

**FAKE HILSA DETECTION USING MACHINE LEARNING ALGORITHM**

**BY**

**MERAJUL ISLAM**

**ID: 172-15-9627**

**AND**

**ABDUR RAHMAN**

**ID: 172-15-10022**

**AND**

**JANNATUL FERDOUS ANI**

**ID: 173-15-10483**

This Report Presented in Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

**Ms. Zakia Zaman**

Senior Lecturer

Department of CSE

Daffodil International University

Co-Supervised By

**Mr. Ahmed Al Marouf**

Senior Lecturer

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

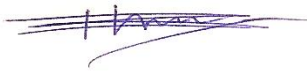
**DHAKA, BANGLADESH**

**MAY 2021**

## **APPROVAL**

This Project titled “**Fake Hilsa Detection Using Machine Learning Algorithm**”, submitted by Merajul Islam, ID No: 172-15-9627, Abdur Rahman, ID No: 172-15-10022 and Jannatul Ferdous Ani, ID No: 173-15-10483 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 01 June 2021.

## **BOARD OF EXAMINERS**



---

**Dr. Touhid Bhuiyan**  
**Professor and Head**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Chairman**



---

**Abdus Sattar**  
**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**



---

**Md. Jueal Mia**  
**Senior Lecturer**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**



**External Examiner**

---

**Dr. Dewan Md. Farid**

**Associate Professor**

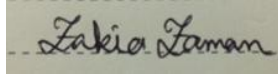
Department of Computer Science and Engineering  
United International University

## DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ms. Zakia Zaman, Senior Lecturer, Department of CSE** Daffodil International University.

We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**



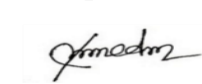
**Zakia Zaman**

**Senior Lecturer**

Department of CSE

Daffodil International University

**Co-Supervised by:**



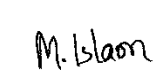
**Ahmed Al Marouf**

**Senior Lecturer**

Department of CSE

Daffodil International University

**Submitted by:**

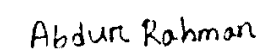


**Merajul Islam**

ID: 172-15-9627

Department of CSE

Daffodil International University

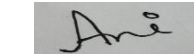


**Abdur Rahman**

ID: 172-15-10022

Department of CSE

Daffodil International University



**Jannatul Ferdous Ani**

ID: 173-15-10483

Department of CSE

Daffodil International University

## ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Ms. Zakia Zaman, Senior Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of *Machine Learning* to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan** Professor and Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

## **ABSTRACT**

Hilsa is the national fish of Bangladesh. Bangladesh is earning a lot of foreign currency by exporting this fish. Unfortunately, in recent days some unscrupulous businessmen are selling fake Hilsa fishes to gain profit. The Sardines and Sardinella are the most sold in the market as Hilsa. The government agency of Bangladesh, namely the Bangladesh Food Safety Authority said that these fake Hilsa fish contain high levels of cadmium and lead which are detrimental for humans. In this research, We have proposed a method that can readily identify original Hilsa fish and fake Hilsa fish. Based on the research available on online literature, we are the first to do research on identifying original Hilsa fish. We have collected more than 16,000 images of original and counterfeit Hilsa fish. To classify these images, we have used several deep learning based models. Then the performance has been compared between them. Among those models, DenseNet201 achieved the highest accuracy of 97.02%.

# TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Board of Examiners	i-ii
Declaration	iii
Acknowledgments	iv
Abstract	v
List of Figures	ix-x
List of Tables	xi
<b>CHAPTER</b>	
<b>CHAPTER 1: INTRODUCTION</b>	<b>1-4</b>
1.1 Introduction	1
1.2 Motivation	1-2
1.3 Rationale of the Study	2
1.4 Research Question	2-3
1.5 Expected Output	3
1.6 Project Management and Finance	3
1.7 Report Layout	3-4
<b>CHAPTER 2: BACKGROUND</b>	<b>5-7</b>
2.1 Preliminaries	5
2.2 Related Works	5-6
2.3 Comparative Analysis and Summary	6
2.4 Scope of the Problem	6-7

2.5 Challenges	7
<b>CHAPTER 3: PROPOSED METHODOLOGY</b>	<b>8-21</b>
3.1 Research Subject and Instrumentation	8
3.2 Work Flow	8-9
3.3 Properties of Dataset	9
3.4 Data Preprocessing	9-10
3.5 Model Generation	11-19
3.5.1 Xception	12
3.5.2 VGG16	13-14
3.5.3 InceptionV3	14-16
3.5.4 DenseNet201	16-17
3.5.5 NASNetMobile	18-19
3.6 Features Extraction	19-20
3.7 Parameter Setting	20-21
<b>CHAPTER 4: EXPERIMENTAL RESULT ANALYSIS</b>	<b>22-29</b>
4.1 Introduction	22
4.2 Confusion Matrix and Evaluation Metrics	22-23
4.3 Experimental Results & Discussion	23-38
4.3.1 Experimental Result Analysis for NASNetMobile	23-26
4.3.2 Experimental Result Analysis for VGG16	26-29
4.3.3 Experimental Result Analysis for Xception	29-32
4.3.4 Experimental Result Analysis for InceptionV3	32-35



4.3.5 Experimental Result Analysis for DenseNet201	35-38
4.4 Comparative Analysis	38-39
<b>CHAPTER 5: IMPACT ON SOCIETY, ENVIRONMENT, AND SUSTAINABILITY</b>	<b>40</b>
5.1 Impact on Society	40
5.2 Impact on Environment	40
5.3 Ethical Aspects	40
5.4 Sustainability Plan	40
<b>CHAPTER 6: CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH</b>	<b>41</b>
6.1 Summary of the Study	41
6.2 Conclusions	41
6.3 Implication for Further Study	41
<b>REFERENCES</b>	<b>42-43</b>
<b>APPENDIX</b>	<b>44</b>

## LIST OF FIGURES

<b>FIGURES</b>	<b>PAGE NO</b>
Figure 3.1: Proposed classification procedure	9
Figure 3.2: Images of two classes a Hilsa fish, b fake Hilsa fish	10
Figure 3.3: Image dataset is divided into training and validation sets	10
Figure 3.4: Basic Structure of CNN	11
Figure 3.5: Visualize the features extraction from the first two conv2d (DenseNet201)	20
Figure 4.1: Each Epochs Accuracy and Loss for NASNetMobile	23
Figure 4.2: Training Accuracy and Loss Vs Validation Accuracy and Loss for NASNetMobile	24
Figure 4.3: Confusion Matrix for NASNetMobile	24
Figure 4.4: Classification Report for NASNetMobile	25
Figure 4.5: ROC/AUC for NASNetMobile	25
Figure 4.6: Sensitivity and Specificity for NASNetMobile	26
Figure 4.7: Each Epochs Accuracy and Loss for VGG16	26
Figure 4.8: Training Accuracy and Loss Vs Validation Accuracy and Loss for VGG16	27
Figure 4.9: Confusion Matrix for VGG16	27
Figure 4.10: Classification Report for VGG16	28
Figure 4.11: ROC/AUC for VGG16	28
Figure 4.12: Sensitivity and Specificity for VGG16	29
Figure 4.13: Each Epochs Accuracy and Loss for Xception	29
Figure 4.14: Training Accuracy and Loss Vs Validation Accuracy and Loss for Xception	30
Figure 4.15: Confusion Matrix for Xception	30
Figure 4.16: Classification Report for Xception	31
Figure 4.17: ROC/AUC for Xception	31

Figure 4.18: Sensitivity and Specificity for Xception	32
Figure 4.19: Each Epochs Accuracy and Loss for InceptionV3	32
Figure 4.20: Training Accuracy and Loss Vs Validation Accuracy and Loss for InceptionV3	33
Figure 4.21: Confusion Matrix for InceptionV3	33
Figure 4.22: Classification Report for InceptionV3	34
Figure 4.23: ROC/AUC for InceptionV3	34
Figure 4.24: Sensitivity and Specificity for InceptionV3	35
Figure 4.25: Each Epochs Accuracy and Loss for DenseNet201	35
Figure 4.26: Training Accuracy and Loss Vs Validation Accuracy and Loss for DenseNet201	36
Figure 4.27: Confusion Matrix for DenseNet201	36
Figure 4.28: Classification Report for DenseNet201	37
Figure 4.29: ROC/AUC for DenseNet201	37
Figure 4.30: Sensitivity and Specificity for DenseNet201	38

## LIST OF TABLES

<b>TABLES</b>	<b>PAGE NO</b>
TABLE 3.1: DEVELOPMENT ENVIRONMENT FOR THE PROPOSED FAKE HILSA DETECTION USING MACHINE LEARNING ALGORITHM	8
TABLE 3.2: MODEL CHARACTERISTICS IN DETAILS	12
TABLE 3.3: MODEL SUMMARY OF XCEPTION	12
TABLE 3.4: MODEL SUMMARY OF VGG16	13-14
TABLE 3.5: MODEL SUMMARY OF INCEPTIONV3	14-16
TABLE 3.6: MODEL SUMMARY OF DENSENET201	16-17
TABLE 3.7: MODEL SUMMARY OF NASNETMOBILE	18-19
TABLE 3.8: PARAMETERS AND FUNCTIONS FOR TRAINING PROCEDURE	21
TABLE 4.1: CONFUSION MATRIX RESULT FOR FIVE CNN MODELS	39
TABLE 4.2: CLASSIFICATION REPORT FOR FIVE CNN MODELS.	39

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Introduction**

Hilsa fish is known as one of the tastiest and favorite fish in the world. It is named as Ilish in Bangladesh. Every year Bangladesh produces 75% of the total Hilsa fish production in the world [1]. And the other 25% produced in India, Myanmar, Pakistan, and some countries along the Arabian sea. Three kinds of Hilsa fishes are available in Bangladesh. These are Tenualosa Ilisha which is called Padma Ilish and it is the most popular Hilsa fish in the world. Another is Nenuacosa Toli, which is called Chandana Ilish. And the last one is Hilsha Kelle that is known as Gurta Ilish. Hilsa is known as saltwater fish, but it lays eggs in freshwater rivers (Padma, Meghna, Jamuna) delta at the Bay of Bengal. The demand for Hilsa fish in Asian countries exists throughout the year. However, it increases a lot especially in festivals like Pohela Boishakh (Bengali new year), Saraswati Puja, etc. In many countries, the production of Hilsa fish is now declining. But its production in Bangladesh is increasing every year. By exporting these fishes in many countries of the world, a large amount of foreign currency is earned every year. It adds 1% of the total GDP in Bangladesh [1].

