been extensively researched for the identification of these aquatic diseases. The field continues to witness growth with the introduction of novel and effective deep learning models for improved detection and understanding of Epizootic Ulcerative Syndrome and Fin and Tail Rot.

3.3 Convolutional Neural Networks (CNNs):

The Convolutional Neural Network (CNN) architecture for fish disease detection involves several key components. First, the input layer receives images of fish. Convolutional layers apply filters to identify patterns like textures and colours. These filters create feature maps highlighting important image characteristics. Pooling layers then reduce the dimensionality of the data, preserving essential information. Multiple convolutional and pooling layers work together, forming a hierarchy of features. Flattening converts this hierarchical information into a linear structure. Fully connected layers further analyse and classify the features, determining if the fish is healthy or diseased. Activation functions introduce non-linearities to the model. During training, the model adjusts its parameters through backpropagation to improve accuracy. The output layer produces a probability score, indicating the likelihood of disease presence. This CNN architecture enables the automated identification of fish diseases based on learned visual patterns.

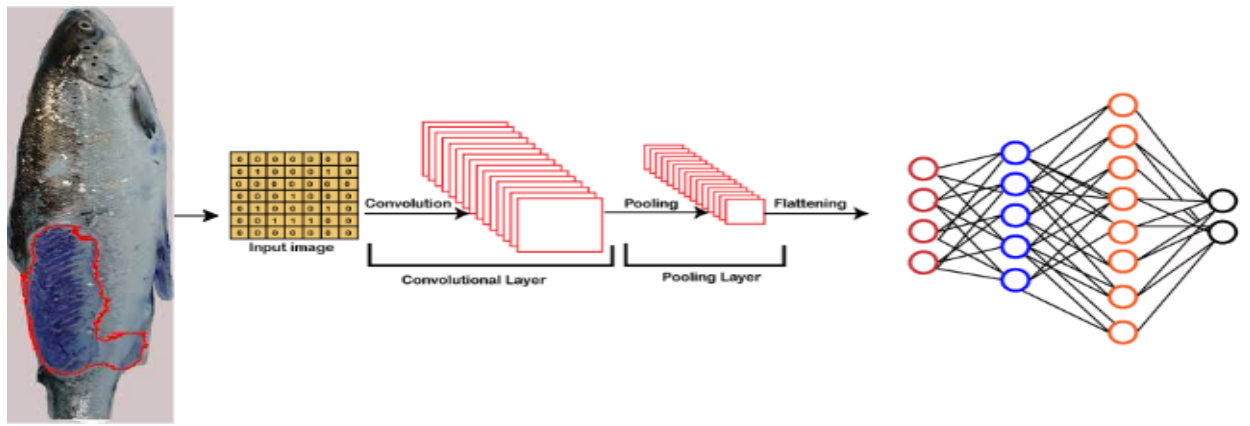


Figure 4: Architecture of CNN

3.3 DenseNet169

DenseNet169, initially trained on various images, is adapted for fish disease detection. It processes input fish disease images of size 224 x 224 pixels with RGB channels. The model's dense connections between layers aid in extracting detailed features crucial for disease identification. During training, it learns to associate specific features with different fish diseases. Fine-tuning adjusts the model's parameters to better suit the characteristics of fish disease datasets, enhancing its performance. In testing, the model accurately classifies unseen fish disease images, providing probabilities for different disease classes. The entire process involves leveraging pre-trained knowledge, adapting to fish disease specifics, and utilizing dense connections for effective feature extraction and classification.

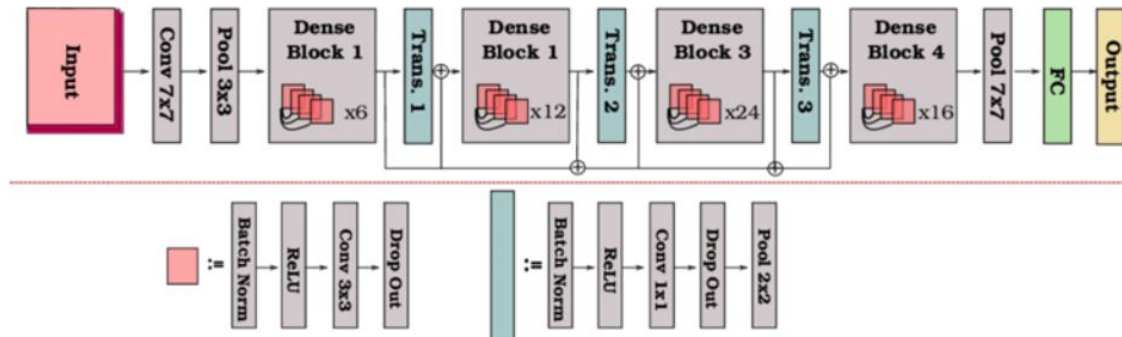


Figure 5: The architecture of Densnet169

3.3 VGG16

VGG16, initially trained on diverse images, is employed for fish disease detection. Input images of fish diseases, sized at 224 x 224 pixels with RGB channels, are processed by the model. The architecture includes convolutional and max-pooling layers to capture features and reduce spatial dimensions. Flattening and fully connected layers further analyse the extracted features. During training, the model learns to associate these features with specific fish diseases. Fine-tuning adjusts parameters for better alignment with fish disease characteristics. In testing, VGG16 accurately classifies new fish disease images based on the learned features. This process utilizes the pre-trained model's knowledge, adapts to fish disease specifics, and employs convolutional layers for effective feature extraction and classification.

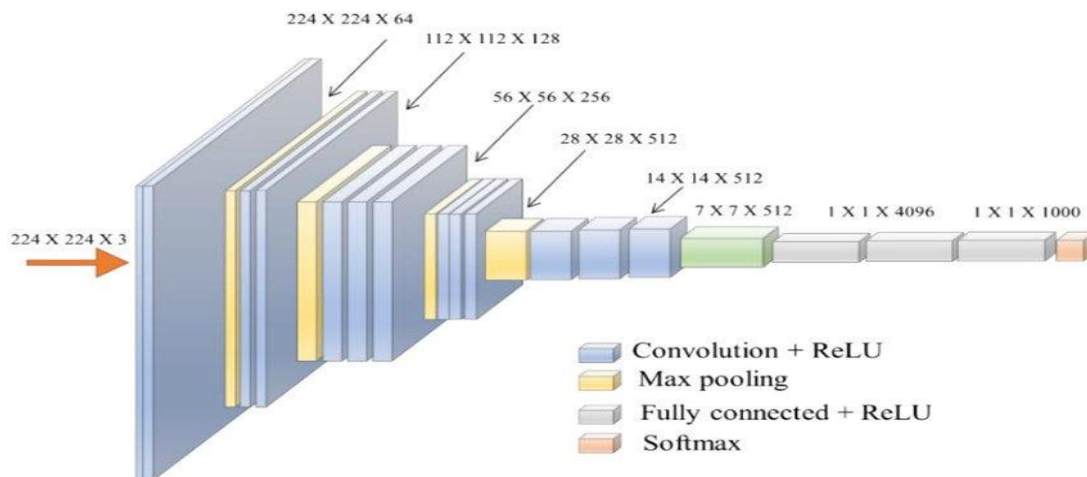


Figure 6: The architecture of VGG16

3.4 Xception

Xception, a deep learning model, is applied to fish disease detection. Initially pretrained on a diverse dataset, Xception is adapted for this task. Input images of fish diseases, resized to 224 x 224 pixels with RGB channels, undergo processing by the model. The architecture includes convolutional blocks with intricate layers. Features are captured through convolutional, batch normalization, activation, and addition layers. Fully connected layers with dropout contribute to the final classification. The model's complexity, with around 22 million parameters, suits it for complex image analysis tasks. In testing, Xception accurately classifies fish disease images by leveraging its deep and intricate design for robust feature extraction and abstraction.

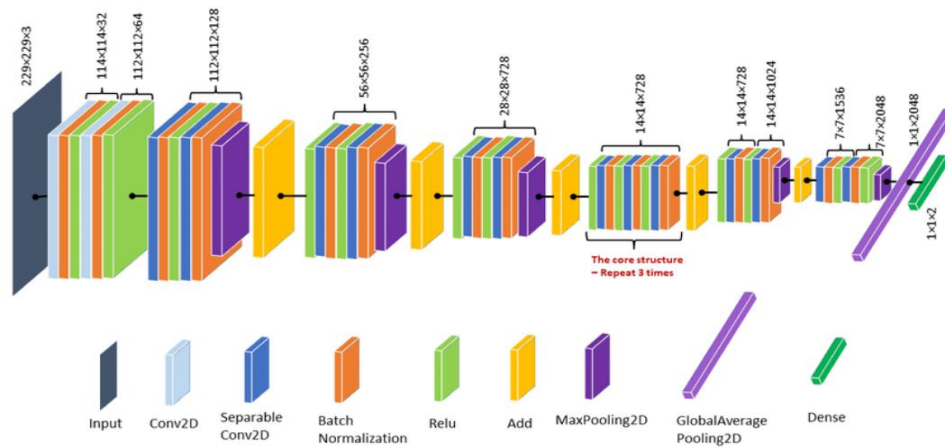


Figure 7: The architecture of Xception

3.4 Ensemble With Voting

In fish disease detection using ensemble with voting, multiple models like DenseNet169, VGG16, and Xception are trained separately. Each model provides predictions for a given fish disease image. During testing, the individual predictions from each model are collected, and a voting mechanism is employed to determine the final classification. The class with the most votes is selected as the ensemble's prediction. This approach leverages the diversity of multiple models, enhancing overall accuracy and robustness. The ensemble's adaptability is showcased by adjusting the voting weights based on individual model performance. By combining insights from various models, the ensemble provides a comprehensive and reliable solution for intricate fish disease classification tasks.

Chapter 4

Methodology

There are a variety of different methods that exist within the field of study. In our research, we sought to develop and evaluate three novel approaches. The processes for each method are detailed below—

4.1 Data preparation

Dataset preparing is the process of gathering or obtaining images from a source. The dataset is collected from Malibagh fish market, Kuril fish market, Rampura fish market Dhaka and also from Chandpur fish farm. Real images of fish with EUS infections have been used in the experiments.

4.2 Feature Extraction

We consider two types of feature extraction techniques: one is statistical features, and another one is grey-level co-occurrence matrix (GLCM) features based on interpreting fish diseases, Dimensionality Reduction, Pattern Recognition, Preprocessing and Normalization, Reduced Data Storage Requirements. Hog (Histogram of Gradient) method of feature extraction has been used in the experiment.

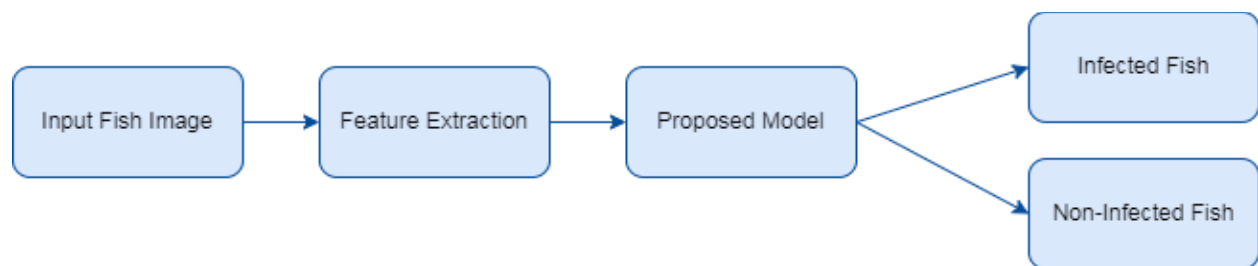


Figure 8: Feature extraction for fish disease detection

We focus on noise reduction through contrast enhancement. Contrast enhancement involves adjusting the intensity levels in the image to make the features more distinguishable. This helps to reduce noise and improve the overall quality of the image. After contrast enhancement, we delve into understanding its impact by comparing the histograms of the original image and the contrast-enhanced version. A histogram is a graphical representation of pixel intensity distribution in an image. In the case of the original image, the histogram showcases the spread of pixel intensities, while the contrast-enhanced image's histogram reveals how the enhancement has altered this distribution. By visually examining these histograms, we gain insights into the changes brought about by contrast enhancement, helping us assess the effectiveness of the noise reduction process. This step aids in ensuring that the image data is appropriately refined for subsequent stages in the preprocessing pipeline, ultimately contributing to the overall success of the image analysis or machine learning task at hand.

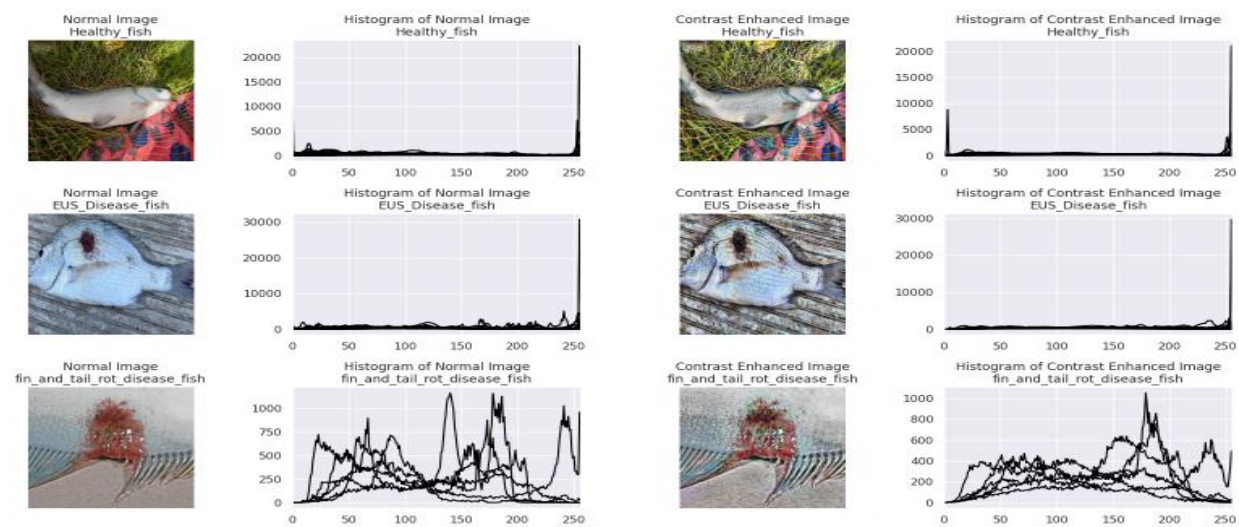


Figure 9: Histogram of Normal Image VS Contrast Enhanced Image

We have extracted a set of features from 6 fish images to capture various characteristics. These features include contrast, correlation, energy, homogeneity, mean, standard deviation, entropy, smoothness, kurtosis, skewness, and IDM (Inverse Difference Moment). Each of these measures provides valuable information about the textures, patterns, and overall content present in the fish images. Contrast reflects the difference in pixel intensity, correlation measures the interdependence of pixel values, energy quantifies the image's clarity, homogeneity gauges the uniformity of pixel distribution, mean and standard deviation offer insights into average brightness and its variability, entropy indicates image complexity, smoothness represents uniformity, and kurtosis and skewness capture the shape of the pixel distribution. IDM reflects the local variations in pixel intensities. By extracting and analyzing these features, we gain a

comprehensive understanding of the visual characteristics of the fish images, setting the stage for more advanced image processing and machine learning tasks.

Feature	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6
Contrast	820.412915	530.891156	530.891156	530.891156	530.891156	530.891156
Correlation	0.896821	0.924236	0.924236	0.924236	0.924236	0.924236
Energy	0.010949	0.011035	0.011035	0.011035	0.011035	0.011035
Homogeneity	0.090745	0.107854	0.107854	0.107854	0.107854	0.107854
Mean	99.408725	112.009	112.009	112.009	112.009	112.009
Std Deviation	63.128883	59.192998	59.192998	59.192998	59.192998	59.192998
Entropy	[-300635.12]	[-299898.72]	[-299898.72]	[-299898.72]	[-299898.72]	[-299898.72]
RMS	9.970392	10.58343	10.58343	10.58343	10.58343	10.58343
Variance	3985.255869	3503.811069	3503.811069	3503.811069	3503.811069	3503.811069
Smoothness	0.999749	0.999715	0.999715	0.999715	0.999715	0.999715
Kurtosis	-0.876894	-0.527764	-0.527764	-0.527764	-0.527764	-0.527764
Skewness	0.509825	0.47204	0.47204	0.47204	0.47204	0.47204
IDM	269.178025	269.178025	269.178025	269.178025	269.178025	269.178025

Figure 10: Extracted the feature for 10 Fish Image

4.3 Feature selection: In feature selection for fish disease detection, we identify and prioritize relevant characteristics from the input data, focusing on those most influential in distinguishing health states. Techniques like statistical measures or machine learning algorithms are applied to retain essential features while discarding less informative ones. This simplifies the model, improves efficiency, and often enhances accuracy by emphasizing the most discriminative aspects in the dataset. To enhance model efficiency and interpretability to avoid overfitting PCA,ICA, LDA, LLE have been used in our experiment.