### **1.2 Motivation**

For several years, in many countries, including Bangladesh, India, some unscrupulous traders are selling fake Hilsa fish. With a large portion of original Hilsa fish, they added part of fake Hilsa fish inside it then exported to many countries. This tarnishes the image of our country and in return lowering the revenues. These fake Hilsa look a lot like the original one, but there are some differences in shape, mostly in the head and tail, and also in the taste and smell. The Sardine, Sardinella, Cokkash, Chapila, Indian oil sardine, and all of these look like original Hilsa fish. The body of the Hilsa fish is equally long toward the abdomen and back, but the abdomen is higher than the back in Sardine. The head of

©Daffodil International University

the Hilsa fish is slightly longer, on the other hand, the head of the sardine fish is slightly smaller and the front side is slightly blunt. Fake Hilsa's eyes are bigger than the real Hilsa fish. The back of the original Hilsa is bluish green. Sardine has black dots at the base of the dorsal fin, but Hilsa fish has black dots on its gill. The front of the sardine's back fins and the edges of the tail fins are blurred but the hind fins of Hilsa are whitey. But the sad truth is, fake Hilsa cannot be easily detected by looking at it many times. To solve this problem, we use machine vision-based deep learning models to identify the original fish.

### 1.3 Rationale of the Study

The hilsa fish has been designated as a unique Bangladeshi product and has been identified with a geographical indication (GI) tag the World Intellectual Property Organization (WIPO) states that. People are now suffering from the bad impact of eating fake Hilsa fish. And its busing and selling are spreading around very fast. People are being deceived into buying these fake fish at a lot of prices. A syndicate of corrupt traders has always been active. They are sending a lot of fish internationally, which is damaging Bangladesh's reputation in the eyes of the rest of the world. Fake Hilsa sales can be stopped by using it in the important sector through our research. And people can easily identify these fake Hilsa.

The most significant factor that inspired us is that there is no research has been completed based on identifying fake Hilsa fish. For this purpose, we chose it. We hope it will be research that is unique from others.

### 1.4 Research Question

- ✓ What are the challenges in classifying real hilsa?
- ✓ Which model is better for predict real hilsa?
- ✓ What are the areas of research in identifying the real hilsa, where I can explore my research topics?
- ✓ How is this project performing?

- ✓ How did collect fake hilsa data?

## 1.5 Expected Output

We determined our scopes of work based on current study and expertise. We use machine learning to present our research. We used a total of five models to distinguish the fake and real hilsa in this research. Completing the process, it possible to classify real hilsa in a more precise and precise manner. There is a lot of space for exploration in the Machine learning field from a research standpoint. Machine learning is the best side to the research field according to Bangladesh's perspective. To solve this challenge, we used a variety of machine learning models. We were able to predict a successful solution to this problem using the models.

## 1.6 Project Management and Finance

We didn't need to buy any software because the component we used to implement this research is open source. For training and testing the models, we used Google Colab. It provides us free GPUs and Opus to run our code on it. For the implementation of this research study, we just have to take the cost of the internet bill.

## 1.7 Report Layout

There is a total of six chapters in this report. The first chapter provides an outline of the entire research study. We've covered the following topics in this chapter: Introduction, Motivation, Rationale of the Study, Research Questions, Expected Output, Project Management and Finance, and Report Layout. The background of our research was addressed in chapter two. We also provide information on some Related Work, as well as Comparative Analysis and Summary, a Research Overview, the Scope of the issue, and the Research Challenges. We covered the entire research methodology in chapter three with subsections. Introduction, Work Flow, Properties of Dataset, Data Preprocessing, Model

©Daffodil International University

Generation, Features Extraction, and Parameter Setting. In chapter four, we covered our experimental results and a discussion about our research study. Chapter five discussed about Impact on Society, Impact on Environment, Ethical Aspects, and Sustainability Plan. In chapter six we have discussed the Summary of our Research Study, Conclusions, and Implication for Further Study. In the end, there have references sections that helped us in our study.



## **CHAPTER 2**

### **BACKGROUND**

#### **2.1 Preliminaries**

Deep learning is a branch of machine learning that uses deep neural network models to infer critical hypotheses from input data implicitly. Deep learning is usually used for unsupervised or semi supervised learning. In the modern world, deep learning is used in all contexts. Deep neural networks play a vital role in Fish image analysis in classification tasks. The image classification receives input images and generates output categorization to determine whether or not the fish is original. We have worked using these concepts in our research study. However, in our study, we utilized real-life problems to develop solutions.

#### **2.2 Related Work**

We studied some notable research work on fish classification and also on image processing. Pavla et al. [2] used a modified Rosenblatt algorithm to classify six species of fish. They have used 2132 silhouette images. Of these, 1406 images were used for model training, and 726 used for model testing. Out of 726 images, their modified model was able to classify 386 images correctly. Israt et al. [3] used SVM, KNN, and ensemble-based algorithms to recognize six types of local fish. They used a histogram-based method to segment the gray-scale fish images. In their research, SVM achieved the highest accuracy of 94.2%. In the paper [4], the authors used pre-trained VGG16 for feature extraction of eight species of fish images and logistic regression has been used for the classification. Then, they achieved 93.8% accuracy. Rowell et al. [5] used the InceptionV3 model to classify Nile tilapia is harvested alive or Hibay. They used Adam optimizer in the last two layers of the model for increasing the accuracy rate. After 1000 iterations, their model achieved the best accuracy. Hafiz et al. [6] used 915 fish images from six different classes. Six different types of CNN model have been used in this research, namely VGG-16 for transfer learning, one-block VGG, two-block VGG, three-block VGG, LeNet 5, AlexNet,

GoogleNet, ResNet 50, and a proposed 32 Layer CNN with 404.4 million parameters, which gives the best accuracy than the rest. Xiang et al. [7] used transferred DenseNet201 for the diagnosis of breast abnormality. Here, 114 abnormal tissue images have been used for classification. Then, their model gave 92.73% accuracy. Jing et al. [8] used a multiclass support vector machine (MSVM) algorithm to classify six species of freshwater fish. They have captured all the images in  $1024 \times 768$  sizes. Then, multiple features have been extracted from these images. Ogunlan et al. [9] used a support vector machine (SVM)-based technique for classifying 150 fish. Of these, 76 have been used for model training and 74 for testing the model. They have used three more models in their research. Those are artificial neural networks (ANNs), K-means clustering, and K-nearest neighbors (KNN). And principal component analysis (PCA) is used to reduce the dimensionality of a dataset. SVM achieved the highest accuracy of 74.32%. Mohamad et al. [10] used the SVM, KNN, and ANN models to classify Nile tilapia fish. They used the scale invariant feature transform (SIFT) and speeded up robust features (SURF) algorithms for feature extraction. SURF achieved the highest accuracy of 94.44%.

### 2.3 Comparative Analysis and Summary

We studied a variety of journal papers, articles, books, and conference papers in order to conduct our research. We got the highest accuracy using the DenseNet201 model. On a variety of datasets, several researchers used the DenseNet201 model. In our study, we used Deep learning, specifically convolutional neural networks (CNN), to develop a method for the detection of fake Hilsa fish. So far there has been a lot of work on fish but no work has been done on Hilsa fish in the machine learning sector. We were the first to do work on it.

### 2.4 Scope of the Problem

During our research, we discovered that selecting an actual research area from a vast sea of fields of knowledge is very difficult for an anonymous source. There are many areas of expertise in which a researcher would like to conduct research. But there are various problems to be faced while doing research. We also faced various problems while doing

our research. In this research, there has not much similar work. By solving all kinds of problems, we have created a process based on machine learning that will classify the real Hilsa. This is because the seller does not face any further inconvenience.

## 2.5 Challenges

When it comes to publishing a research paper or thesis, there are many difficulties and roadblocks to overcome. Our thesis faces a number of difficulties. The dataset is the most difficult aspect of our study. For our dataset, we collect real hilsa images from different places in Bangladesh. To gather data on fake Hilsa, we had to overcome several challenges. There are several different types of fake hilsa fish, making it difficult to collect images of fake Hilsa fish. We performed binary classification due to fake Hilsa fish's data shortage. If we did multiclass classification, the dataset would not be balanced and the model would be overfitting. We put all types of fake Hilsa fish into one class because we couldn't gather the same amount of data for each kind of fake Hilsa fish. However, we were able to collect more than a total of 16,000 image data. Another challenge is that we'd like to expand our data collection. The second challenge that we have faced is to test the model train with this large amount of data. We used Google Colab to use require GPUs for train and testing each CNN model. Each of the CNN models took more than 4 hours for training and testing. During this time, if for any reason the Internet connection is interrupted, it has to be restarted from the beginning. Due to this, it took a long time in our research study.