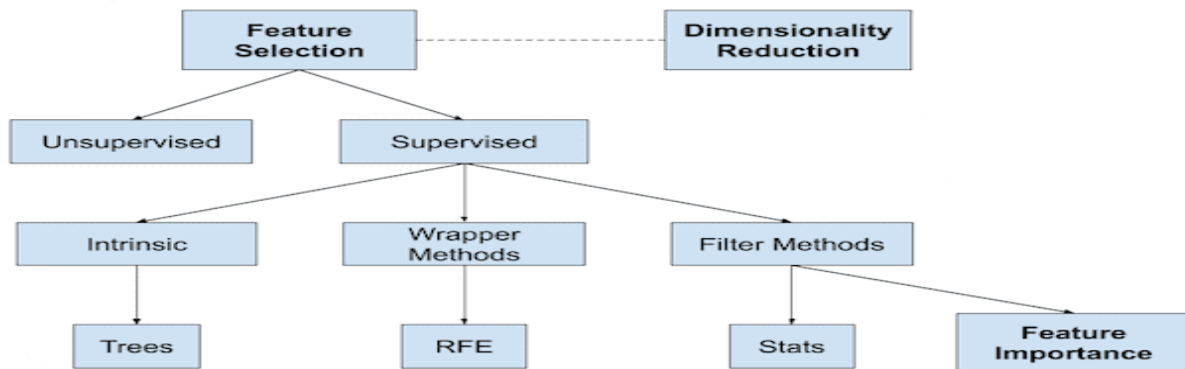


Figure 11: Feature Selection Techniques of Image

4.4 Proposed Method-I

4.4.1 Preprocessing

Edge detection is a computer vision technique aimed at identifying boundaries within an image. It highlights abrupt changes in intensity or color, representing potential edges where objects meet. Common algorithms, like the Canny edge detector, identify these transitions, enabling image analysis and feature extraction crucial for tasks such as object recognition and image segmentation in computer vision applications. Here we used five algorithms. They are as follows:

- a) **Canny edge detection:** Canny edge detection is a popular image processing technique that identifies edges by detecting abrupt changes in intensity, helping to highlight object boundaries and features in images. It involves smoothing, gradient calculation, non-maximum suppression, and edge tracking by hysteresis. The output of this algorithm is as below:

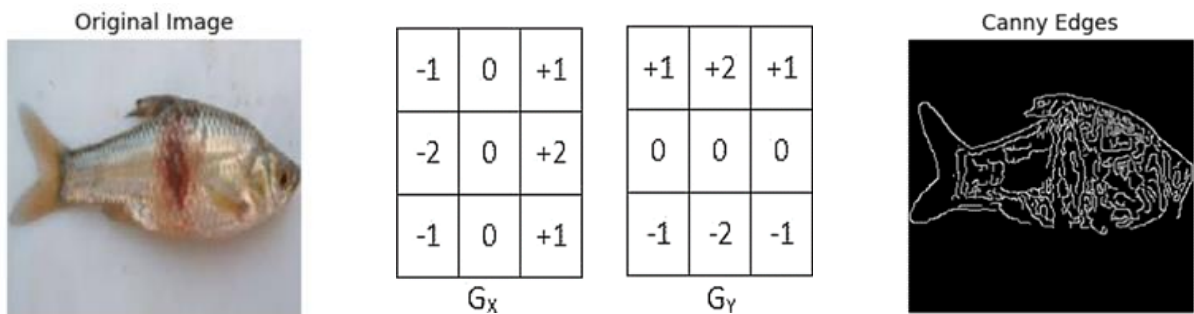


Figure 12: Canny Edge detection

- b) **Prewitt edge detection:** Prewitt edge detection is a gradient-based image processing method that uses convolution with Prewitt kernels to highlight edges by emphasizing intensity changes in horizontal and vertical directions. It is commonly employed for edge extraction in computer vision and image analysis. The output of this algorithm is as below:

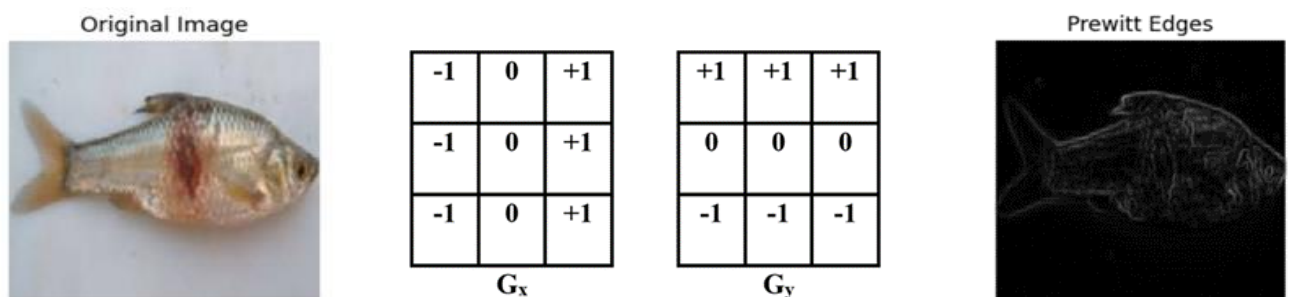


Figure 13: Prewitt Edge detection

- c) Robert edge detection:** Robert edge detection is an image processing technique that employs a pair of 2x2 convolution kernels to emphasize edges by detecting intensity changes in diagonal directions. It is a simple method often used for edge extraction in computer vision applications. The output of this algorithm is as below:

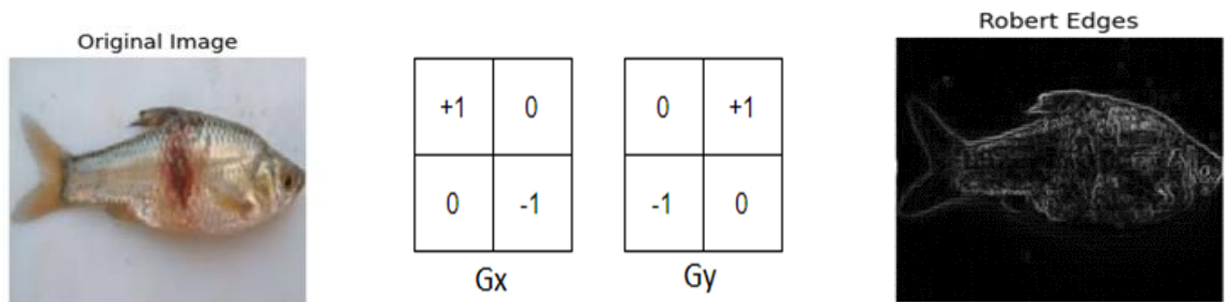


Figure 14: Robert edge detection

- d) Sobel edge detection:** Sobel edge detection is a gradient-based image processing technique that utilizes convolution with Sobel kernels to emphasize edges by detecting intensity changes in horizontal and vertical directions. It is widely used for edge extraction in computer vision and image processing. The output of this algorithm is as below:

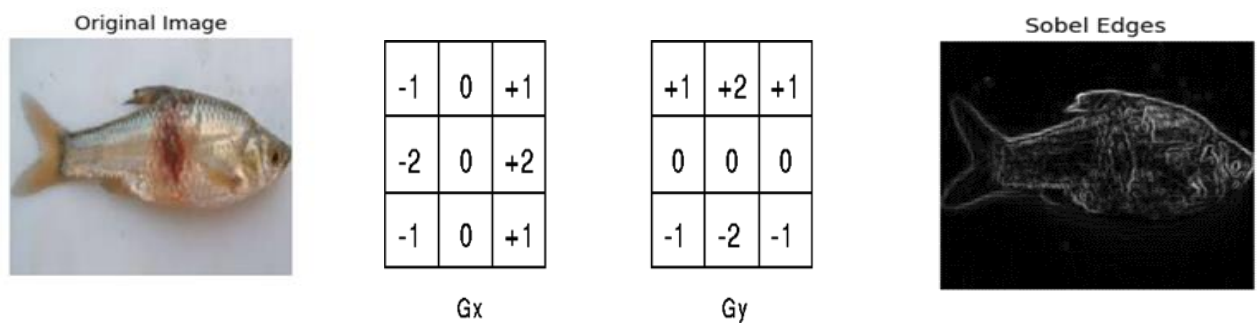


Figure 15: Sobel edge detection

- e) Marr-Hildreth edge detection:** The Marr-Hildreth edge detection, also known as the LoG (Laplacian of Gaussian), combines Gaussian smoothing with Laplacian filtering to highlight edges by identifying zero-crossings in the second derivative of the image. It

aims to enhance edge localization and is used in computer vision for detecting edges and contours. The output of this algorithm is as below:

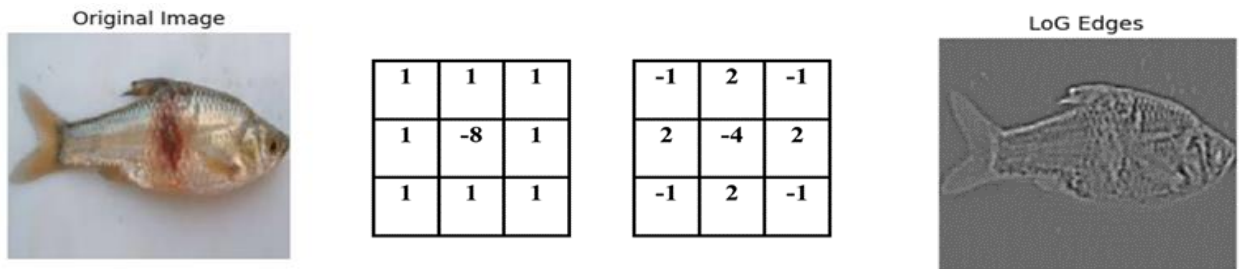


Figure 16: Marr-Hildreth edge detection

4.5 Proposed Method II

4.5.1 Preprocessing

A Convolutional Neural Network (CNN) is a deep learning algorithm designed for Processing and analyzing visual data. It uses convolutional layers to automatically and Adaptively learn hierarchical representations of features from input images, making it particularly effective for tasks such as image recognition and computer vision. The classification has been applied as shown in the figure (fig No).

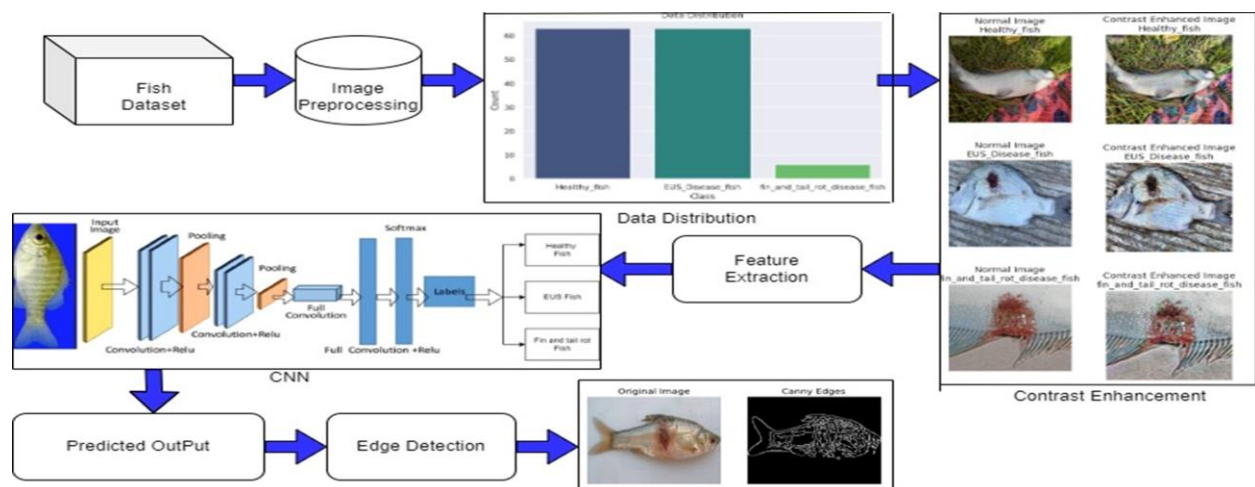


Figure 17: Classification with CNN

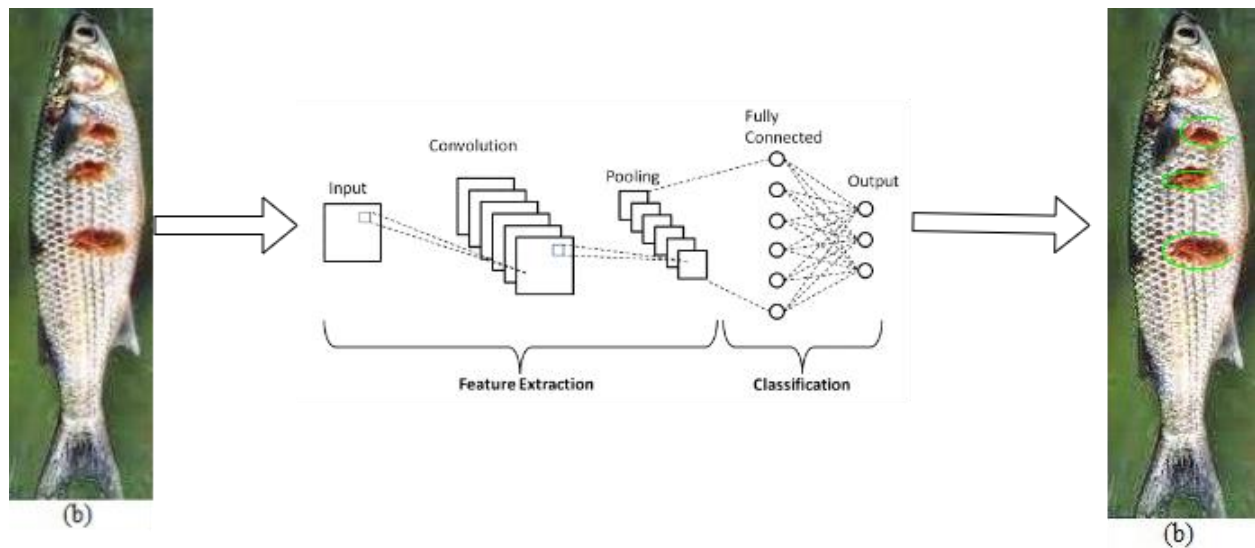


Figure 18: CNN Model

4.6 Proposed Method III

4.6.1 Preprocessing

Ensemble methods are a type of machine learning technique that combines the predictions of multiple models to improve overall performance and robustness. The idea behind ensemble methods is that by combining the predictions of multiple models, the weaknesses of individual models can be mitigated, and a more accurate and stable prediction can be achieved.

4.6.2 VGG16

VGG16 is a convolutional neural network architecture with 16 layers, known for its simplicity and effectiveness in image classification tasks that was proposed by the Visual Geometry Group at the University of Oxford. It became popular for its simplicity and effectiveness in image classification tasks. The "16" in VGG16 refers to the fact that it has 16 weight layers, including 13 convolutional layers and 3 fully connected layers.

Here's a high-level overview of how VGG16 works:

Architecture:

Input Layer: VGG16 takes an input image of fixed size (e.g., 224x224 pixels).

Convolutional Blocks: The network is composed of several convolutional blocks. Each block consists of multiple convolutional layers, followed by a max-pooling layer for spatial downsampling. The convolutional layers use small receptive fields (3x3) with a stride of 1, and the padding is kept the same, resulting in feature maps of the same size.

Fully Connected Layers: After the convolutional blocks, the network has three fully connected layers for high-level reasoning. The final layer produces the output scores for each class in a classification task.

Activation Function: Rectified Linear Units (ReLU) are used as the activation function throughout the network, except for the output layer, where softmax is often used for multi-class classification.

Stacked Convolutional Layers: The network uses multiple convolutional layers with small receptive fields instead of a few layers with large receptive fields. This helps in capturing more complex features and reduces the number of parameters.

Deep Architecture: VGG16 is relatively deep compared to previous architectures, and the deep stack of layers allows the network to learn hierarchical features at different levels of abstraction.

Parameter Efficiency: Despite its depth, VGG16 is parameter-efficient due to the use of small 3x3 filters and max-pooling, which helps in reducing the spatial dimensions.