## CHAPTER 3

### PROPOSED METHODOLOGY

#### 3.1 Research Subject and Instrumentation

The name of our research subject is “Fake Hilsa Detection Using Machine Learning Algorithm”. This chapter mainly covers the project's development phase, including the detailed conceptual and theoretical process.

The platform for training and testing the models-

TABLE 3.1: DEVELOPMENT ENVIRONMENT FOR THE PROPOSED FAKE HILSA DETECTION USING MACHINE LEARNING ALGORITHM.

Platform	Google Colab
Language	Python
Backend	TF version: 2.3.0 TF hub version 0.8.0
GPUs	12GB NVIDIA Tesla K80
Browser	Google Chrome
Library	Numpy, Matplotlib, Keras

#### 3.2 Work Flow

Figure 3.1 shows the working strides of our study. In brief, after assembling the datasets, it is labeled, pre-processed, and divided into the train and validation set. Using the datasets, five different CNN models are trained and validated to evaluate model performance. Their performance has been visualized through the confusion matrix and several evaluation metrics such as accuracy, precision, recall, and F1 score are used to compare among different models.

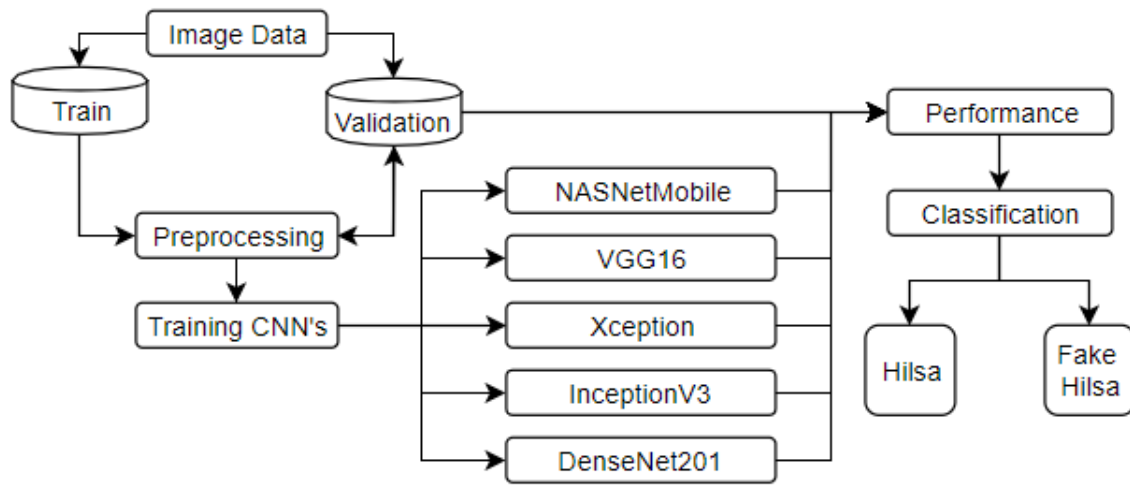


Figure 3.1: Proposed classification procedure

### 3.3 Properties of Dataset

There have been 16,622 different sizes of images that have been categorized into two groups named real Hilsa and fake Hilsa. Those different sizes of images are taken from the endemic fish market and many alternative sources. There are three different types of Hilsa fish in the 8722 images: *Tenuialosa ilisha* (Padma Ilish), *Nenuacosa toli* (Chandana Ilish), and *Hilsha kelle* (Gurta Ilish). The other 7900 images are of different types of fake Hilsa fish. Mostly Sardine, *Sardinella*, Indian oil sardine, Cokkash, Chapila, and few other fishes that look similar to Hilsa.

### 3.4 Data Preprocessing

Data Pre-processing is a standard method of preparing data for model implementation. Pre-processing requires putting raw images and using various methods to format the data for the next step. All 16,622 images were resized into  $224 \times 224 \times 3$  to train the model. All the collected images are divided into training and validation datasets. 80% (13,301) images are in the training set and another 20% (3321) images are in the validation (Fig. 3.3). We have named the test set as a validation set. And labeled our dataset into two classes, namely

Class 0 (fake Hilsa fish) and Class 1 (Hilsa fish) are shown in Fig. 3.2. All of the images from the training and test sets are rescaled to values between 0 and 1 using Min-Max Normalization (MMN).

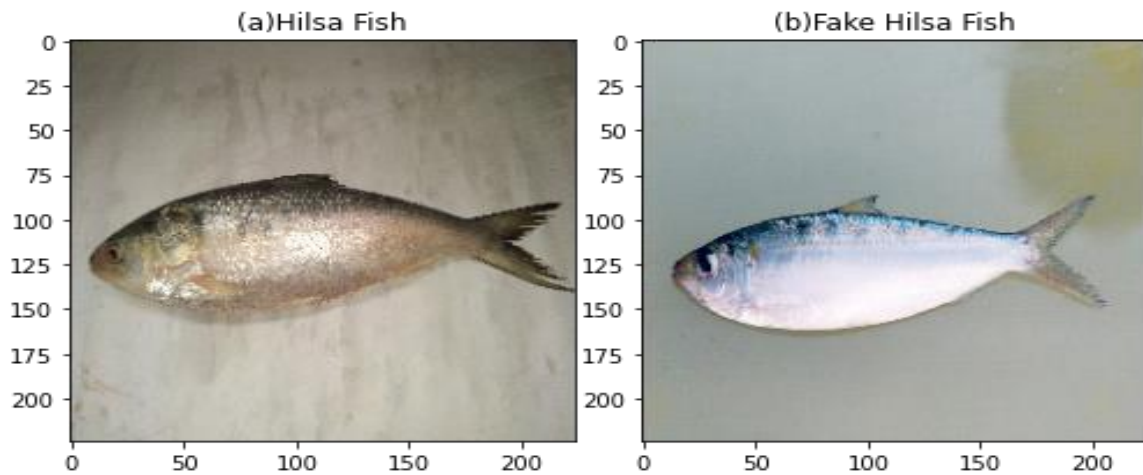


Figure 3.2: Images of two classes a Hilsa fish, b fake Hilsa fish

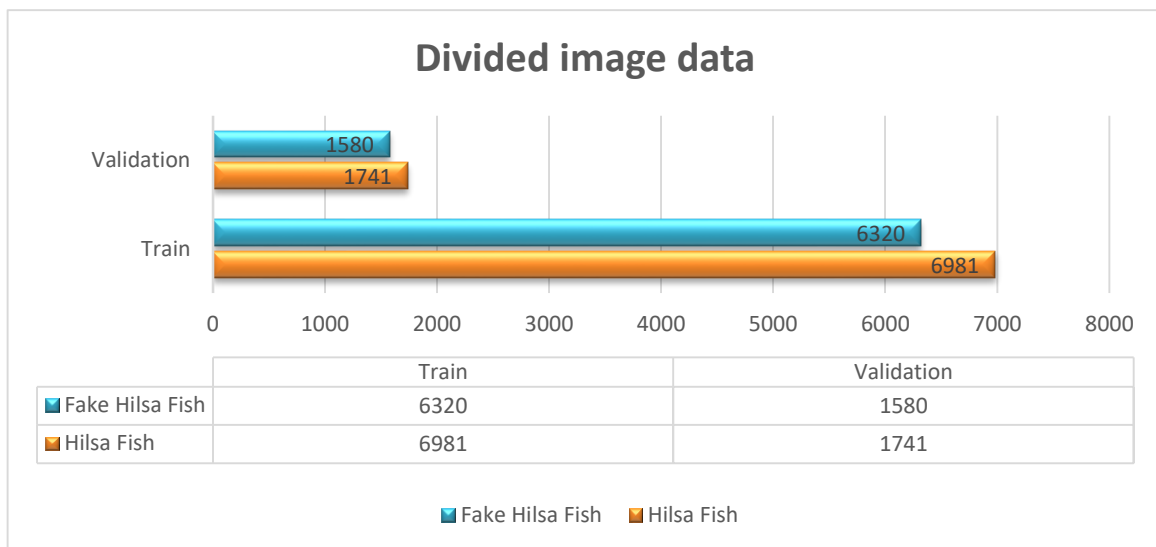


Figure 3.3: Image dataset is divided into training and validation sets

### 3.5 Model Generation

Convolution neural network (CNN) is a type of deep neural network (DNN) that applies to image processing, natural language processing, and many other classifications, recognition, and detection tasks. It includes an input layer, some hidden layers known as convolution layers, pooling layers, flattening layers, and fully connected layers. The convolution layer is the most significant layer for extracting features, edges, colors, shapes, patterns, etc., from the images. The output of this layer is the feature map. Pooling layers reduce the size of an image. Mainly, three types of pooling layers are used: max, average, and sum. It removes 3/4% of the activation, seen in the previous layers. Fully connected layers (FC) are called dense layers [12]. Fully connected layers mean all nodes in one layer connected to the output of the next layers.