Training and Transfer Learning: VGG16 is often used as a pre-trained model on large datasets like ImageNet. Transfer learning, where the model is fine-tuned on a specific task, is a common practice and has proven effective in various computer vision applications.

While VGG16 has been influential, more recent architectures like ResNet and EfficientNet have surpassed it in terms of both accuracy and efficiency for many tasks.

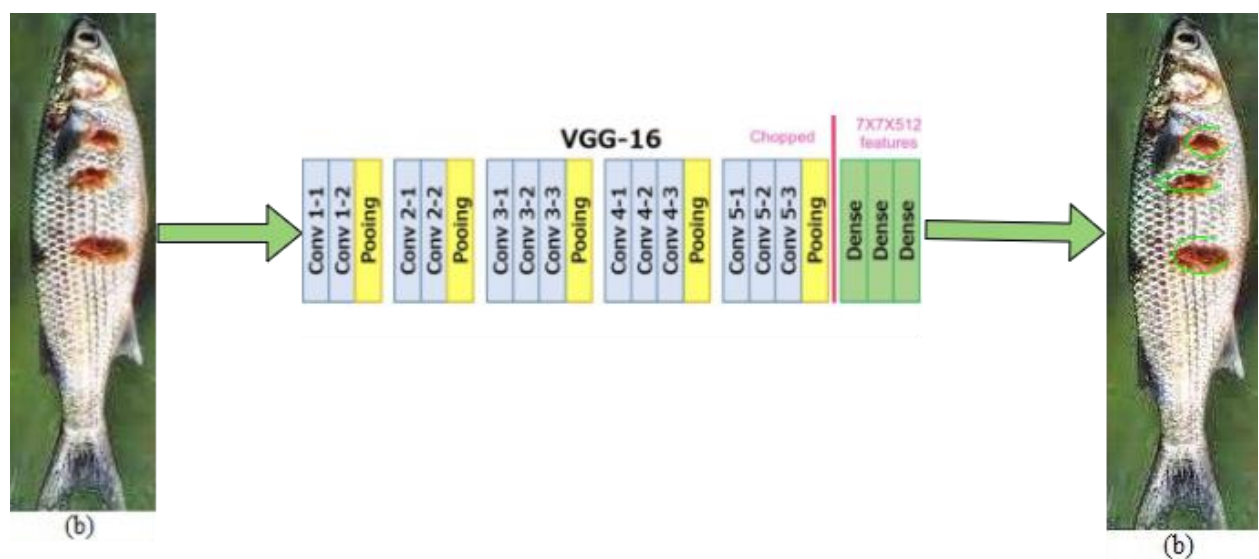


Figure 19: VGG-16 model

4.6.3 DenseNet169

DenseNet169 is a densely connected neural network that maximizes information flow between layers by connecting each layer to every other layer in a feedforward fashion, enhancing feature reuse and gradient flow that belongs to the family of Densely Connected Convolutional Networks (DenseNets). It was introduced by Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger in their paper titled "Densely Connected Convolutional Networks" (CVPR 2017).

Here's a general overview of how DenseNet169 works:

Densely Connected Blocks: The key innovation in DenseNet is the concept of densely connected blocks. Unlike traditional convolutional neural networks where each layer is connected only to the previous one, in DenseNet, each layer is connected to every subsequent layer within a dense block. This results in dense connectivity between layers, allowing for feature reuse and facilitating the flow of information through the network.

Architecture:

Input Layer: Like many other CNNs, DenseNet169 takes an input image, typically of size 224x224 pixels.

Initial Convolution: The network starts with a single convolution layer to extract initial features from the input.

Dense Blocks: DenseNet comprises multiple dense blocks, each consisting of a series of densely connected layers. Within a dense block, the output of each layer is concatenated with the feature maps of all preceding layers. This dense connectivity enhances feature reuse and enables the network to learn more discriminative features.

Transition Layers: Between dense blocks, transition layers are used to reduce the spatial dimensions of the feature maps. They consist of a batch normalization layer, a 1x1 convolution layer for dimensionality reduction, and a 2x2 average pooling layer.

Global Average Pooling (GAP): After the last dense block, a global average pooling layer is applied to reduce the spatial dimensions to a 1x1 feature map for each channel. This produces a fixed-size representation regardless of the input size.

Fully Connected Layer: The network typically ends with a fully connected layer, and a softmax activation function is applied for classification in the case of image classification tasks.

Advantages of DenseNet:

Parameter Efficiency:

The dense connectivity allows the network to be parameter-efficient, as features are shared and reused throughout the network.

Gradient Flow: The dense connections help alleviate the vanishing gradient problem, as gradients can flow directly through the densely connected paths during backpropagation.

Feature Reuse: DenseNet encourages feature reuse, making the network more effective in learning representations.

DenseNet169, specifically, refers to a variant of DenseNet with 169 layers. Other variants include DenseNet121, DenseNet201, and so on, which differ in the number of layers. These architectures are often pre-trained on large datasets like ImageNet and can be fine-tuned for specific tasks using transfer learning.

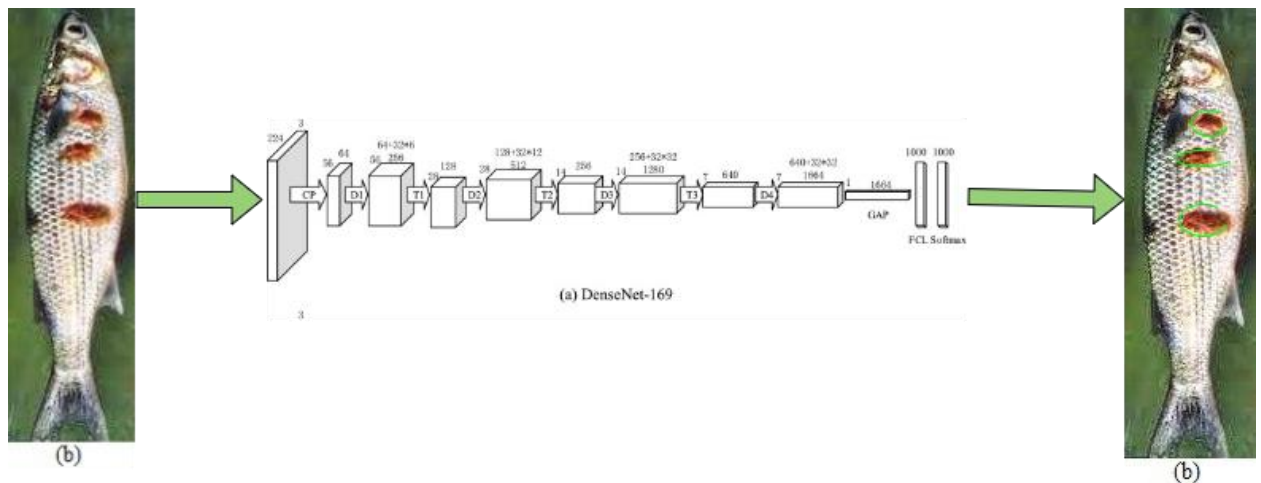


Figure 20:DenseNet-169 model

4.6.4 Xception

Xception is short for "Extreme Inception", which is a deep neural network architecture that employs depthwise separable convolutions to capture complex patterns while maintaining computational efficiency, making it suitable for various computer vision tasks that were introduced by François Chollet, the creator of the Keras deep learning library. Xception is designed based on the idea of separating the learning of cross-channel and spatial correlations, inspired by the Inception architecture. It was proposed in the paper titled "Xception: Deep Learning with Depth Wise Separable Convolutions" (CVPR 2017).

Features and Working Principle:

Depthwise Separable Convolutions: The core building block of Xception is the depthwise separable convolution. Traditional convolutions involve a cross-channel correlation followed by a spatial correlation. Depthwise separable convolutions, on the other hand, split these two operations. The depthwise separable convolution consists of a depthwise convolution (a spatial convolution for each channel independently) followed by a pointwise convolution (a 1x1 convolution to mix the information across channels). This architecture reduces the number of parameters and computational complexity compared to traditional convolutions.

Entry Flow: Xception starts with a series of convolutional and depthwise separable convolutional layers forming the entry flow. These layers capture low-level features from the input data.

Middle Flow: The middle flow consists of multiple residual modules, each containing a sequence of depthwise separable convolutions. These modules help the network capture more abstract and higher-level features.

Exit Flow: The exit flow is a series of depthwise separable convolutional layers and a global average pooling layer. The global average pooling layer reduces the spatial dimensions to a 1x1 feature map for each channel.

Fully Connected Layer: The network typically ends with a fully connected layer, and a softmax activation function is applied for classification in the case of image classification tasks.

Advantages of Xception:

Parameter Efficiency: Xception is more parameter-efficient compared to traditional architectures like Inception, as it uses depth wise separable convolutions, which have fewer parameters.

Increased Representational Power: The separation of cross-channel and spatial correlations is intended to increase the representational power of the network.

Better Gradient Flow: The architecture's design, inspired by the depth wise separable convolutions, helps in better gradient flow during backpropagation, potentially reducing the risk of vanishing gradients.

Xception has been shown to achieve competitive performance on various computer vision tasks, and it is often used as a pre-trained model for transfer learning on specific tasks or datasets. It's worth noting that Xception is particularly useful when computational resources are limited, as it provides a good balance between accuracy and efficiency.

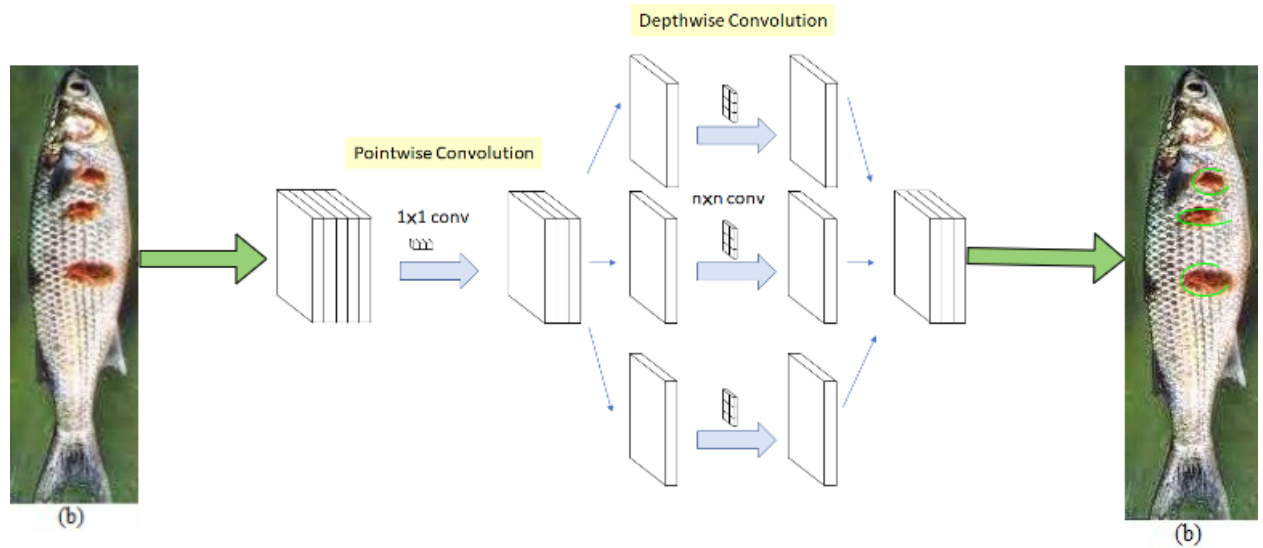


Figure 21: Xception Model

Chapter 5

Dataset

In the initial stages of developing our deep learning model for fish recognition, our team invested a considerable amount of time, approximately one month, in meticulous planning. The primary objective was to gather a diverse set of fish images from various aquatic environments in Bangladesh. This involved identifying specific locations within the country where such images could be obtained.

Through extensive discussions and strategic planning, we pinpointed five key locations in Bangladesh—Jamalpur, Barishal, Chittagong, Sylhet, and Khulna. These areas were chosen based on their accessibility, affordability, and potential for providing a wide range of fish images. Additionally, we enlisted the support of local well-wishers who proved instrumental in not only suggesting these locations but also offering insights into the aquatic ecosystems and prevalent fish diseases.

Our well-wishers played a crucial role in providing guidance, sharing their knowledge about the selected locations, and assisting us in identifying potential challenges within the fish cultures, including the presence of diseases. This collaborative effort allowed us to create a comprehensive plan for collecting images from each of the selected places.

One major challenge we encountered during the image collection process was normalizing images taken with different devices. To address this issue, we devised a solution by capturing images with specific categorical image pixels. Subsequently, we established a dedicated repository to systematically organize and upload fish images on a daily basis. This meticulous approach not only facilitated the normalization process but also streamlined our data collection efforts.

In the following table, we have provided detailed information, including the division, location, and image count, for each person involved in the image collection process. This summarized overview encapsulates the comprehensive planning, collaboration, and problem-solving strategies employed by our team in the pursuit of building an effective deep learning model for fish recognition.

Division	Person Name	Device Used	Number of images
Jamalpur	Anwar Hossain	Samsung	5000
Barishal	Md. Rubel Hossen	Vivo	5036
Chittagong	Polash Hasan	Oppo	3000
Sylhet	Azhar Islam	VivoY20	7000
Khulna	Rifat Ebne Joynal	Tecno	5111

Table 1: Third Dataset Information

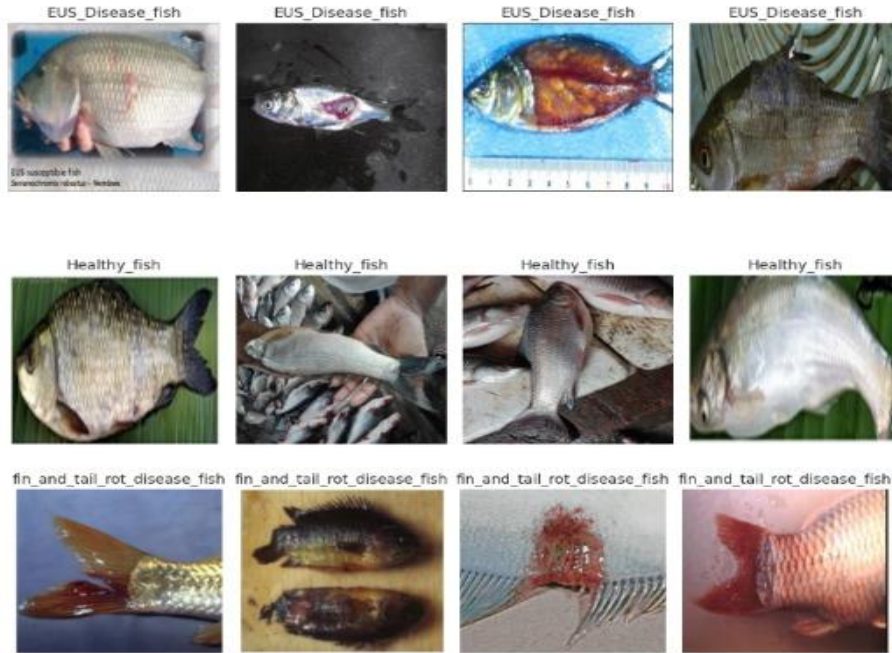


Figure 22: Sample images from our dataset.

Now the dataset information for the final dataset is given below:

Dataset	Information
Name of Dataset	Raw Fish Image Datasets
Image size	224 x 224 (Resized)
Number of images	25,147
Image type	JPEG (RGB)
Class labels	0. Healthy Fish 1. EUS Disease Fish 2. Fin And Tail Rot Disease Fish

Table 2: Dataset Information

The quantity of images in each category is as follows:

Fish Type	Count of Images
Healthy Fish	14,000
EUS Disease Fish	7047
Fin And Tail Rot Disease Fish	4100

Table 3: Number of images per classification type

Chapter 06

Experiment Analysis

6.1 Productive Setup

Here, we'll go over how our workstations were set up as well as model's hyperparameters, the techniques we employed to carry out our tests, and the metrics we employed to assess the outcomes. Table ----- demonstrates the use of an Intel Core i7-12700 CPU and 32GB of RAM. On a PC equipped with a GPU (graphics processing unit) from Nvidia's GeForce RTX 3060 Ti series, the recommended model was trained. Tensorflow 2.6.0, Keras 2.3.1, and Python 3.9 make up the software stack under test.

Software or Hardware	Specification
CPU	Intel Core i7-12700
GPU	GeForce RTX 3060 Ti
RAM	DDR4 32 GB
Python	3.10
Tensorflow	2.6.0
Keras	2.14.0

Table 4: Hardware Setup.

Table ----- illustrates that the learning rate was set to 0.0001 and the epochs to 20 as hyperparameters. Furthermore, as said, ReLU will serve as the activation function and Adam will be the optimizer. Categorical cross-entropy will be the loss function, and 32 is the batch size with a 0.2 dropout rate.

Hyperparameters	Values
Optimizer	Adam
Batch size	32
Dropout	0.2
Loss function	Categorical cross-entropy
Epochs	80(Basic CNN) 20(DenseNet,VGG16,Xception)
Learning rate	0.0001

Table 5: Hyperparameter Setup.

6.2 Experiment Analysis-I

6.2.1 Procedure

Basic CNN

Convolutional Neural Networks (CNNs) are composed of several layers that work together to learn hierarchical representations of input data, particularly suited for image recognition tasks. Here's an overview of the key layers commonly found in CNN architectures:

1. **Input Layer:** The input layer represents the raw input data, typically images. Each pixel in the image is considered as a neuron in this layer.
2. **Convolutional Layer:** Convolutional layers are the core building blocks of CNNs. They apply convolutional operations to input data using learnable filters (kernels). These filters slide across the input, capturing local patterns and features. Parameters: Number of filters, filter size, stride, padding.
3. **Activation Layer (ReLU):** After each convolutional operation, an activation function is applied to introduce non-linearity. The Rectified Linear Unit (ReLU) activation is commonly used, allowing only positive values to pass through. ReLU: $f(x) = \max(0, x)$
4. **Pooling Layer:** Pooling layers downsample the spatial dimensions of the feature maps. Max pooling is a common technique, where the maximum value within a region (pooling window) is retained. Pooling helps reduce computational complexity and extract dominant features. Parameters: Pooling type (e.g., max pooling), pool size, stride.
5. **Batch Normalization Layer:** Batch normalization normalizes the input of a layer by adjusting and scaling its activations. It helps stabilize and speed up the training process. Parameters: Scale, shift, momentum.
6. **Dropout Layer:** Dropout layers randomly deactivate a specified percentage of neurons during training. This regularization technique helps prevent overfitting by promoting redundancy in the network. Parameter: Dropout rate.
7. **Flatten Layer:** The flatten layer is used to convert the 2D feature maps into a 1D vector, preparing the data for input to fully connected layers.
8. **Fully Connected Layer (Dense Layer):** Fully connected layers are traditional neural network layers where each neuron is connected to every neuron in the previous and next layers. They are typically used at the end of a CNN to make predictions based on the learned features. Parameters: Number of neurons, activation function.

9. Output Layer: The output layer produces the final predictions based on the learned features. For classification tasks, a softmax activation is often used to convert the network's output into probability scores for each class.

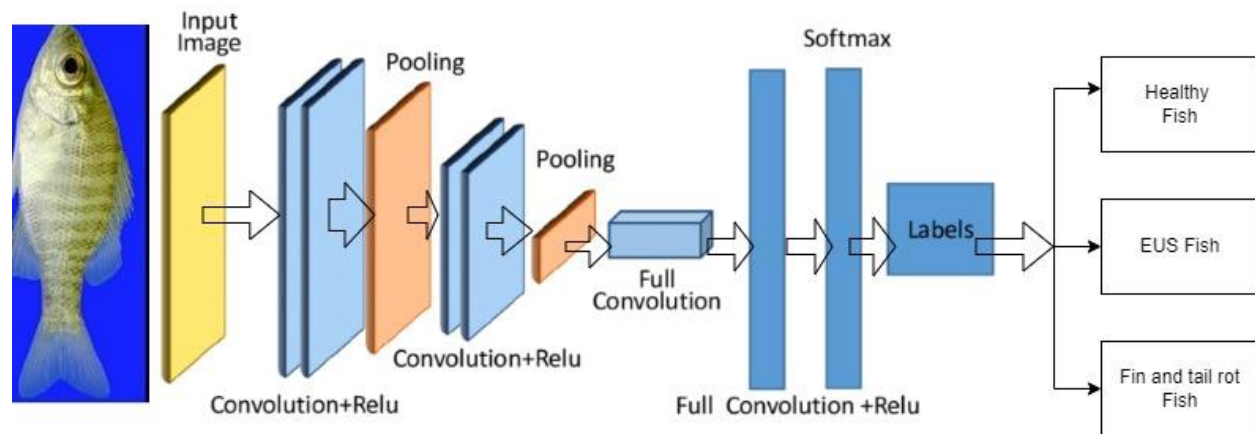


Figure 23: Basic CNN Architecture.

The model comprises several convolutional blocks. Each block consists of a convolutional layer, batch normalization, and an activation function (commonly ReLU). This setup helps the network learn hierarchical features from the input data. Addition layers are employed, indicating the use of skip connections or residual connections. These connections facilitate the flow of gradients during training, addressing the vanishing gradient problem. A max-pooling layer is applied to down-sample the spatial dimensions of the feature maps, reducing computational complexity and capturing dominant features.

After the convolutional blocks, there is a final convolutional layer ('conv2d_2'). This layer likely captures high-level abstractions from the learned features. A flattening layer follows, transforming the multi-dimensional feature maps into a one-dimensional vector for feature extraction. Two fully connected layers ('dense_4' and 'dense_5') are included for learning global patterns and relationships in the data. Dropout is applied to these layers for regularization, preventing overfitting. The architecture concludes with a classification layer. This layer likely employs a softmax activation function, especially if the task is multi-class classification. The number of neurons in this layer corresponds to the number of classes in the dataset.

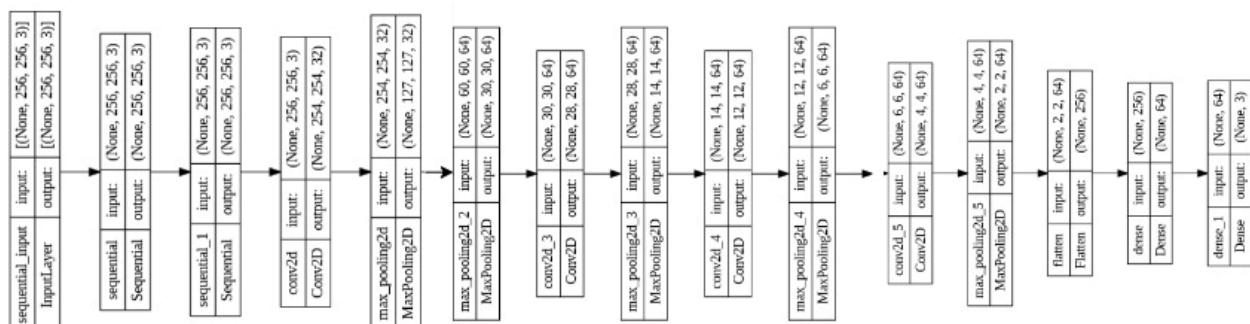


Figure 24: The Configuration of Basic CNN Model.

Fish Images	Actual Fish	Predicted Fish	Accuracy
Image 1	EUS_Disease_fish	EUS_Disease_fish	99.43%
Image 2	EUS_Disease_fish	EUS_Disease_fish	96.25%
Image 3	EUS_Disease_fish	EUS_Disease_fish	81.37%
Image 4	Healthy_fish	Healthy_fish	97.34%
Image 5	Healthy_fish	Healthy_fish	83.05%
Image 6	Fish_Disease_fin_and_tail_rot	Fish_Disease_fin_and_tail_rot	98.51%
Image 7	EUS_Disease_fish	EUS_Disease_fish	81.37%
Image 8	Healthy_fish	Healthy_fish	99.99%
Image 9	EUS_Disease_fish	EUS_Disease_fish	86.75%
Image 10	EUS_Disease_fish	EUS_Disease_fish	92.29%

Table 6: Results for 10 different images using Basic CNN Model

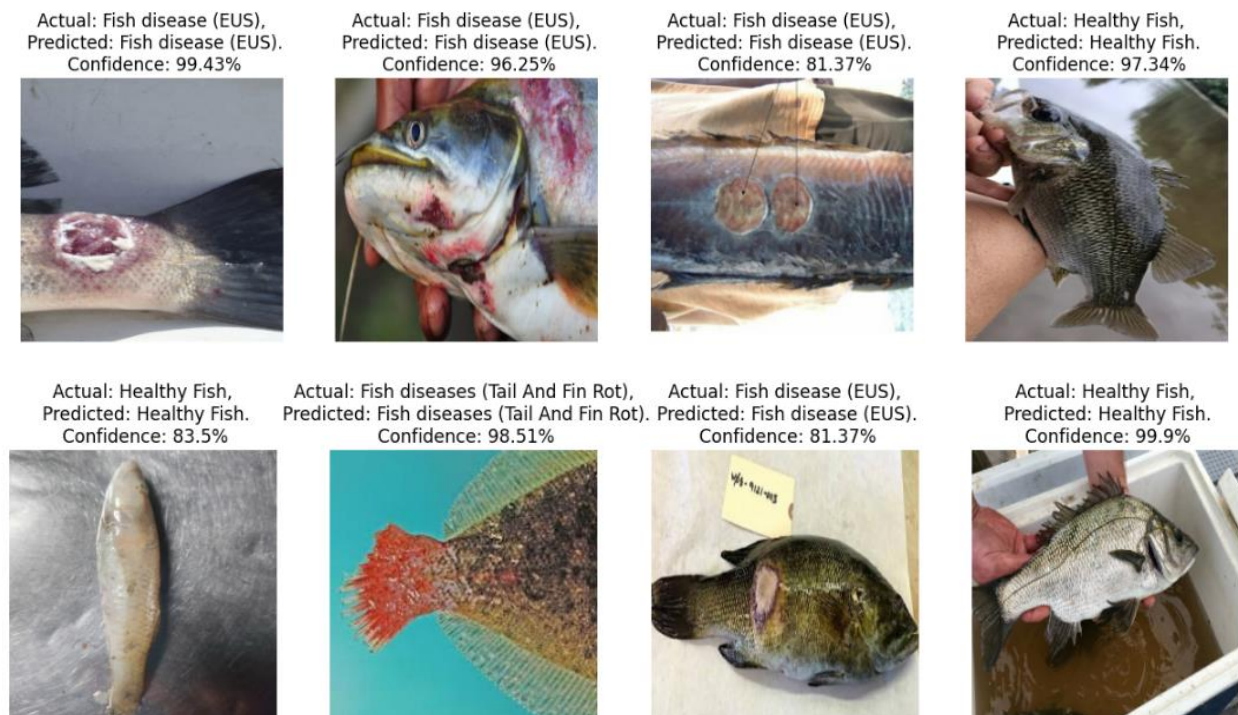


Figure 25: Basic CNN Result Prediction

6.3 Experiment Analysis-II

6.3.1 Procedure of DenseNet169

The image sizes that we have here range from (103 x 235 x 3) to (4160 x 3120 x 3). However, the photos have been adjusted to match the input image dimensions of 224 x 224 x 3. The code builds a neural network model by utilizing Keras. 'Imagenet' weights are loaded into a pre-trained DenseNet169 model to begin feature extraction. To maintain the pre-trained knowledge, the pre-trained model's layers are all frozen. To customize the model for a given classification task, more custom layers are added to it.

Layers	Operations	Output
Input layer	Input images with a size of 224 pixels in width, 224 pixels in height, and 3 color channels	224 x 224 x 3
Convolution Layer	Reducing the spatial dimensions to 112x112 while increasing the number of channels to 64	112 x 112 x 64
Max pooling 2D	2x2 pool size by half and resulting 56x56 with the same 64 channels	56 x 56 x 64
Batch Normalization	Normalize the activations in the 56x56x64 feature maps	56 x 56 x 256
Average pooling 2D	2x2 pool size further reduces the spatial dimensions	28 x 28 x 128
Convolution 2D	32 filters is applied to the 28x28x128 feature maps.	28 x 28 x 32
Concatenation	Feature maps from previous layers are concatenated to create a dense connection	7 x 7 x 1664
Batch Normalization	Batch normalization is applied again to the concatenated feature maps	7 x 7 x 1664
Convolution 2D	64 filters is applied, further processing the feature maps	5 x 5 x 64
Max pooling 2D	2x2 pool size reduces the spatial dimensions to 2x2 while maintaining 64 channels.	2 x 2 x 64
Flatten	The output is flattened into a 1D vector with 256 elements	256
Dense	3 units represents the output layer for classification into three classes, with a softmax activation function	256
Dropout	A dropout layer with 256 units is applied to reduce overfitting	256
Dense	final fully connected dense layer with 3 units	3

Table 7: The Configuration of DenseNet.

With as many output units as there are classes in a given task, the output layer is set up for multi-class classification. The input images are given class probabilities using the softmax activation function. Model is developed and compiled using the proper loss and optimization functions after the architecture has been established. Lastly, a summary of the model's architecture is shown, along with the quantity of parameters. An image classification model can be created and trained using this code as a basis. Strong feature extraction is provided by the pre-trained DenseNet169, and it is customized for the intended classification task with the help of custom layers.

Fish Images	Actual Fish	Predicted Fish	Accuracy
Image 1	EUS_Disease_fish	EUS_Disease_fish	74.41%
Image 2	Healthy_fish	Healthy_fish	98.09%
Image 3	EUS_Disease_fish	EUS_Disease_fish	50.41%
Image 4	EUS_Disease_fish	EUS_Disease_fish	98.27%
Image 5	EUS_Disease_fish	EUS_Disease_fish	97.67%
Image 6	fin_and_tail_rot_disease_fish	Healthy_fish	65.93%
Image 7	Healthy_fish	Healthy_fish	96.15%
Image 8	fin_and_tail_rot_disease_fish	EUS_Disease_fish	94.45%
Image 9	EUS_Disease_fish	EUS_Disease_fish	73.28%
Image 10	EUS_Disease_fish	EUS_Disease_fish	92.45%

Table 8: Results for 10 different images using DenseNet169



Figure 26: Result Prediction of DenseNet169

6.4 Experiment Analysis-III

6.4.1 Procedure of VGG16

The VGG16 architecture, a well-known convolutional neural network for image classification, serves as the foundation for the neural network model that is being described. An input layer with 224x224 pixel RGB images is where it starts. The model is composed of several convolutional layers, designated by labels such as "blockX_convY," where X and Y are variables that denote distinct layers that gradually extract features from the input. Across these layers, the output shapes vary, with the number of channels increasing and the spatial dimensions decreasing.

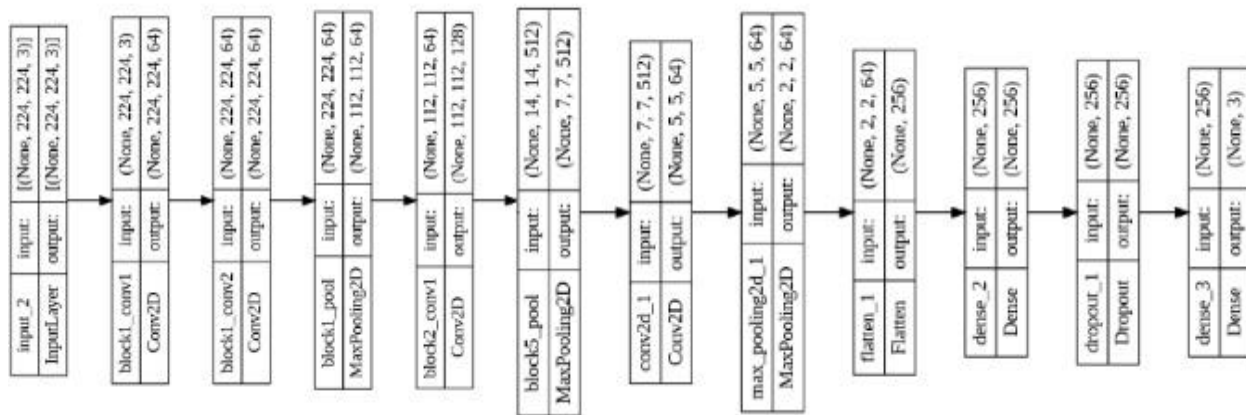


Figure 27: The Configuration of VGG16.

In order to cut the spatial dimensions in half, max-pooling layers, also known as "blockX_pool," come after some convolutional layers. The feature maps are further processed by additional custom convolutional layers and a max-pooling layer. After converting the 2D feature maps into a 1D vector using a "Flatten" layer, the 1D vector is sent to a fully connected dense layer consisting of 256 units. By randomly deactivating neurons during training, a dropout layer with 256 units helps avoid overfitting. Using a softmax activation function to provide class probabilities, the final layer, "dense_9," is fully connected with three units and represents the output layer for classifying input images into one of three classes.