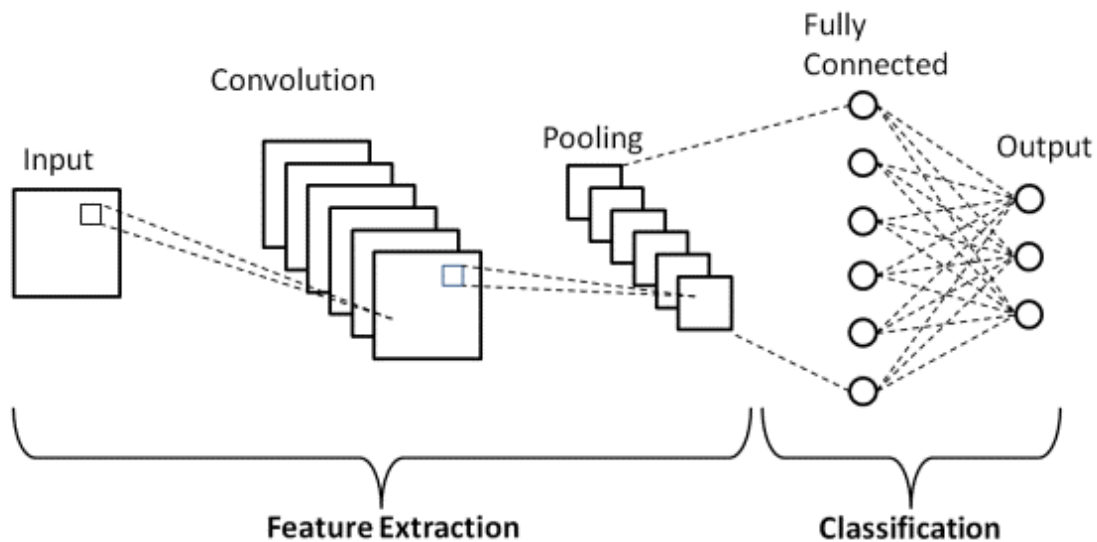


Figure 3.4: Basic Structure of CNN

We use five different types of CNN models to identify the original Hilsa fish and the fake one. The characteristics of these CNN models [11] are shown in Table 1. A brief discussion of these models is given below.

TABLE 3.2: MODEL CHARACTERISTICS IN DETAILS

Model	Size(MB)	Parameters	Depth
Xception	88	22,910,480	126
VGG16	528	138,357,544	23
InceptionV3	92	23,851,784	159
DenseNet201	80	20,242,984	201
NASNetMobile	23	5,326,716	-

### 3.5.1 Xception

Xception [13] was invented by Google researchers. It is an architecture based on depthwise separable convolution layers. It has 36 convolution layers into 14 modules. Parameters are quite similar to InceptionV3. It has been trained with millions of images from the ImageNet database [14].

TABLE 3.3: MODEL SUMMARY OF XCEPTION

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
=====		
xception (Model)	(None, 2048)	20861480
-----		
dropout_1 (Dropout)	(None, 2048)	0
-----		
dense_1 (Dense)	(None, 2)	4098
=====		

Total params: 20,865,578  
Trainable params: 20,811,050



Non-trainable params: 54,528

### 3.5.2 VGG16

VGG16 [15] is sometimes called OxfordNet because it was invented by a visual geometry group from Oxford in 2014. It was first used to win the ILSVRC (ImageNet [14]) competition. The input shape is fixed for the conv1 layer ( $224 \times 224 \times 3$ ). Although its size is quite larger than the rest, it is very useful for learning purposes and easy to implement.

TABLE 3.4: MODEL SUMMARY OF VGG16

Model: "model\_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, None, None, 3)	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73856
block2_conv2 (Conv2D)	(None, None, None, 128)	147584
block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295168
block3_conv2 (Conv2D)	(None, None, None, 256)	590080
block3_conv3 (Conv2D)	(None, None, None, 256)	590080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1180160
block4_conv2 (Conv2D)	(None, None, None, 512)	2359808
block4_conv3 (Conv2D)	(None, None, None, 512)	2359808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0
block5_conv1 (Conv2D)	(None, None, None, 512)	2359808

block5_conv2 (Conv2D)	(None, None, None, 512)	2359808
block5_conv3 (Conv2D)	(None, None, None, 512)	2359808
block5_pool (MaxPooling2D)	(None, None, None, 512)	0
global_average_pooling2d_1 (	(None, 512)	0
dense_1 (Dense)	(None, 1024)	525312
batch_normalization_1 (Batch	(None, 1024)	4096
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
batch_normalization_2 (Batch	(None, 1024)	4096
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 2)	2050
=====		
Total params: 16,299,842		
Trainable params: 16,295,746		
Non-trainable params: 4,096		

### 3.5.3 InceptionV3

InceptionV3 [16, 17] was invented in 2015 by Google Inc. It has a total of 48 deep layers. It has been trained with millions of images from the ImageNet database [14]. InceptionV3 is the most used convolution neural network model for image recognition. The Architecture of InceptionV3 CNN model is too large which cannot be fully disclosed in this article. That is why we have visualized here some beginning layers and some ending layers.

TABLE 3.5: MODEL SUMMARY OF INCEPTIONV3

Model: "model\_1"

Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	(None, 224, 224, 3)	0
conv2d_1 (Conv2D)	(None, 111, 111, 32)	864
batch_normalization_1 (BatchNor	(None, 111, 111, 32)	96

activation_1 (Activation)	(None, 111, 111, 32)	0
conv2d_2 (Conv2D)	(None, 109, 109, 32)	9216
batch_normalization_2 (BatchNor	(None, 109, 109, 32)	96
activation_2 (Activation)	(None, 109, 109, 32)	0
conv2d_3 (Conv2D)	(None, 109, 109, 64)	18432
batch_normalization_3 (BatchNor	(None, 109, 109, 64)	192
activation_3 (Activation)	(None, 109, 109, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_4 (Conv2D)	(None, 54, 54, 80)	5120
batch_normalization_4 (BatchNor	(None, 54, 54, 80)	240
activation_4 (Activation)	(None, 54, 54, 80)	0

(a) Beginning Layers

batch_normalization_94 (BatchNo	(None, 5, 5, 192)	576
activation_86 (Activation)	(None, 5, 5, 320)	0
mixed9_1 (Concatenate)	(None, 5, 5, 768)	0
concatenate_2 (Concatenate)	(None, 5, 5, 768)	0
activation_94 (Activation)	(None, 5, 5, 192)	0
mixed10 (Concatenate)	(None, 5, 5, 2048)	0
global_average_pooling2d_1 (Glo	(None, 2048)	0
dense_1 (Dense)	(None, 1024)	2098176
batch_normalization_95 (BatchNo	(None, 1024)	4096
activation_95 (Activation)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
batch_normalization_96 (BatchNo	(None, 1024)	4096

activation_96 (Activation)	(None, 1024)	0
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 2)	2050
=====		
Total params: 24,960,802		
Trainable params: 24,922,274		
Non-trainable params: 38,528		

### (b) Ending Layers

#### 3.5.4 DenseNet201

DenseNet201 [19] has a total of 201 deep layers. We achieved the highest accuracy by using this model. The input shape is  $224 \times 224 \times 3$  for the “channel-last” data format and  $3 \times 224 \times 224$  is for the “channel-first” format. The summary of the DenseNet201 CNN model is too large which cannot be fully disclosed in this article. That is why we have visualized here some beginning layers and some ending layers.

TABLE 3.6: MODEL SUMMARY OF DENSENET201

Model: "model\_1"

Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	(None, 224, 224, 3)	0
zero_padding2d_1 (ZeroPadding2D)	(None, 230, 230, 3)	0
conv1/conv (Conv2D)	(None, 112, 112, 64)	9408
conv1/bn (BatchNormalization)	(None, 112, 112, 64)	256
conv1/relu (Activation)	(None, 112, 112, 64)	0
zero_padding2d_2 (ZeroPadding2D)	(None, 114, 114, 64)	0
pool1 (MaxPooling2D)	(None, 56, 56, 64)	0
conv2_block1_0_bn (BatchNormali	(None, 56, 56, 64)	256
conv2_block1_0_relu (Activation	(None, 56, 56, 64)	0
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 128)	8192

conv2_block1_1_bn	(BatchNormali	(None, 56, 56, 128)	512
conv2_block1_1_relu	(Activation	(None, 56, 56, 128)	0
conv2_block1_2_conv	(Conv2D)	(None, 56, 56, 32)	36864

#### (a) Beginning Layers

bn	(BatchNormalization)	(None, 7, 7, 1920)	7680
relu	(Activation)	(None, 7, 7, 1920)	0
global_average_pooling2d_1	(Glo	(None, 1920)	0
dense_1	(Dense)	(None, 1024)	1967104
batch_normalization_1	(BatchNor	(None, 1024)	4096
activation_1	(Activation)	(None, 1024)	0
dropout_1	(Dropout)	(None, 1024)	0
dense_2	(Dense)	(None, 1024)	1049600
batch_normalization_2	(BatchNor	(None, 1024)	4096
activation_2	(Activation)	(None, 1024)	0
dropout_2	(Dropout)	(None, 1024)	0
dense_3	(Dense)	(None, 2)	2050

=====

Total params: 21,348,930  
Trainable params: 21,115,778  
Non-trainable params: 233,152

#### (b) Ending Layers

### 3.5.5 NASNetMobile

NASNetMobile [18] is another type of convolution neural network (CNN) that is divided into two cells: a normal cell that returns a feature map in the same aspect, and a reduction cell that reduces the height and width of a feature map. It has been trained with millions of images from the ImageNet database [14]. It has three input channels, height, width, and

RGB color channel. It requires  $224 \times 224 \times 3$  input shapes for images. The Architecture of the NASNetMobile CNN model is too large which cannot be fully disclosed in this article. That is why we have visualized here some beginning layers and some ending layers.