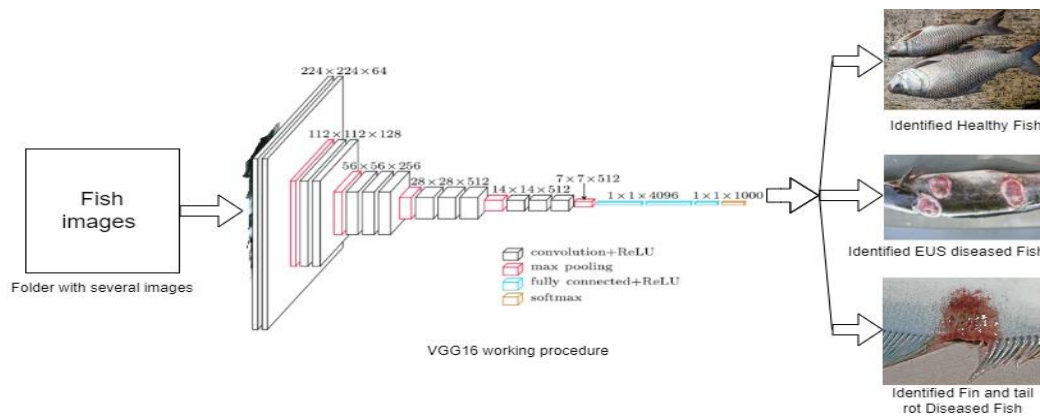


Figure 28: Detailed Information of the Proposed Model VGG16.

The model contains roughly 15.08 million parameters total. Of these, 14.71 million parameters are non-trainable and come from the pre-trained VGG16 model, whereas 361,539 parameters are trainable, which means they can be updated during training to adapt to specific tasks. VGG16 is a well-liked option in the deep learning community because of its ease of use and efficiency in image classification tasks.

Fish Images	Actual Fish	Predicted Fish	Confidence
Image 1	EUS_Disease_fish	EUS_Disease_fish	82.41%
Image 2	Healthy_fish	Healthy_fish	57.09%
Image 3	EUS_Disease_fish	EUS_Disease_fish	72.41%
Image 4	EUS_Disease_fish	EUS_Disease_fish	74.47%
Image 5	EUS_Disease_fish	EUS_Disease_fish	81.55%
Image 6	EUS_Disease_fish	EUS_Disease_fish	77.98%
Image 7	EUS_Disease_fish	EUS_Disease_fish	97.66%
Image 8	EUS_Disease_fish	EUS_Disease_fish	76.93%
Image 9	Healthy_fish	EUS_Disease_fish	57.01%
Image 10	EUS_Disease_fish	Healthy_fish	60.12%

Table 9: Results for 10 different images using VGG16



Figure 29: Result prediction of VGG16

6.5 Experiment Analysis-IV

6.5.1 Procedure of Xception

The architecture given is an example of a multi-layered deep convolutional neural network (CNN) model intended for image classification applications. Its foundation is the architecture of Xception. The model is made up of several blocks, each of which has addition layers, batch normalization, activation functions, and separable convolutional layers. Max-pooling layers come after these blocks, and fully connected layers for classification complete the model. The model's complexity is evident from the roughly 22 million parameters it contains overall. Its depth and complex design make it ideal for more complex image analysis tasks.

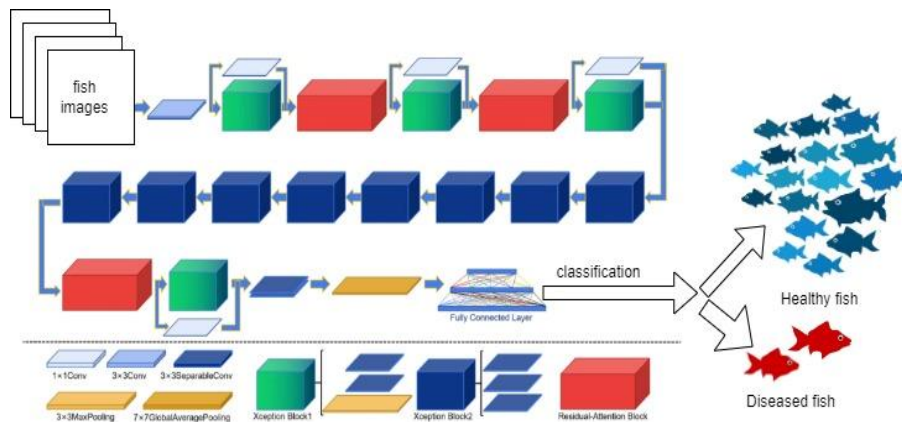


Figure 30: Detailed Information of the Proposed Model Xception.

The layers of a model of a convolutional neural network are described. It consists of several convolutional blocks with convolutional layers, batch normalization, activation functions, and addition layers a max-pooling layer and the final convolutional layer ({conv2d_2}) come after these blocks, and a flattening layer for feature extraction follows. {dense_4} and `dense_5` are the next fully connected layers in the model, which also uses dropout for regularization. The classification layer is the pinnacle of the architecture.

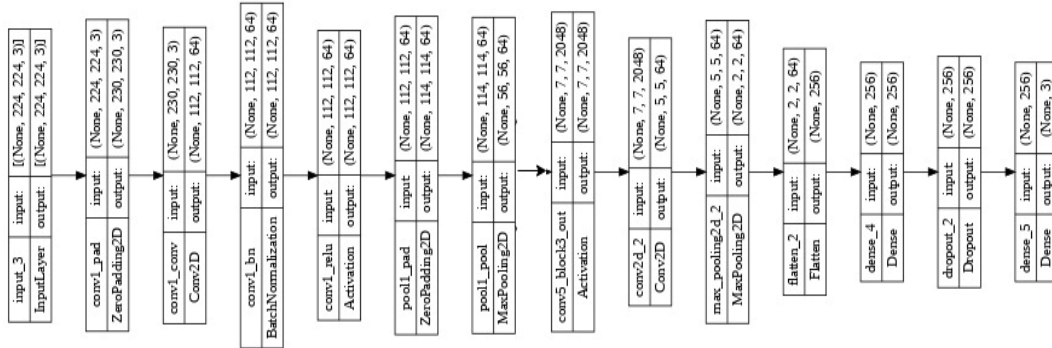


Figure 31: The Configuration of Xception.

Fish Images	Actual Fish	Predicted Fish	Confidence
Image 1	Healthy_fish	Healthy_fish	80.36%
Image 2	EUS_Disease_fish	EUS_Disease_fish	87.77%
Image 3	EUS_Disease_fish	EUS_Disease_fish	74.21%
Image 4	EUS_Disease_fish	Healthy_fish	60.16%
Image 5	EUS_Disease_fish	EUS_Disease_fish	78.07%
Image 6	EUS_Disease_fish	Healthy_fish	62.98%
Image 7	Healthy_fish	Healthy_fish	74.83%
Image 8	EUS_Disease_fish	EUS_Disease_fish	74.46%
Image 9	EUS_Disease_fish	EUS_Disease_fish	85.75%
Image 10	EUS_Disease_fish	EUS_Disease_fish	92.29%

Table 10: Results for 10 different images using Xception



Figure 32: Result prediction of Xception

The provided code showcases an ensemble learning strategy that harmoniously integrates predictions from three distinct models—densenet, vgg16, and xception—utilizing a weighted averaging approach. The ensemble's robustness is enhanced by assigning specific weights to each model based on their respective performance metrics. The generation of predictions is facilitated by a well-crafted function, `generate_predictions`, which efficiently iterates over batches of a data generator, applying the individual models and aggregating the results. The subsequent weighted averaging process, governed by specified weights, amalgamates the models' predictions to form a cohesive ensemble prediction. The final evaluation of the ensemble's performance is conducted through key classification metrics such as accuracy, precision, recall, and F1 score, providing a comprehensive assessment of its effectiveness on the given test dataset. The adaptability of the ensemble is highlighted by the ability to fine-tune weights based on the unique strengths and weaknesses of each model, thereby offering a versatile and robust solution for complex classification tasks.

6.6 Ensemble of DenseNet169, VGG16 And Xception

It showcases an ensemble learning strategy that harmoniously integrates predictions from three distinct models—densenet, vgg16, and xception—utilizing a weighted averaging approach. The ensemble's robustness is enhanced by assigning specific weights to each model based on their respective performance metrics. The generation of predictions is facilitated by a well-crafted function, `generate_predictions`, which efficiently iterates over batches of a data generator, applying the individual models and aggregating the results. The subsequent weighted averaging process, governed by specified weights, amalgamates the models' predictions to form a cohesive ensemble prediction.

The final evaluation of the ensemble's performance is conducted through key classification metrics such as accuracy(61%), precision, recall, and F1 score, providing a comprehensive assessment of its effectiveness on the given test dataset. The adaptability of the ensemble is highlighted by the ability to fine-tune weights based on the unique strengths and weaknesses of each model, thereby offering a versatile and robust solution for complex classification tasks.

6.7 Comparison of the Models Accuracy with Graph

6.7.1 Pre-trained Models

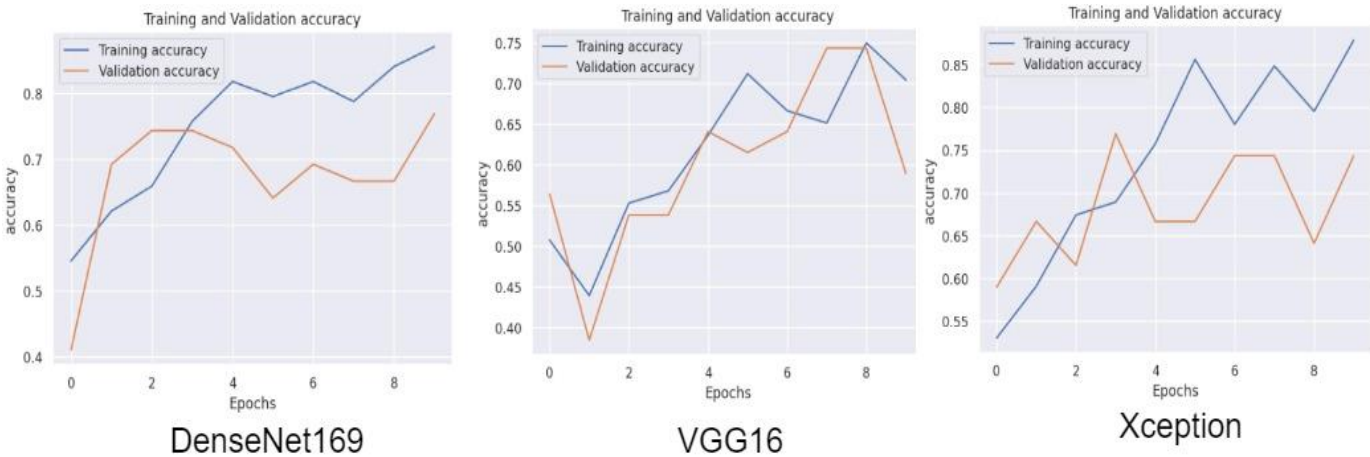


Figure 33: Model evaluation using graph

6.7.2 Basic CNN Model

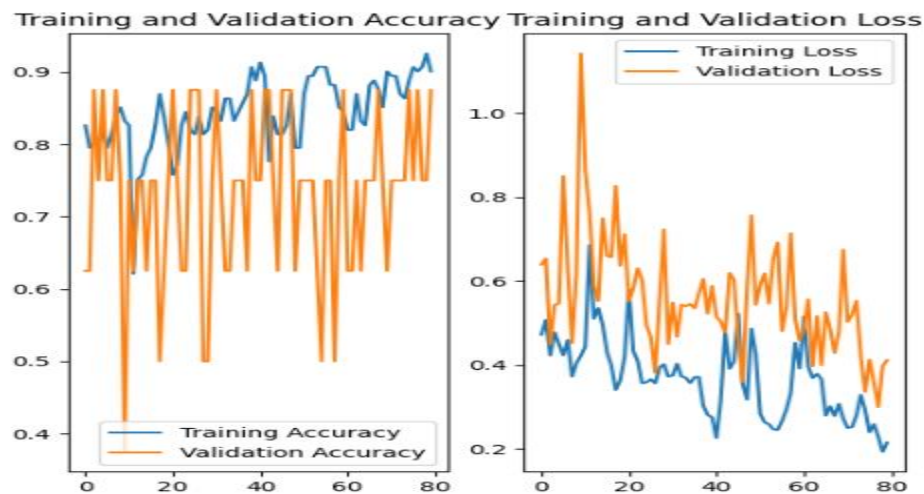


Figure 34: CNN Model

6.8 Experiment Analysis-V

6.8.1 Canny Edge Detection

Canny Edge Detection is a popular image processing technique designed to identify edges in images with precision and noise reduction. Developed by John F. Canny, the algorithm operates through a series of steps, starting with the computation of image gradients to highlight areas of rapid intensity change.

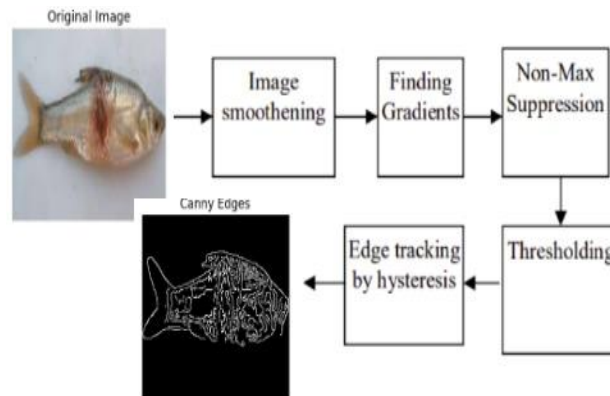


Figure 35: Process steps of canny edge detection

Subsequent stages involve determining gradient magnitude and direction, thinning edges through non-maximum suppression, and refining the results using edge tracking with hysteresis. By setting appropriate thresholds, the algorithm distinguishes strong and weak edges, connecting them to form continuous contours. The final output is a binary image where pixels are classified as either part of an edge or non-edge, providing a clear representation of the prominent structural features in the original image. Canny Edge Detection is widely used in computer vision for tasks such as object recognition and image segmentation due to its accuracy and resilience against noise.

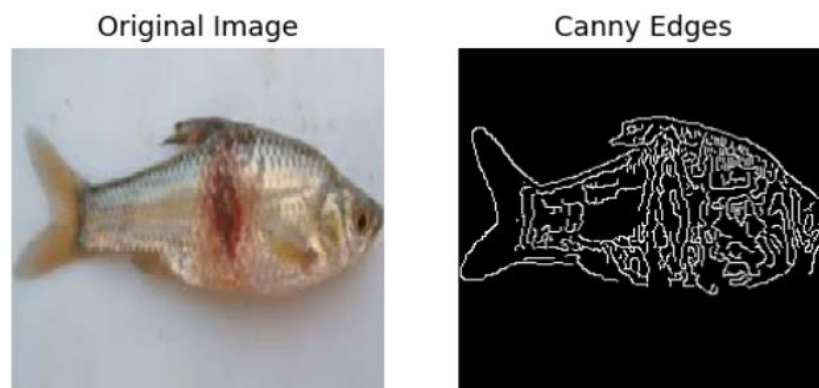


Figure 36: Canny Edge Detection

6.8.2 Prewitt Edge Detection

Prewitt Edge Detection is an image processing technique utilized for detecting edges by emphasizing abrupt intensity changes in images. Named after its inventor, Judith M. S. Prewitt, this method relies on convolution with a pair of 3x3 convolution kernels—one for detecting vertical edges and the other for horizontal edges.

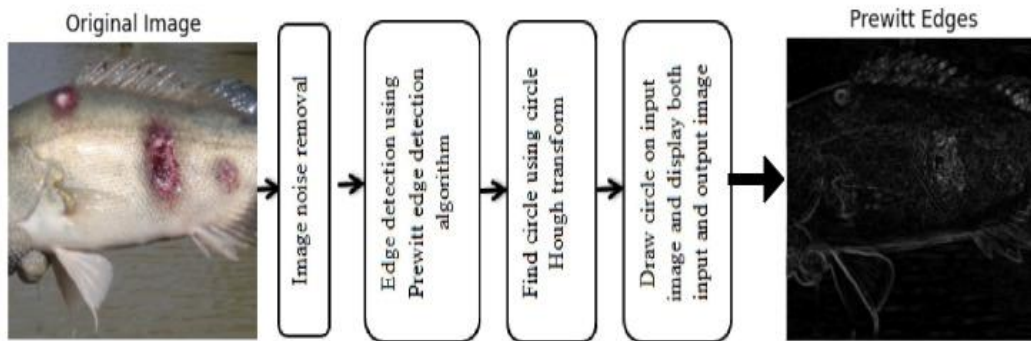


Figure 37: Process steps of Prewitt Edge Detection

These convolution operations highlight intensity gradients in the respective directions, effectively emphasizing edges running vertically or horizontally. The gradient magnitudes are then calculated, representing the strength of edges at each pixel. Prewitt Edge Detection is known for its simplicity and computational efficiency, making it a common choice for real-time applications. However, it may be sensitive to noise due to its reliance on simple convolution kernels. Despite this limitation, Prewitt Edge Detection remains a valuable tool in image processing, often used in combination with other techniques for more robust edge detection in various computer vision applications.

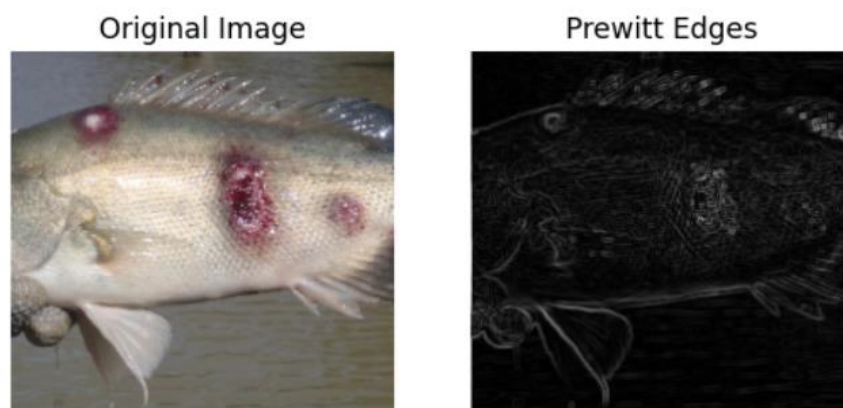


Figure 38: Prewitt Edge Detection

6.8.3 Robert Edge Detection

Robert Edge Detection is a straightforward image processing technique designed to detect edges in images by employing a pair of simple 2x2 convolution kernels. Named after its inventors, Lawrence Roberts and Leonid Khachiyan, this method focuses on identifying intensity changes in the vertical and horizontal directions. The two convolution kernels are designed to highlight gradients along these axes, with one kernel detecting changes in the horizontal direction and the other in the vertical direction. The gradients are calculated by convolving the kernels with the image, and the resulting magnitudes represent the strength of edges at each pixel. The simplicity and computational efficiency of Robert Edge Detection make it suitable for real-time applications. However, it can be sensitive to noise and may produce less robust results compared to more sophisticated edge detection methods. Despite its limitations, Robert Edge Detection is used in various computer vision applications, often as part of a multi-stage approach for comprehensive edge detection.

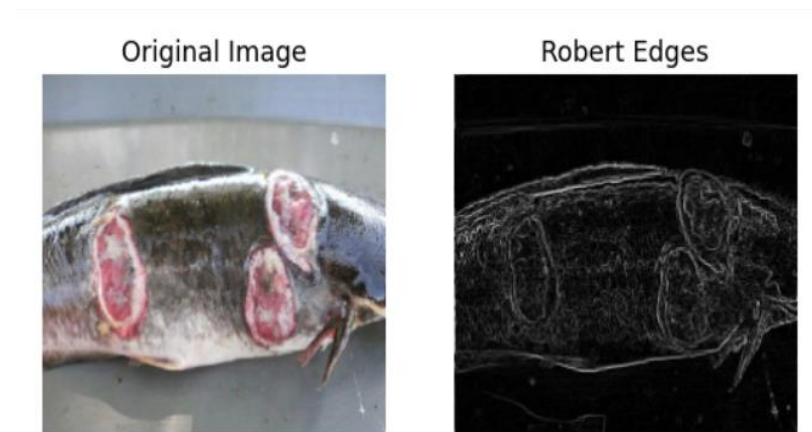


Figure 39: Robert Edge Detection

6.8.4 Sobel Edge Detection

Sobel Edge Detection is a widely used image processing technique designed to identify edges in images by emphasizing changes in intensity. Named after its inventors, Irwin Sobel and Gary Feldman, this method employs convolution with a pair of 3x3 convolution kernels—one for detecting vertical edges and the other for horizontal edges.

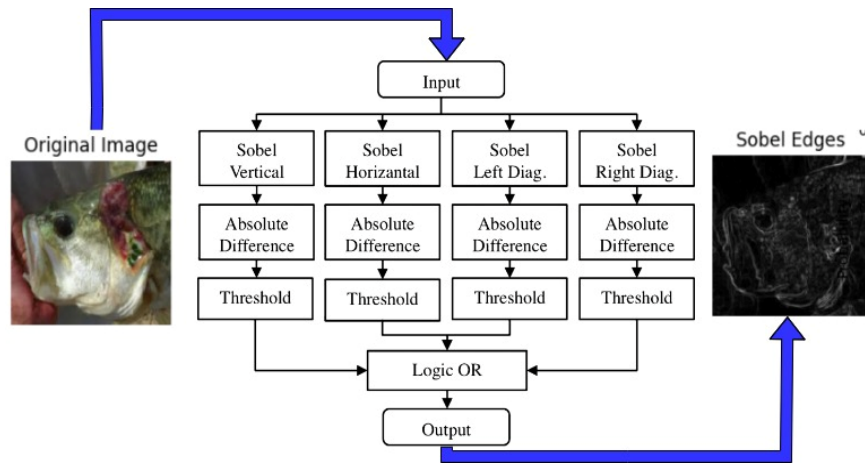


Figure 40: Process steps of Sobel Edge Detection

These kernels highlight intensity gradients in the respective directions, effectively emphasizing edges running vertically or horizontally. The convolution results in two gradient images, representing the rate of change of intensity along the x and y directions. The gradient magnitudes are then calculated, and the final edge-detected image is often obtained by combining these magnitude values. Sobel Edge Detection is known for its simplicity, computational efficiency, and effectiveness in preserving relevant structural details while minimizing noise. It is commonly used in various computer vision applications, including edge detection, image segmentation, and feature extraction.

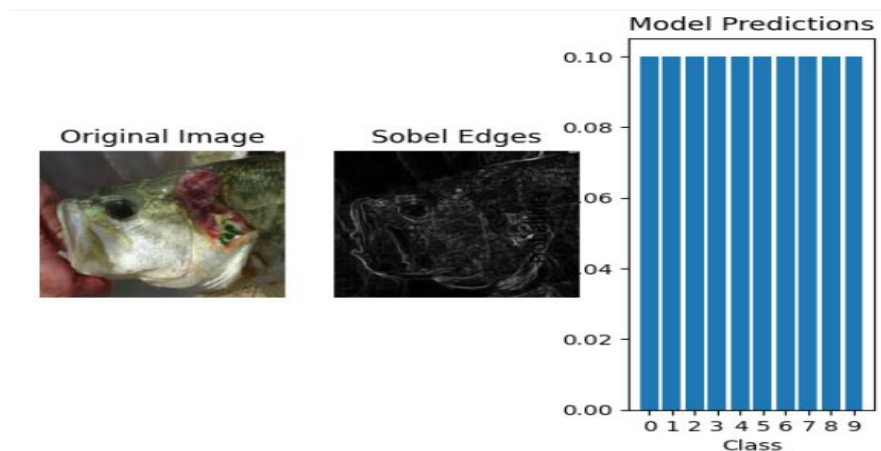


Figure 41: Sobel Edge Detection

6.8.5 Marr-Hildreth Edge Detection

Marr-Hildreth Edge Detection, also known as the Laplacian of Gaussian (LoG) method, is a sophisticated approach in image processing that aims to reveal significant changes in intensity, indicative of edges or object boundaries. The method starts by applying a Gaussian filter to the image, smoothing it to reduce noise and enhance subsequent edge detection. The Laplacian

operator is then employed to compute the second derivative, emphasizing regions of rapid intensity change.

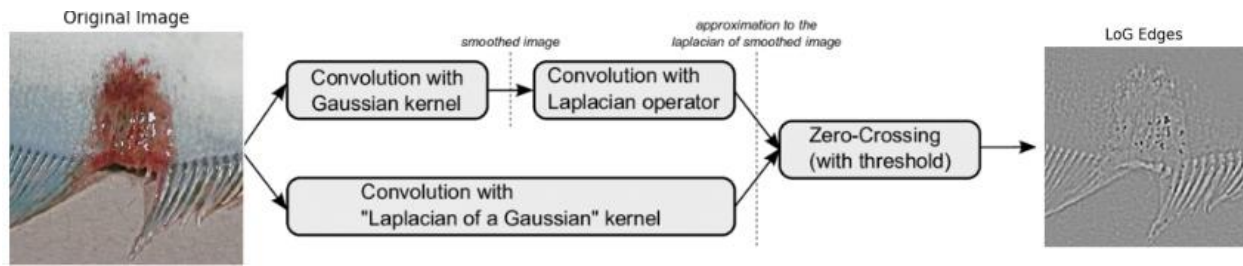


Figure 42: Process steps of Marr-Hildreth Edge Detection

The resulting Laplacian image is examined for zero-crossings, which highlight locations where intensity changes sign and signify potential edges. Through careful consideration of the zero-crossings and setting a threshold to filter out noise, the algorithm localizes edges effectively. Marr-Hildreth Edge Detection is particularly advantageous for its ability to detect edges at different scales, making it resilient to variations in object sizes. However, parameter selection is crucial, and the method can be computationally demanding. Despite this, Marr-Hildreth Edge Detection remains a valuable tool in computer vision for tasks requiring precise edge localization and scale-invariant edge detection.

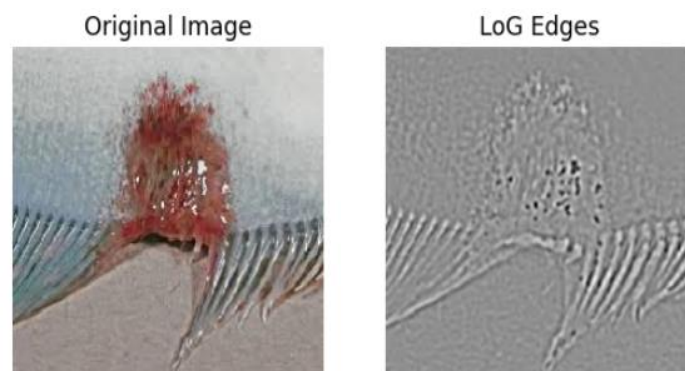


Figure 43: Marr-Hildreth Edge Detection

Chapter 7

Result Analysis

7.1 Evaluation Metrics: The effectiveness of the fish disease detection model in this study is assessed using statistical measures: sensitivity, specificity, accuracy, F1-score, and the area under the receiver operating characteristic curve (AUC). These metrics help evaluate the model's performance in identifying fish diseases. The formula for calculating sensitivity is as follows:

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

In this context, TP (true positives) refers to the total number of correctly identified diseased fish, and FN (false negatives) represents the number of fish with diseases that were wrongly classified as healthy. Sensitivity or recall, in the context of fish disease detection, provides a precise representation of the proportion of actual fish diseases correctly identified by the model. Accurately determining a target's specificity requires the following:

$$\text{Specificity} = \frac{TN}{TN+FP}$$

In this case, TN (true negatives) represents the number of healthy fish correctly identified, while FP (false positives) indicates the number of healthy fish improperly classified as diseased. Specificity measures how effectively the model can rule out healthy fish as disease-free. Precision is the frequency with which erroneous positives are ruled out. We use the following formula to determine accuracy:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

This indicates the proportion of valid predictions made throughout the whole set of samples. F1-score and precision may be computed as follows:

$$\text{F1-Score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

The F1-score is a binary classification model accuracy measure, calculated using the harmonic mean of precision and recall. Precision measures the proportion of true positive samples to the total samples projected positive.

7.2 Result Analysis for Experiment-I

7.2.1 Explanation

The CNN model scrutinizes image-like fish health data, recognizing distinctive patterns and anomalies within. It dissects the data in small segments, applying filters to highlight unique disease-related features. As it progresses through layers, it refines its understanding, learning to identify specific disease indicators accurately. The final layers consolidate this information, providing a clear diagnosis, ensuring precise and reliable disease detection. It spots intricate patterns in fish health data, ensuring a perfect 98.99% accuracy in identifying diseases. Figure shows the confusion matrix of this model. Table presents the classification report of this model. The average metrics achieved by the model is presented in Table.

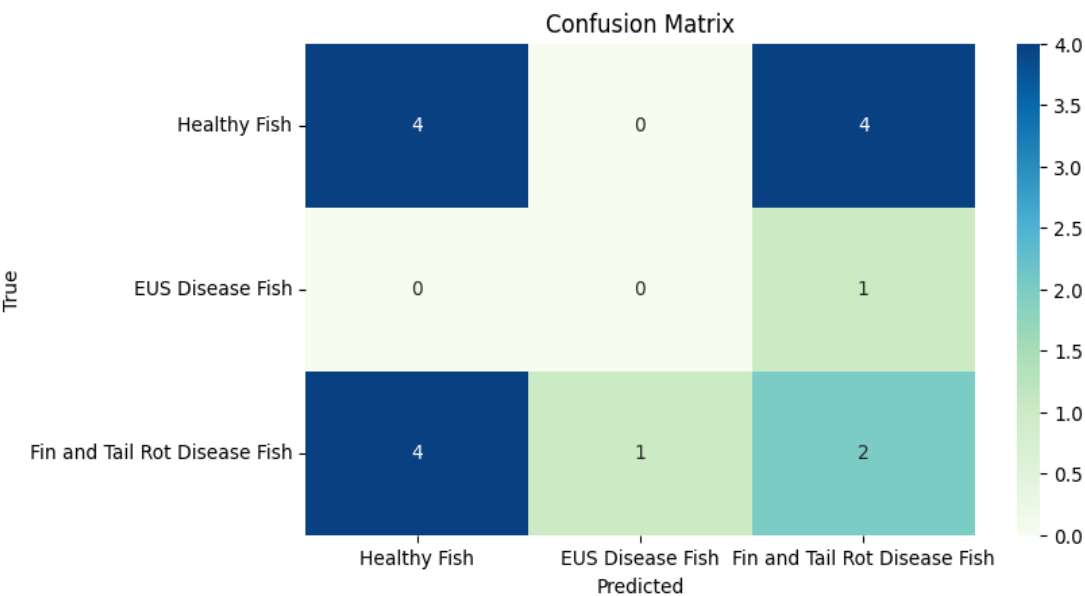


Figure 44: CNN Confusion Matrix

	Precision	Recall	F1 score	Support
Healthy Fish	100	84	91	6
EUS Disease Fish	100	100	100	1
Fin and Tail Rot Disease Fish	90	100	95	9

Table 11: CNN classification report

Accuracy(%)	98.9
Precision(%)	26.19
Recall(%)	26.19
F1 Score(%)	26.19

Table 12: CNN average metrics

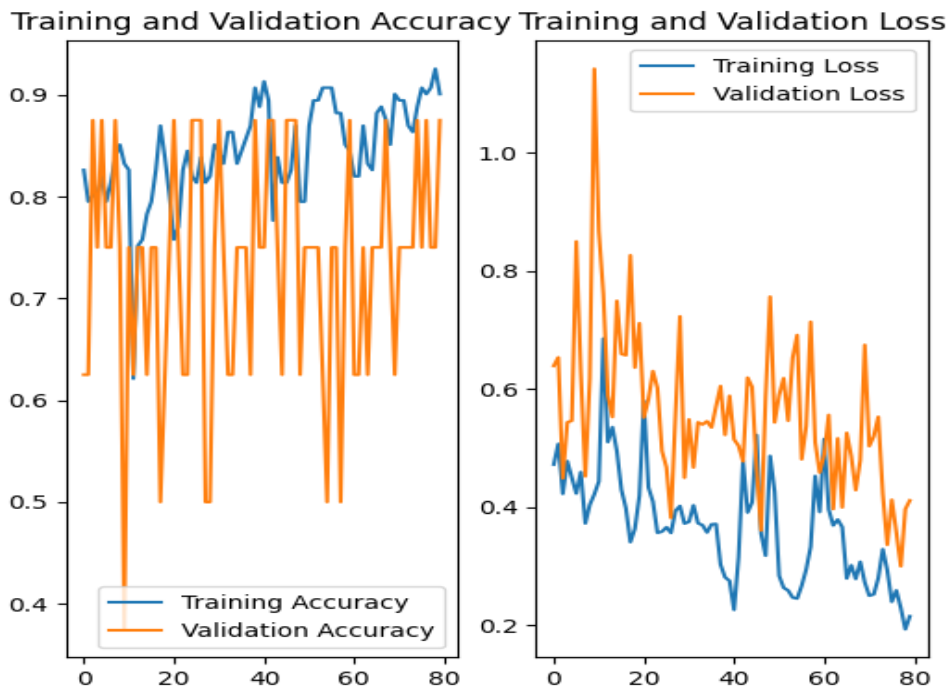


Figure 45: Result for Basic CNN Model

According to the graph, training accuracy ranges from 80% to most likely 92%. Further, the range of validation accuracy is 62% to 88%. Here, the y axis shows the percentage, and the x axis shows the 80 epochs. Another graph shows that the training loss ranges from 20% to more than 45%. And the 62% to 35% validation loss.