TABLE 3.7: MODEL SUMMARY OF NASNETMOBILE

Model: "model\_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
stem_conv1 (Conv2D)	(None, 111, 111, 32)	864
stem_bn1 (BatchNormalization)	(None, 111, 111, 32)	128
activation_1 (Activation)	(None, 111, 111, 32)	0
reduction_conv_1_stem_1 (Conv2D)	(None, 111, 111, 11)	352
reduction_bn_1_stem_1 (BatchNor	(None, 111, 111, 11)	44
activation_2 (Activation)	(None, 111, 111, 11)	0
activation_4 (Activation)	(None, 111, 111, 32)	0
separable_conv_1_pad_reduction_	(None, 115, 115, 11)	0
separable_conv_1_pad_reduction_	(None, 117, 117, 32)	0

(a) Beginning Layers

activation_188 (Activation)	(None, 7, 7, 1056)	0
global_average_pooling2d_1 (Glo	(None, 1056)	0
dense_1 (Dense)	(None, 1024)	1082368
batch_normalization_1 (BatchNor	(None, 1024)	4096
activation_189 (Activation)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
batch_normalization_2 (BatchNor	(None, 1024)	4096

activation_190 (Activation)	(None, 1024)	0
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 2)	2050
=====		
Total params: 6,411,926		
Trainable params: 6,371,092		
Non-trainable params: 40,834		

(b) Ending Layers

### 3.6 Features Extraction

There are various convolution layers in each CNN model. Convolution layers that process the two-dimensional images are known as conv2d layers. Each conv2d gets input images, having three color channels. Then, this input image is processed through a convolution filter. This filter is known as a convolution kernel (extract features of the input image) or a feature detector. Each input image is arranged in a matrix that consists of combining the value of three color channels. The matrix tends to get smaller for each filter to read the features of the input image, therefore, there are many differences to see the same image in each filter. High-level filters read the special pattern of input images and low-level filters read the normal features of the input image [20]. Each filter normally carries pre-defined weights. The term feature extraction means the reduction of features and RGB from the raw input image in every conv2d layer. Each conv2d has some arguments [21] or parameters for image processing, those are filter, kernel-size, strides, padding, data format, dilation-rate, groups, activation, use bias, and a few more. These arguments are different in each CNN model which we have used in our research to identify Hilsa fishes. In our research, 32 feature maps are extracted by each conv2d layer. We can see from Fig. 5, many changed images of the Hilsa fish after applying filters in each conv2d. Among these filters, some have highlighted the edges, some have highlighted the shape, and some have highlighted the background and heat map. CNN reads the images in each conv2d layer and learns the features of every input image for doing better classification.

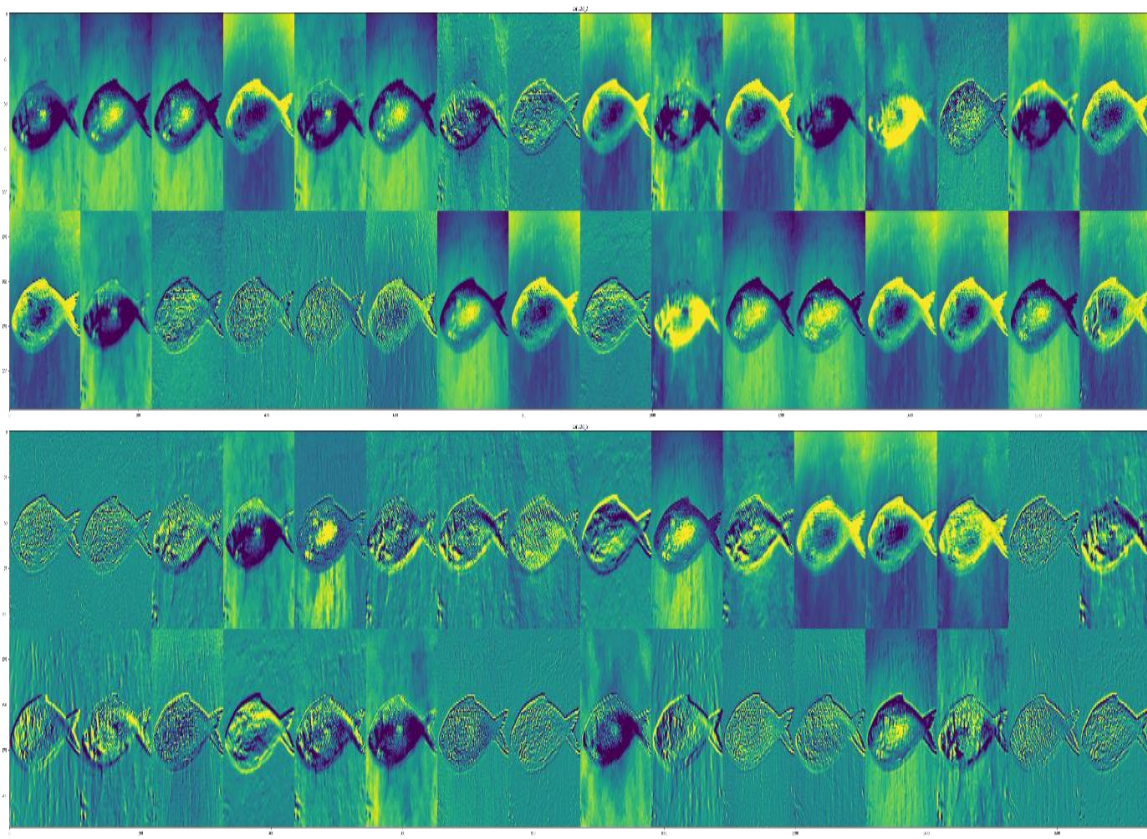


Figure 3.5: Visualize the features extraction from the first two conv2d (DenseNet201)

### 3.7 Parameter Setting

We've predefined several model parameters, also known as hyper-parameters, in our research. If we do not define those parameters for the model, this will not perform. To achieve the highest performance, we must first tune them. The hyper-parameters of our model are seen in Table 3.8. We used the same parameters for all five Convolutional Neural Networks (CNN).



TABLE 3.8: PARAMETERS AND FUNCTIONS FOR TRAINING PROCEDURE.

Training Parameters	Xception	VGG16	InceptionV3	DenseNet201	NASNetMobile
Epochs	5	5	5	5	5
Batch Size	16	16	16	16	16
Input Shape	224x224x3	224x224x3	224x224x3	224x224x3	224x224x3
Output Class	2	2	2	2	2
Train Samples	13301	13301	13301	13301	13301
Test Samples	3321	3321	3321	3321	3321
Re-scaling	1/255	1/255	1/255	1/255	1/255
Weights	Imagenet	Imagenet	Imagenet	Imagenet	Imagenet
Optimizer	Adam	Adam	Adam	Adam	Adam
Learning Rate	1e-5	1e-5	1e-5	1e-5	1e-5
Beta_1 Rate	0.9	0.9	0.9	0.9	0.9
Beta_2 Rate	0.999	0.999	0.999	0.999	0.999
Epsilon Rate	1e-08	1e-08	1e-08	1e-08	1e-08
Decay Rate	1e-5	1e-5	1e-5	1e-5	1e-5
Loss Function	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy

## CHAPTER 4

### EXPERIMENTAL RESULT ANALYSIS

#### 4.1 Introduction

This section discusses the experimental results of the five CNN models. We train those models with 6981 original Hilsa and 6320 fake Hilsa fish images. Models are tested with 1741 Hilsa fish and 1580 fake Hilsa images to evaluate performance. After that NASNetMobile gave the lowest accuracy of 86.75% and DenseNet201 gave the highest accuracy of 97.02%.

#### 4.2 Confusion Matrix and Evaluation Metrics

A confusion matrix is the measure and visualized the total right and wrong predictions made doing by the classifiers or models. It is predicted the result for two-class positive and negative, those two-class, work with the main four terms true positive(TP), true negative(TN), false positive(FP), and false negative (FN). When the model predicts it is a hilsa fish and the actual output is also hilsa fish then it is a true positive. When the model predicts it is a fake hilsa fish and the actual output also says it is fake hilsa fish then that term is a true negative. When the classifier predicts it a hilsa fish but the actual output is fake hilsa fish then it is a false positive. After all when the classifier predicts it is a fake hilsa fish and the actual output says that this is a hilsa fish then it is a false negative. For measurement of the performance of every classifier predicted ability, we will discuss some evaluation metrics such as accuracy, precision, f1 score, sensitivity or recall, specificity, false positive rate and false negative rate. The equation of these seven evaluation matrices as follows:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (1)$$

$$Precisions = \frac{TP}{TP+FP} \quad 100\% \quad (2)$$

$$F1\ score = 2X \frac{Precisions \times Recall}{Precisions + Recall} \quad (3)$$

$$Sensitivity\ or\ Recall = \frac{TP}{TP+FN} \quad (4)$$

$$Specificity = \frac{TN}{FP+TN} \quad (5)$$

$$False\ Positive\ Rate(FPR) = \frac{FP}{FP+TN} \quad (6)$$

$$False\ Negative\ Rate(FNR) = \frac{FN}{FN+TP} \quad (7)$$

### 4.3 Experimental Results & Discussion

The outputs of the 5 models are shown in this section.

#### 4.3.1 Experimental Result Analysis for NASNetMobile

##### **Accuracy:**

Here is a record of the variation of accuracy in each epoch. In 1<sup>st</sup> epochs, the training accuracy was 82.62% and validation accuracy was 96.98%. In the 5<sup>th</sup> epochs the training accuracy was 89.10% and validation accuracy 78.02%

```
Epoch 1/5
156/156 [=====] - 162s 1s/step - loss: 0.4064 - accuracy: 0.8262 - val_loss: 0.1068 - val_accuracy: 0.9698
Epoch 2/5
156/156 [=====] - 198s 1s/step - loss: 0.3896 - accuracy: 0.8357 - val_loss: 0.2801 - val_accuracy: 0.9597
Epoch 3/5
156/156 [=====] - 165s 1s/step - loss: 0.3371 - accuracy: 0.8594 - val_loss: 0.0419 - val_accuracy: 0.9758
Epoch 4/5
156/156 [=====] - 211s 1s/step - loss: 0.3149 - accuracy: 0.8734 - val_loss: 0.2961 - val_accuracy: 0.8024
Epoch 5/5
156/156 [=====] - 240s 2s/step - loss: 0.2812 - accuracy: 0.8910 - val_loss: 0.5456 - val_accuracy: 0.7802
```

Figure 4.1: Each Epochs Accuracy and Loss for NASNetMobile

##### **Training Accuracy and Loss Vs Validation Accuracy and Loss:**

Here, the accuracy and loss of Figure 4.1 are shown through the graph.