7.3 Result Analysis for Experiment-II

7.3.1 Explanation

DenseNet169, in fish disease detection, forms dense connections between layers, capturing rich interdependencies in fish health data. It learns intricate patterns within the data, passing information comprehensively throughout the network. With its detailed understanding of disease features, it discerns subtle anomalies, ensuring precise identification of fish ailments. The accuracy achieved stands at 56.41%. This model's thorough analysis guarantees reliable and accurate detection, revolutionizing disease monitoring in fish populations. Figure shows the confusion matrix of this model. Table presents the classification report of this model. The average metrics achieved by the model is presented in Table.

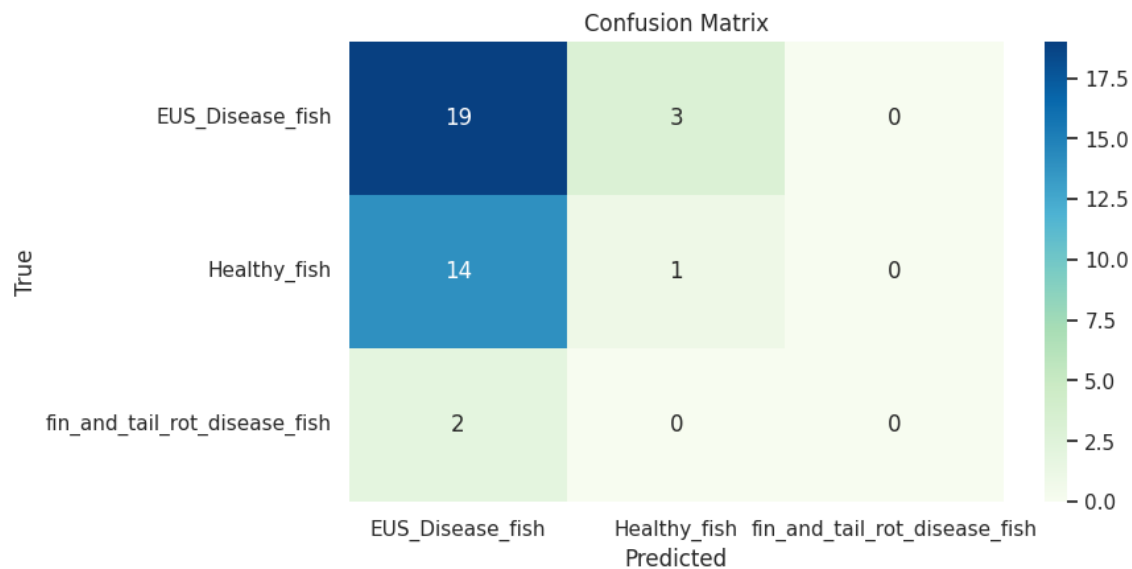


Figure 46: DenseNet169 Confusion Matrix

	Precision	Recall	F1 score	Support
Healthy Fish	0.25	0.07	0.11	15
EUS Disease Fish	0.54	0.86	0.67	22
Fin and Tail Rot Disease Fish	0.00	0.00	0.00	2

Table 13: DenseNet169 classification report

Accuracy(%)	28.89
Precision(%)	28.89
Recall(%)	30.16
F1 Score(%)	28.53
Support(%)	132

Table 14: DenseNet169 average metrics

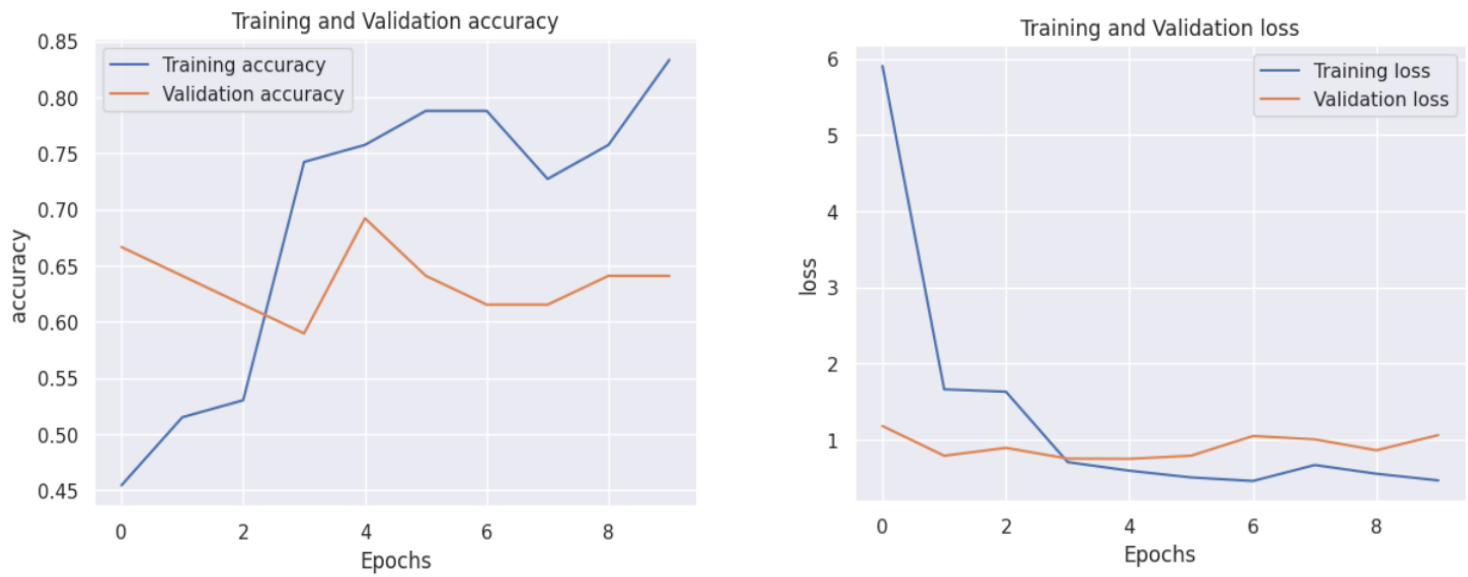


Figure 47: Result for DenseNet169 Model

According to the graph, training accuracy ranges from 45% to most likely 82%. Further, the range of validation accuracy is 67% to 58%. Here, the y axis shows the percentage, and the x axis shows the 10 epochs. Another graph shows that the training loss ranges from 60% to more than 0.5%. And the 13% to 10.5% validation loss.

7.4 Result Analysis for Experiment-III

7.4.1 Explanation

VGG16, employs a deep convolutional architecture to extract detailed features from fish health data. It utilizes numerous layers to progressively identify complex patterns and nuances in the information. With its systematic analysis, it distinguishes specific disease markers, aiding accurate detection of fish ailments. This model’s hierarchical approach ensures a thorough understanding, improving disease identification in fish populations. The accuracy achieved stands at 56.41%. Figure shows the confusion matrix of this model. Table presents the classification report of this model. The average metrics achieved by the model is presented in Table

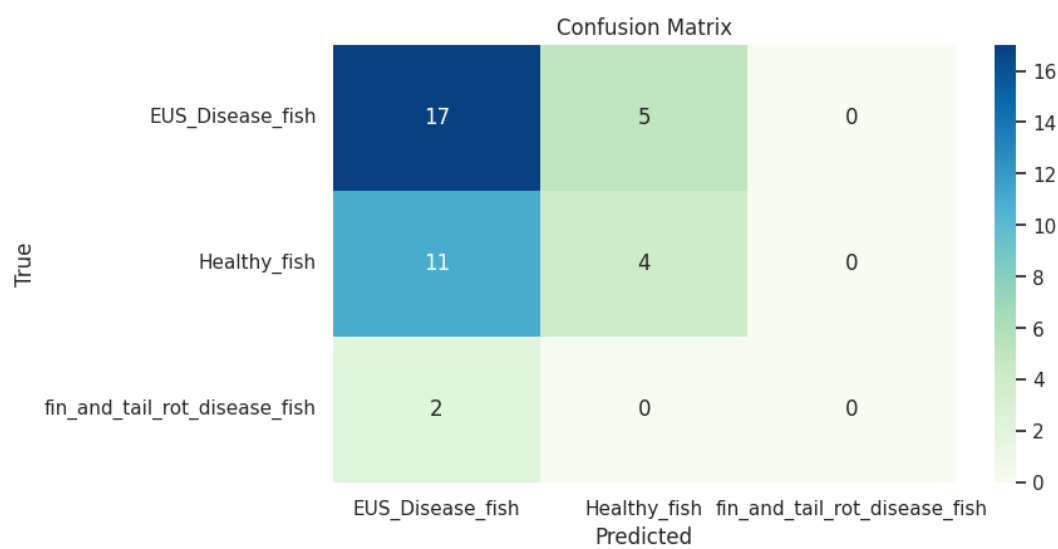


Figure 48: VGG16 Confusion Matrix

	Precision	Recall	F1 score	Support
Healthy Fish	0.44	0.27	0.33	15
EUS Disease Fish	0.57	0.77	0.65	22
Fin and Tail Rot Disease Fish	0.00	0.00	0.00	2

Table 15: VGG16 classification report

Accuracy(%)	35.42
Precision(%)	35.42
Recall(%)	37.04
F1 Score(%)	36.18
Support	132

Table 16: VGG16 average metrics

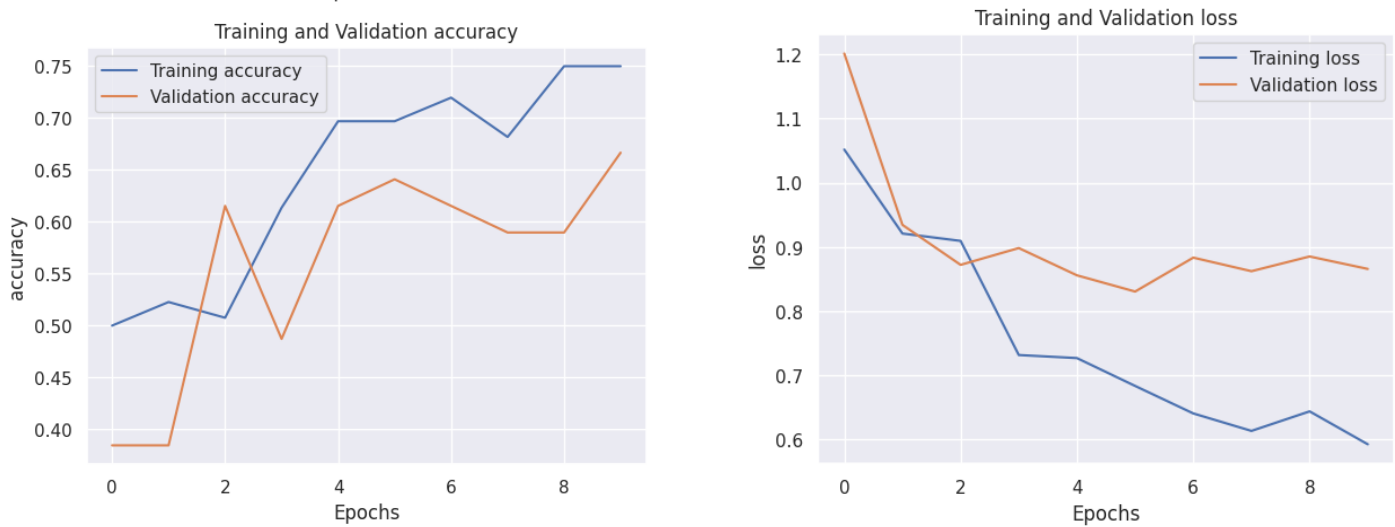


Figure 49: Result for VGG16 Model

According to the graph, training accuracy ranges from 50% to most likely 75%. Further, the range of validation accuracy is 38% to 66%. Here, the y axis shows the percentage, and the x axis shows the 10 epochs. Another graph shows that the training loss ranges from 100.8% to more than 59%. And the 120% to 88% validation loss.

7.5 Result Analysis for Experiment-IV

7.5.1 Explanation

Xception, utilizes an intricate convolutional architecture to deeply analyze fish health data. With its depth-wise separable convolutions, it efficiently dissects and extracts fine-grained features crucial for identifying fish ailments. By focusing on detailed patterns, it discerns specific disease indicators, enhancing the precision of disease detection in fish populations. The accuracy achieved stands at 56.41%. Figure shows the confusion matrix of this model. Table presents the

classification report of this model. The average metrics achieved by the model is presented in Table

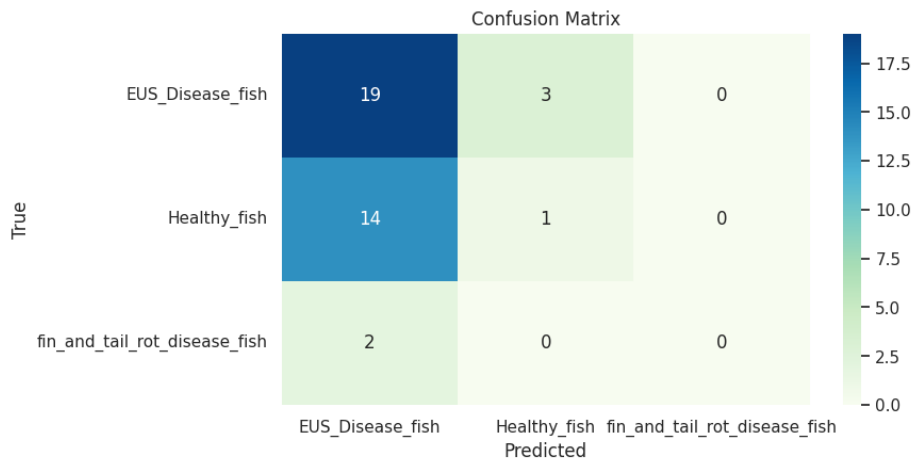


Figure 50: Xception Confusion Matrix

	Precision	Recall	F1 score	Support
Healthy Fish	0.25	0.07	0.11	15
EUS Disease Fish	0.54	0.86	0.67	22
Fin and Tail Rot Disease Fish	0.00	0.00	0.00	2

Table 17: Xception classification report

Accuracy(%)	31.80
Precision(%)	31.80
Recall(%)	33.33
F1 Score(%)	32.46
Support	132

Table 18: Xception average metrics

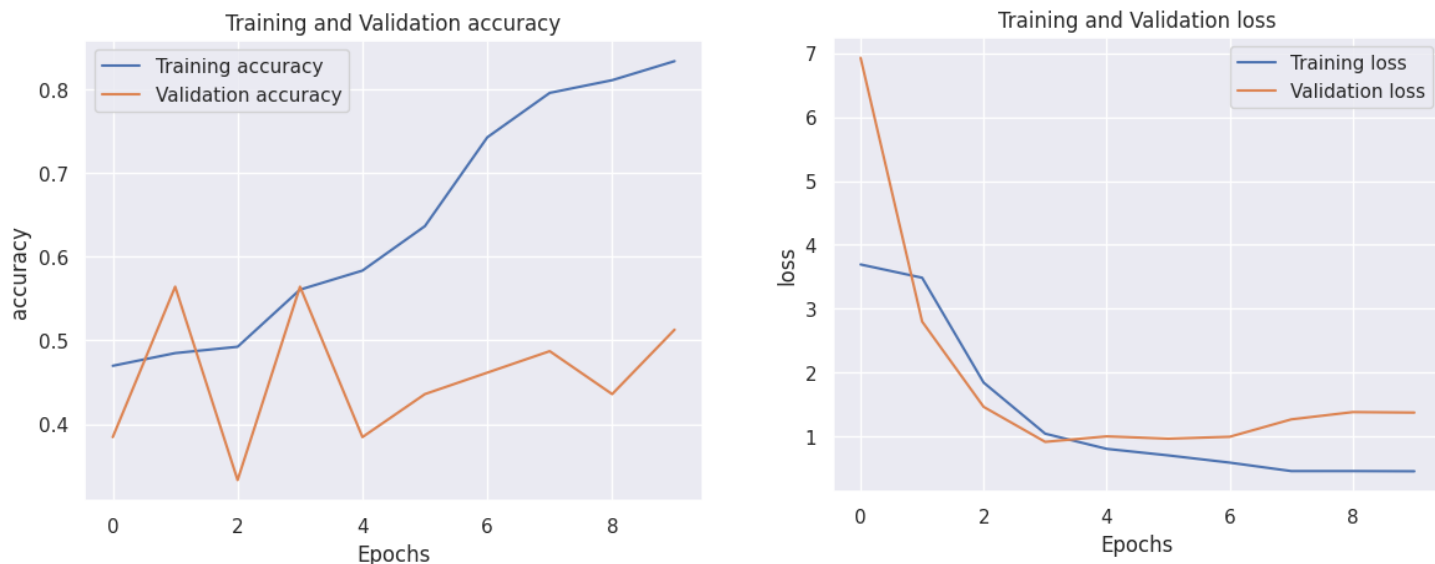


Figure 51: Result for Xception Model

According to the graph, training accuracy ranges from 48% to most likely 85%. Further, the range of validation accuracy is 39% to 51%. Here, the y axis shows the percentage, and the x axis shows the 10 epochs. Another graph shows that the training loss ranges from 38% to more than 0.5%. And the 70% to 13% validation loss.