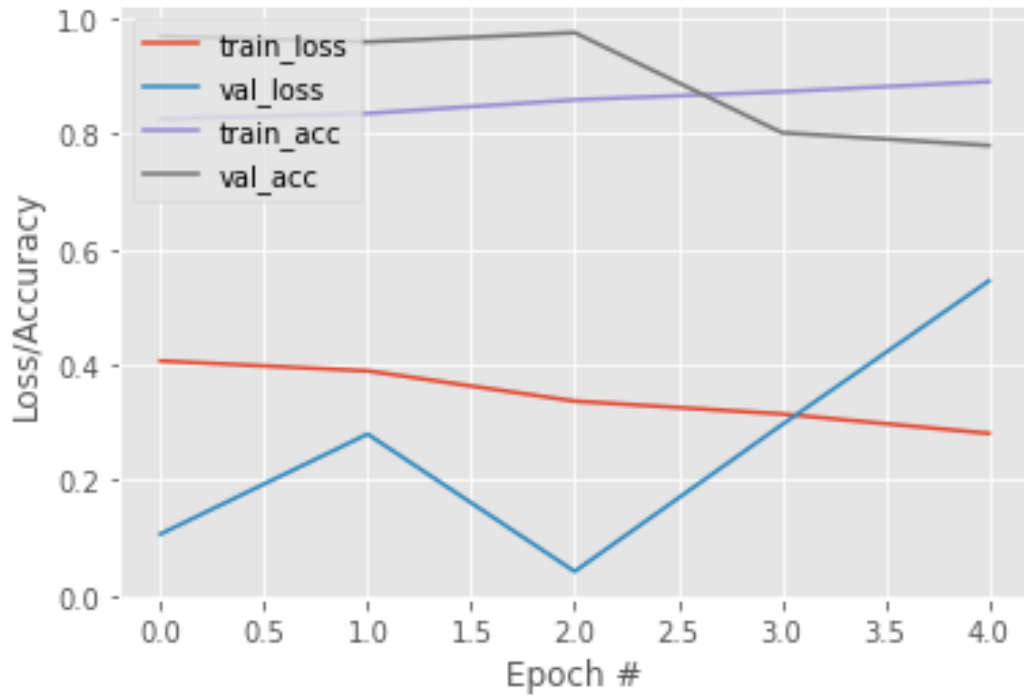


Figure 4.2: Training Accuracy and Loss Vs Validation Accuracy and Loss for NASNetMobile

### Confusion Matrix:

From this Confusion Matrix, we can see that this model has performed well in terms of Fake Hilsa fish classification.

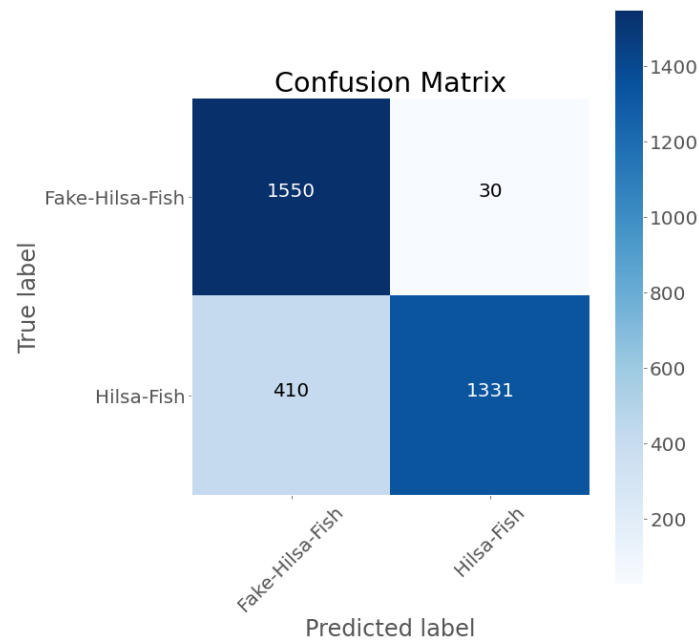


Figure 4.3: Confusion Matrix for NASNetMobile

### Classification Report:

Here Precision, recall, f1-score, and accuracy for each class are shown in figure 4.4.

	precision	recall	f1-score	support
Fake-Hilsa-Fish	0.79	0.98	0.88	1580
Hilsa-Fish	0.98	0.76	0.86	1741
accuracy			0.87	3321
macro avg	0.88	0.87	0.87	3321
weighted avg	0.89	0.87	0.87	3321

Figure 4.4: Classification Report for NASNetMobile

### ROC/AUC Curve:

The probability is represented by the ROC curve, and the degree or measure of separability is represented by the AUC.

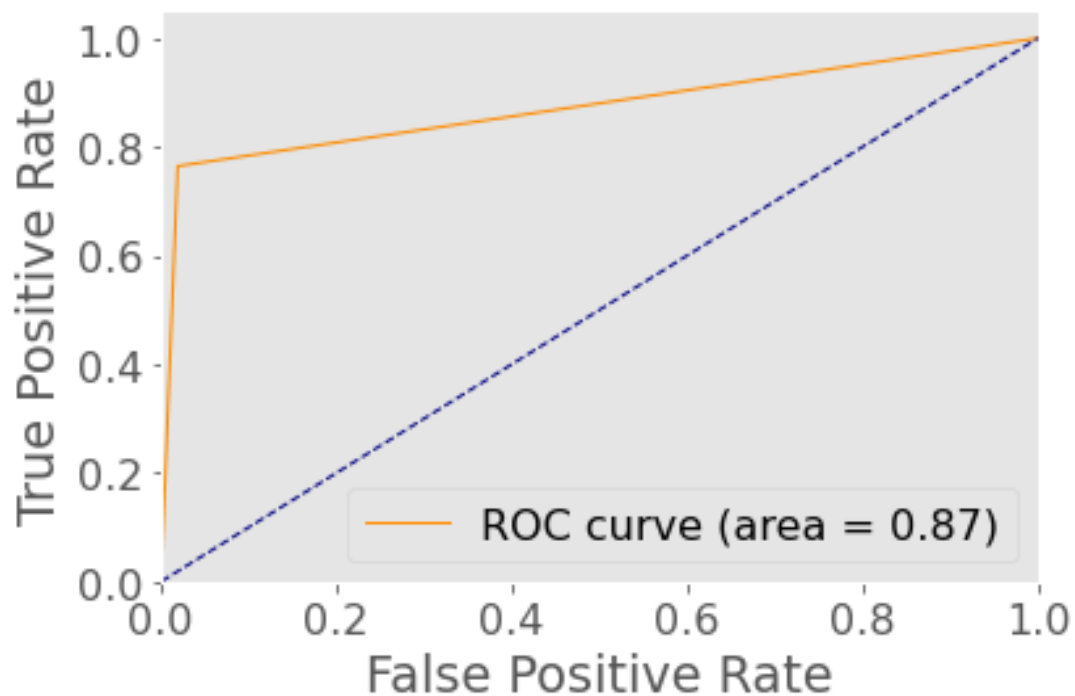


Figure 4.5: ROC/AUC for NASNetMobile

### Visualized Sensitivity and Specificity:

Here the values of Sensitivity and Specificity are shown through graphs.

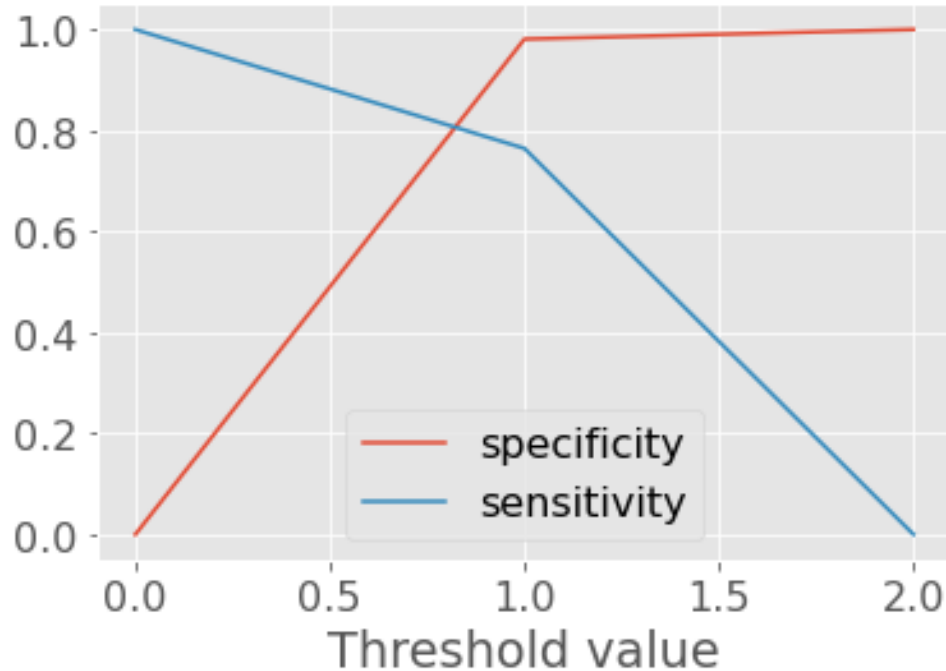


Figure 4.6: Sensitivity and Specificity for NASNetMobile

### 4.3.2 Experimental Result Analysis for VGG16

#### Accuracy:

Here is a record of the variation of accuracy in each epoch.

```
Epoch 1/5
187/187 [=====] - 86s 457ms/step - loss: 0.4321 - accuracy: 0.9054 - val_loss: 0.2857 - val_accuracy: 0.9520
Epoch 2/5
187/187 [=====] - 81s 431ms/step - loss: 0.3770 - accuracy: 0.9231 - val_loss: 0.1757 - val_accuracy: 0.9985
Epoch 3/5
187/187 [=====] - 92s 493ms/step - loss: 0.3775 - accuracy: 0.9201 - val_loss: 0.2126 - val_accuracy: 0.9622
Epoch 4/5
187/187 [=====] - 96s 511ms/step - loss: 0.3722 - accuracy: 0.9296 - val_loss: 0.4343 - val_accuracy: 0.9826
Epoch 5/5
187/187 [=====] - 92s 491ms/step - loss: 0.3279 - accuracy: 0.9425 - val_loss: 0.1520 - val_accuracy: 0.9119
```

Figure 4.7: Each Epochs Accuracy and Loss for VGG16

### Training Accuracy and Loss Vs Validation Accuracy and Loss:

Here, the accuracy and loss of Figure 4.8 are shown through the graph.