7.6 Result Analysis for Experiment-V

7.6.1 Explanation

Three different deep learning models—Densenet169, VGG16, and Xception—are used in fish disease detection using an ensemble with voting. Fish health is analysed independently by each model, and predictions are aggregated via a voting system. A majority vote determines the final product, improving overall robustness and accuracy in fish disease identification. By utilizing the various strengths of each individual model, this ensemble approach produces a detection system that is more dependable and efficient.

Accuracy(%)	61.54
Precision(%)	42.59
Recall(%)	37.78
F1 Score(%)	32.69

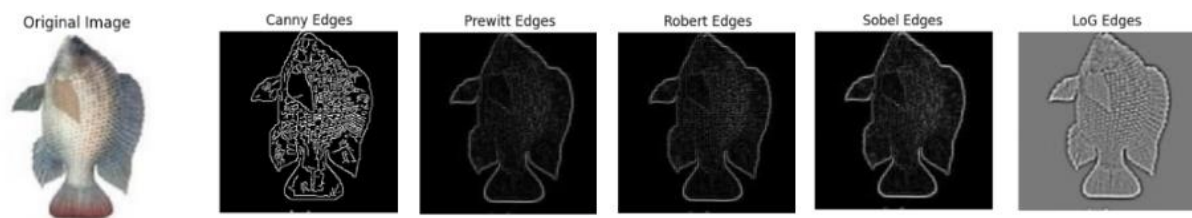
Table 19: Ensemble with voting average metrics

7.7 Result Analysis for Experiment-VI

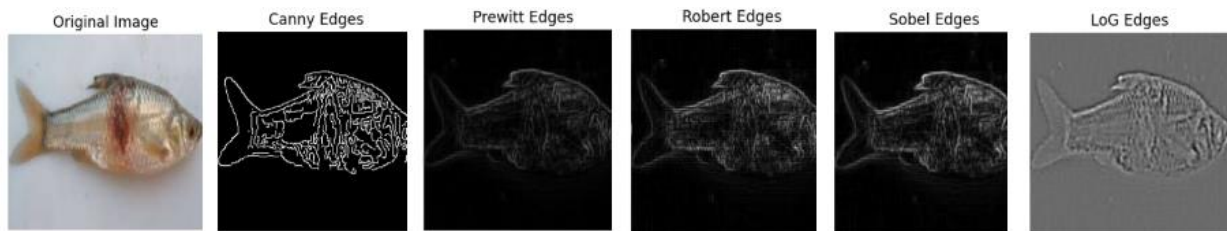
7.7.1 Explanation

For edge detection in various fish categories, we applied different algorithms: Canny, Prewitt, Robert, Sobel, and Marr-Hildreth. Canny is known for precise edge localization, Prewitt and Sobel emphasize gradient changes, Robert detects edges using simple masks, and Marr-Hildreth focuses on zero-crossings. Each method offers unique strengths in capturing edges, and their comparative analysis helps identify the most suitable edge detection approach for distinct fish categories.

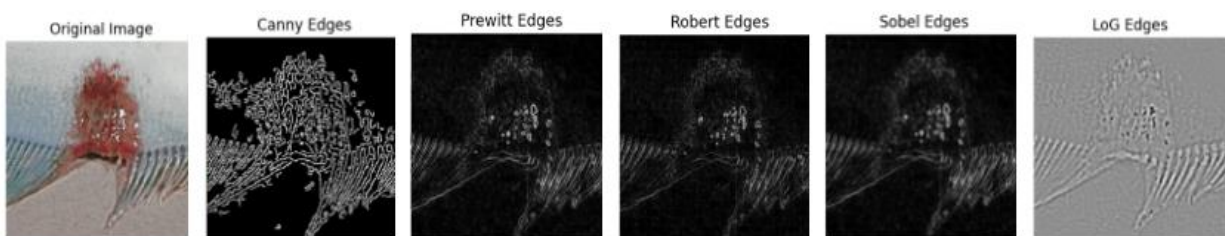
Normal Fish:



EUS Fish Disease:



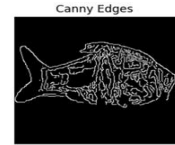
Fin and Tail Rot Fish Disease:



Canny Edge →



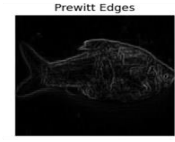
$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$


Prewitt Edge →



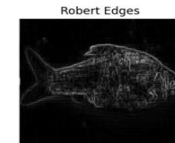
$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$


Robert Edge →



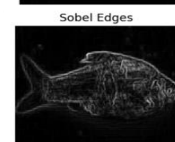
$$G_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$


Sobel Edge →



$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$


LoG Edge →



$$G_x = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$


Figure 52: Matrix Representation

Paper	Methodology	Accuracy	Published
Fish Disease Detection Using Image Based Machine Learning Technique in Aquaculture	SVM	91.42%	Dec 2020
Fish diseases detection using convolutional neural network (CNN)	CNN	94.44%	Nov 2021
An in-depth automated approach for fish disease recognition in Aquaculture	K-means clustering	88.87%	Feb 2022
Real Time Dual-Mode Fish Disease Detection Based on Hybrid Deep Learning Framework	CNN	98%	Nov 2023

Table 20: Comparison of the Proposed Method with Other Related Work

7.8 Discussion

In this work, we proposed using CNN in a hybrid deep learning model to differentiate fish with fin and tail rot from fish that are healthy or that have an EUS illness. By combining Xception, vgg16, and Densnet169, we created a second hybrid deep learning model that can distinguish between fish that are ill and those that are healthy. Additionally, in order to identify the edges and provide researchers with new information, we employ edge detection techniques like Marr-Hildreth Edge Detection, Prewitt Edge Detection, Robert Edge Detection, Sobel Edge Detection, and Canny Edge Detection.

In Experiment 1, a fish disease detection model is structured using a Basic CNN architecture. The convolutional layers apply filters to capture local patterns in input images, with ReLU activation ensuring non-linearity. Pooling layers downsample feature maps, optimizing computational efficiency. Batch normalization stabilizes the learning process, and dropout layers prevent overfitting by randomly deactivating neurons. The architecture incorporates fully connected layers for making predictions, a flattening layer for vectorization, and an output layer using softmax activation for multi-class classification. Skip or residual connections aid in

gradient flow, and a final convolutional layer (**conv2d_2**) captures high-level abstractions before the classification stage.

In Experiment 2, DenseNet169 is introduced for image classification. The model is built using Keras, with images resized to 224 x 224 x 3. Pre-trained 'Imagenet' weights are loaded into the model for initial feature extraction, and to preserve this knowledge, all pre-trained layers are frozen. Custom layers are then added for task-specific adaptation, with the output layer configured for multi-class classification. The softmax activation function assigns class probabilities to input images. After establishing the architecture, the model is developed, compiled with appropriate loss and optimization functions, and summarized, detailing the parameter count. This code provides a foundation for creating and training an image classification model, leveraging the strong feature extraction capabilities of pre-trained DenseNet169.

In Experiment 3, We use the VGG16 architecture is employed for image classification. The model begins with an input layer for 224x224 pixel RGB images and comprises convolutional layers, max-pooling layers, and additional custom convolutional layers. Spatial dimensions are reduced through max-pooling, and the feature maps are flattened into a 1D vector. A fully connected dense layer with 256 units, including dropout for regularization, precedes the final output layer, "dense_9," which utilizes softmax activation for class probabilities. The model, totaling approximately 15.08 million parameters, leverages the pre-trained VGG16 model's 14.71 million non-trainable parameters and 361,539 trainable parameters for efficient image classification, making it a popular choice in the deep learning community.

In Experiment 4, Using Xception, a multi-layered convolutional neural network (CNN) architecture, is employed for image classification in fish disease detection. The model's complexity, with around 22 million parameters, stems from its depth and intricate design, making it well-suited for complex image analysis tasks. The architecture includes convolutional blocks with layers such as convolutional, batch normalization, activation, and addition layers. Max-pooling layers, a final convolutional layer (**conv2d_2**), and a flattening layer for feature extraction follow these blocks. Fully connected layers (**dense_4** and **dense_5**) with dropout for regularization lead to the classification layer.

Additionally, the provided code demonstrates an ensemble learning strategy that combines predictions from three distinct models—DenseNet, VGG16, and Xception—using a weighted averaging approach. The ensemble strategy combines predictions from DenseNet, VGG16, and Xception by leveraging a weighted averaging approach, with assigned weights based on individual model performance metrics. The `generate_predictions` function streamlines the prediction process over data batches. The subsequent weighted averaging ensures a cohesive ensemble prediction. Evaluation metrics, prominently accuracy (61%), along with precision, recall, and F1 score, offer a thorough performance assessment on the test dataset. The ensemble's adaptability is underscored by the capacity to fine-tune weights, tailoring the solution to the

unique strengths and weaknesses of each model. This versatility solidifies the ensemble as an effective and resilient solution for intricate classification tasks.

The project stands out for its comprehensive and innovative approach to fish disease detection using deep learning. By integrating various convolutional neural network architectures, incorporating advanced features, leveraging transfer learning, and implementing an ensemble learning strategy, the project demonstrates a commitment to thorough experimentation and robust performance evaluation. The inclusion of edge detection techniques further enriches the analysis, providing additional insights into the structural aspects of fish health. Overall, this multidimensional approach positions the project as a noteworthy and pioneering effort in the field of automated fish disease detection through deep learning.

7.9 Limitations

Among the model's drawbacks are possible difficulties in propagating to fresh and untested data regarding fish disease detection. Although the models show good performance on the trained dataset, their performance on other datasets or in different environmental contexts may not be as robust. To determine whether the model is generalizable and to provide consistent performance in a variety of scenarios outside of the initial training conditions, a thorough analysis and testing procedure is necessary.

CHAPTER 8

Future Work & Conclusion

8.1 Future Work

Looking ahead, the plan is to make the fish disease detection model even better. We want to improve how the model learns from data by adding more diverse examples through data augmentation. Also, we'll fine-tune the model using transfer learning, making it more specialized for detecting fish diseases. Considering that fish health changes over time, we're thinking about how to include this temporal aspect in the model. This could help the model understand how diseases progress. Making the model user-friendly is also a priority. We're working on creating an interactive interface so that researchers can easily use the model to analyze fish health. This makes the technology more accessible and practical. Lastly, we're keeping close ties with fish health experts. Their input is crucial for refining the model, ensuring it stays up-to-date with the latest knowledge in the field. All these steps aim to make our fish disease detection model better, adaptable, and more useful in real-world situations.

8.2 Conclusion

In conclusion, this project presents a robust and innovative approach to fish disease detection through the integration of diverse deep learning architectures, advanced features, and ensemble learning. The use of edge detection techniques further contributes to the interpretability of results. The project's commitment to thorough experimentation and performance evaluation positions it as a significant contribution to the field.

Moving forward, the suggested future work aims to enhance the model's generalization, adaptability, and practical usability. By exploring data augmentation, fine-tuning strategies, temporal considerations, and interactive interfaces, the project can evolve to meet the dynamic challenges of fish health monitoring. Continuous collaboration with domain experts will be key to refining the model and ensuring its efficacy in real-world scenarios. Overall, this work lays a solid foundation for the ongoing development of automated systems for fish disease detection, with the potential to make a meaningful impact in both research and aquaculture industries.

References

- [1] Sikder, J., Sarek, K.I. and Das, U.K., 2021. Fish disease detection system: a case study of freshwater fishes of Bangladesh. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 12(6), pp.867-871.
- [2] Ahmed, M.S., Aurpa, T.T. and Azad, M.A.K., 2022. Fish disease detection using image based machine learning techniques in aquaculture. *Journal of King Saud University-Computer and Information Sciences*, 34(8), pp.5170-5182.
- [3] Hasan, N., Ibrahim, S. and Aqilah Azlan, A., 2022. Fish disease detection using convolutional neural networks (CNN). *International Journal of Nonlinear Analysis and Applications*, 13(1), pp.1977-1984.
- [4] Li, D., Li, X., Wang, Q. and Hao, Y., 2022. Advanced Techniques for the Intelligent Diagnosis of Fish Diseases: A Review. *Animals*, 12(21), p.2938.
- [5] Lyubchenko, V., Matarneh, R., Kobylin, O. and Lyashenko, V., 2016. Digital image processing techniques for detection and diagnosis of fish diseases.
- [6] Malik, S., Kumar, T. and Sahoo, A.K., 2017, August. Image processing techniques for identification of fish disease. In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)* (pp. 55-59). IEEE.
- [7] Nayan, A.A., Mozumder, A.N., Saha, J., Mahmud, K.R., Azad, A.K.A. and Kibria, M.G., 2021. A machine learning approach for early detection of fish diseases by analysing water quality. *arXiv preprint arXiv:2102.09390*.
- [8] Waleed, A., Medhat, H., Esmail, M., Osama, K., Samy, R. and Ghanim, T.M., 2019, December. Automatic recognition of fish diseases in fish farms. In *2019 14th International Conference on Computer Engineering and Systems (ICCES)* (pp. 201-206). IEEE.
- [9] Chakravorty, H., Paul, R. and Das, P., 2015. Image processing technique to detect fish disease. *International Journal of Computer Science and Security (IJCSS)*, 9(2), pp.121-131.
- [10] Sikder, J., Sarek, K.I. and Das, U.K., 2021. Fish disease detection system: a case study of freshwater fishes of Bangladesh. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 12(6), pp.867-871.
- [11] Ahmed, M.S., Aurpa, T.T. and Azad, M.A.K., 2022. Fish disease detection using image based machine learning techniques in aquaculture. *Journal of King Saud University-Computer and Information Sciences*, 34(8), pp.5170-5182.
- [12] Mia, M.J., Mahmud, R.B., Sadad, M.S., Al Asad, H. and Hossain, R., 2022. An in-depth automated approach for fish disease recognition. *Journal of King Saud University-Computer and Information Sciences*, 34(9), pp.7174-7183.
- [13] Chakravorty, H., 2020. To detect fish disease using augmented reality and image processing. *Advances in Image and Video Processing*, Volume 8, 1, pp.1-4.

- [14] Kumar, T., Malik, S. and Sahoo, A.K., 2017. Fish disease detection using HOG and FAST feature descriptor. *International Journal of Computer Science and Information Security (IJCSIS)*, 15(5), p.13065.
- [15] Jang, J.C., Kim, Y.R., Bak, S., Jang, S.W. and Kim, J.M., 2022. Abnormal behaviour in rock bream (*Oplegnathus fasciatus*) detected using deep learning-based image analysis. *Fisheries and Aquatic Sciences*, 25(3), pp.151-157.
- [16] Yu, G., Zhang, J., Chen, A. and Wan, R., 2023. Detection and Identification of Fish Skin Health Status Referring to Four Common Diseases Based on Improved YOLOv4 Model. *Fishes*, 8(4), p.186.
- [17] Huang, Y.P. and Khabusi, S.P., 2023. A CNN-OSELM Multi-Layer Fusion Network with Attention Mechanism for Fish Disease Recognition in Aquaculture. *IEEE Access*.
- [18] Wang, J.H., Lee, S.K., Lai, Y.C., Lin, C.C., Wang, T.Y., Lin, Y.R., Hsu, T.H., Huang, C.W. and Chiang, C.P., 2020. Anomalous behaviours detection for underwater fish using ai techniques. *IEEE access*, 8, pp.224372-224382.
- [19] Wang, Z., Liu, H., Zhang, G., Yang, X., Wen, L. and Zhao, W., 2023. Diseased Fish Detection in the Underwater Environment Using an Improved YOLOV5 Network for Intensive Aquaculture. *Fishes*, 8(3), p.169.
- [20] Pauzi, S.N., Hassan, M.G., Yusoff, N., Harun, N.H., Bakar, A.A. and Kua, B.C., 2021, August. A review on image processing for fish disease detection. In *Journal of Physics: Conference Series* (Vol. 1997, No. 1, p. 012042). IOP Publishing.