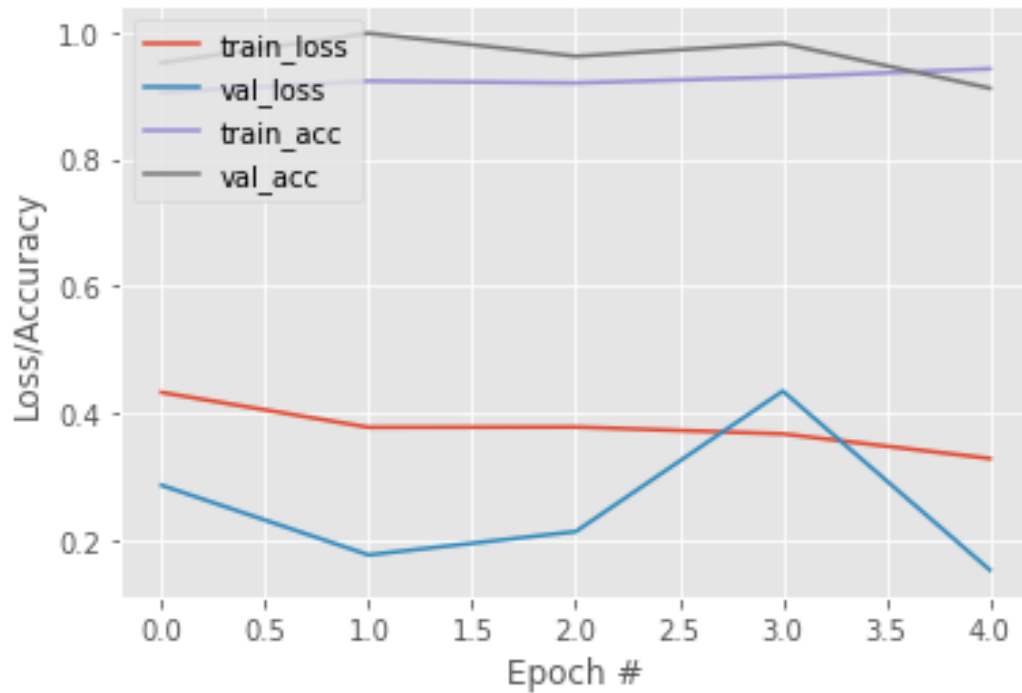


Figure 4.8: Training Accuracy and Loss Vs Validation Accuracy and Loss for VGG16

### Confusion Matrix:

From this Confusion Matrix, we can see that this model has performed well in terms of Fake Hilsa fish classification.

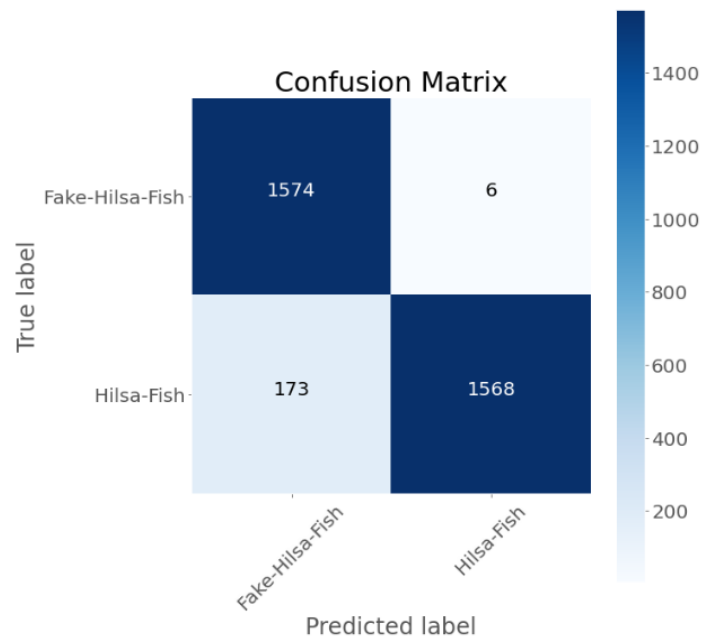


Figure 4.9: Confusion Matrix for VGG16

### Classification Report:

Here Precision, recall, f1-score, and accuracy for each class are shown in figure 4.10.

	precision	recall	f1-score	support
Fake-Hilsa-Fish	0.90	1.00	0.95	1580
Hilsa-Fish	1.00	0.90	0.95	1741
accuracy			0.95	3321
macro avg	0.95	0.95	0.95	3321
weighted avg	0.95	0.95	0.95	3321

Figure 4.10: Classification Report for VGG16

### ROC/AUC Curve:

The probability is represented by the ROC curve, and the degree or measure of separability is represented by the AUC.

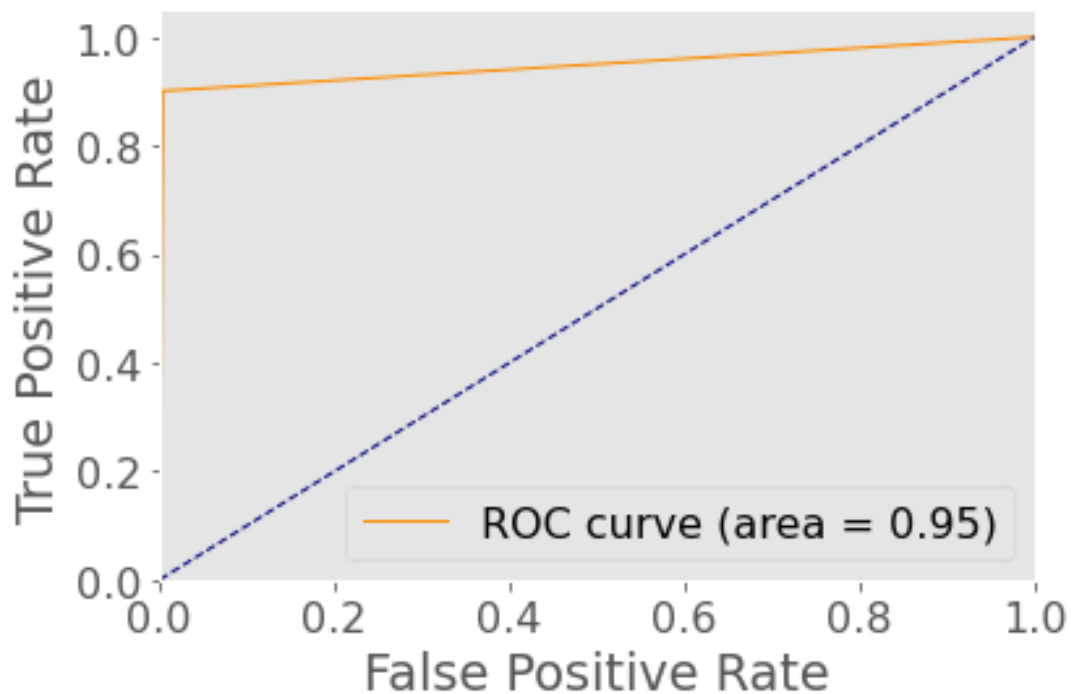


Figure 4.11: ROC/AUC for VGG16



## Visualized Sensitivity and Specificity:

Here the values of Sensitivity and Specificity are shown through graphs.

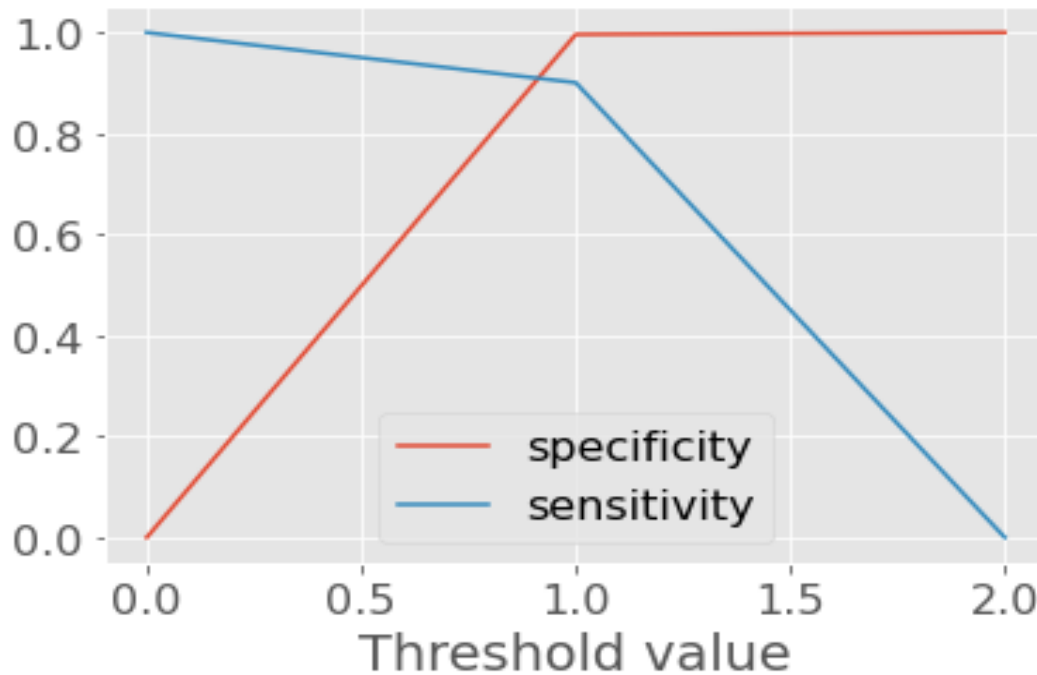


Figure 4.12: Sensitivity and Specificity for VGG16

### 4.3.3 Experimental Result Analysis for Xception

#### Accuracy:

Here is a record of the variation of accuracy in each epoch.

```
Epoch 1/5
156/156 [=====] - 1453s 9s/step - loss: 0.6336 - accuracy: 0.6362 - val_loss: 0.4779 - val_accuracy: 0.7036
Epoch 2/5
156/156 [=====] - 1352s 9s/step - loss: 0.4351 - accuracy: 0.8225 - val_loss: 0.4770 - val_accuracy: 0.8992
Epoch 3/5
156/156 [=====] - 1440s 9s/step - loss: 0.3123 - accuracy: 0.8794 - val_loss: 0.2223 - val_accuracy: 0.9315
Epoch 4/5
156/156 [=====] - 1383s 9s/step - loss: 0.2350 - accuracy: 0.9107 - val_loss: 0.0534 - val_accuracy: 0.9395
Epoch 5/5
156/156 [=====] - 1334s 9s/step - loss: 0.1824 - accuracy: 0.9395 - val_loss: 0.1382 - val_accuracy: 0.9677
```

Figure 4.13: Each Epochs Accuracy and Loss for Xception

#### Training Accuracy and Loss Vs Validation Accuracy and Loss:

Here, the accuracy and loss of Figure 4.14 are shown through the graph.

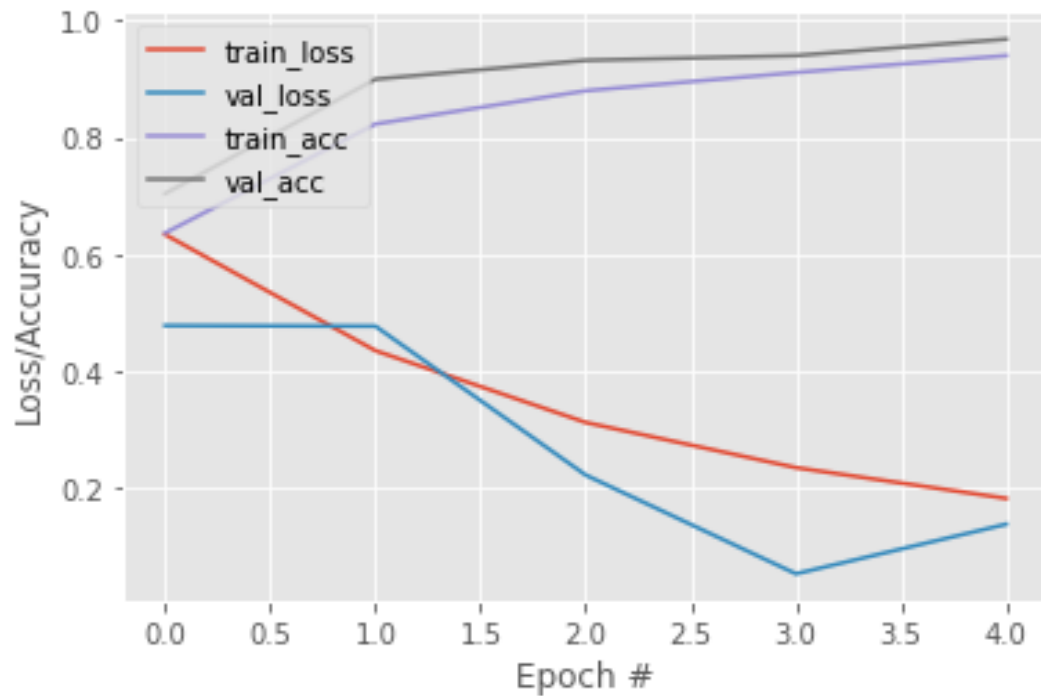


Figure 4.14: Training Accuracy and Loss Vs Validation Accuracy and Loss for Xception

### Confusion Matrix:

From this Confusion Matrix, we can see that this model has performed well in terms of Fake Hilsa fish classification.

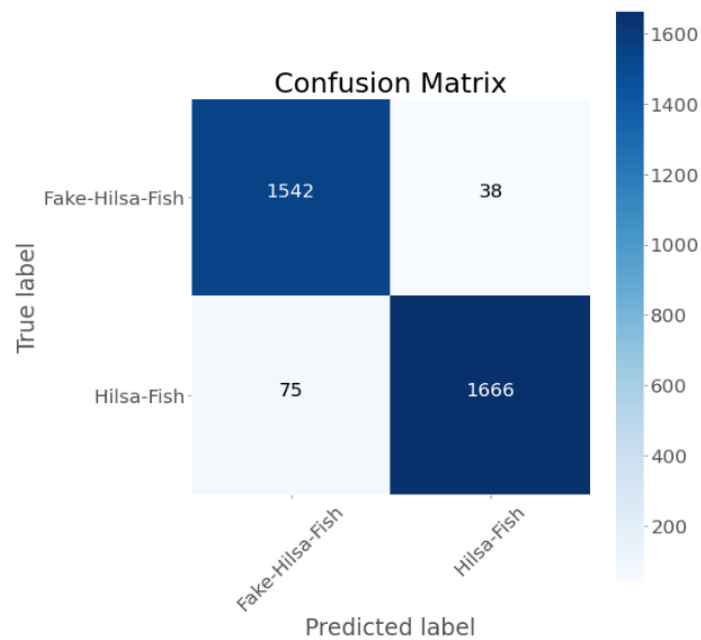


Figure 4.15: Confusion Matrix for Xception

### Classification Report:

Here Precision, recall, f1-score, and accuracy for each class are shown in figure 4.16.

	precision	recall	f1-score	support
Fake-Hilsa-Fish	0.95	0.98	0.96	1580
Hilsa-Fish	0.98	0.96	0.97	1741
accuracy			0.97	3321
macro avg	0.97	0.97	0.97	3321
weighted avg	0.97	0.97	0.97	3321

Figure 4.16: Classification Report for Xception

### ROC/AUC Curve:

The probability is represented by the ROC curve, and the degree or measure of separability is represented by the AUC.

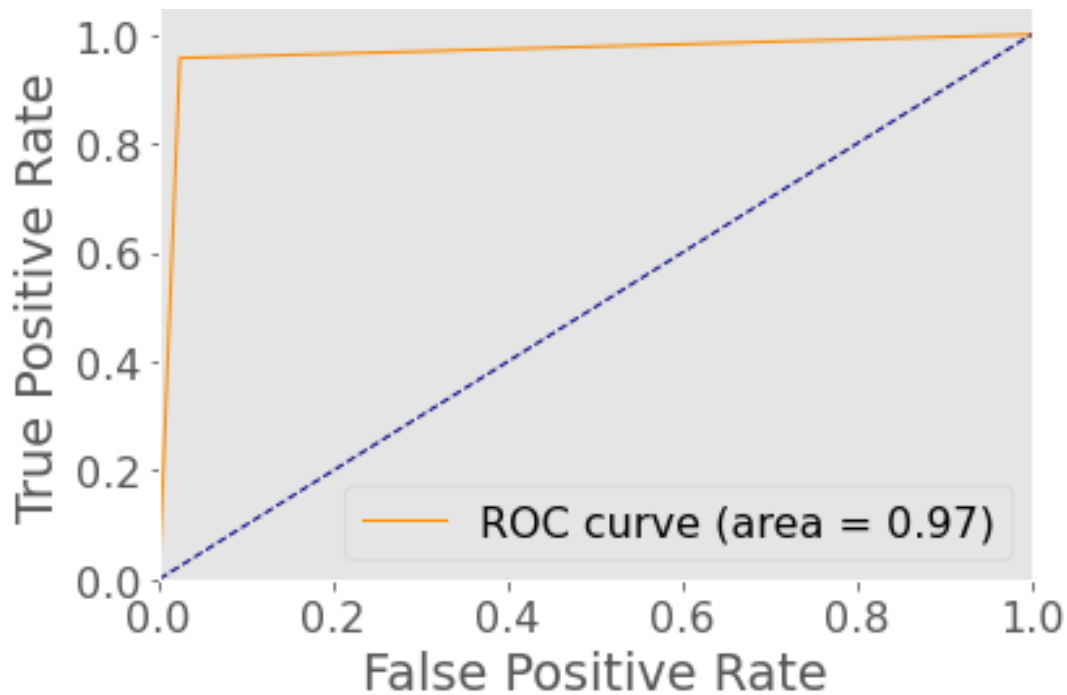


Figure 4.17: ROC/AUC for Xception

### Visualized Sensitivity and Specificity:

Here the values of Sensitivity and Specificity are shown through graphs.

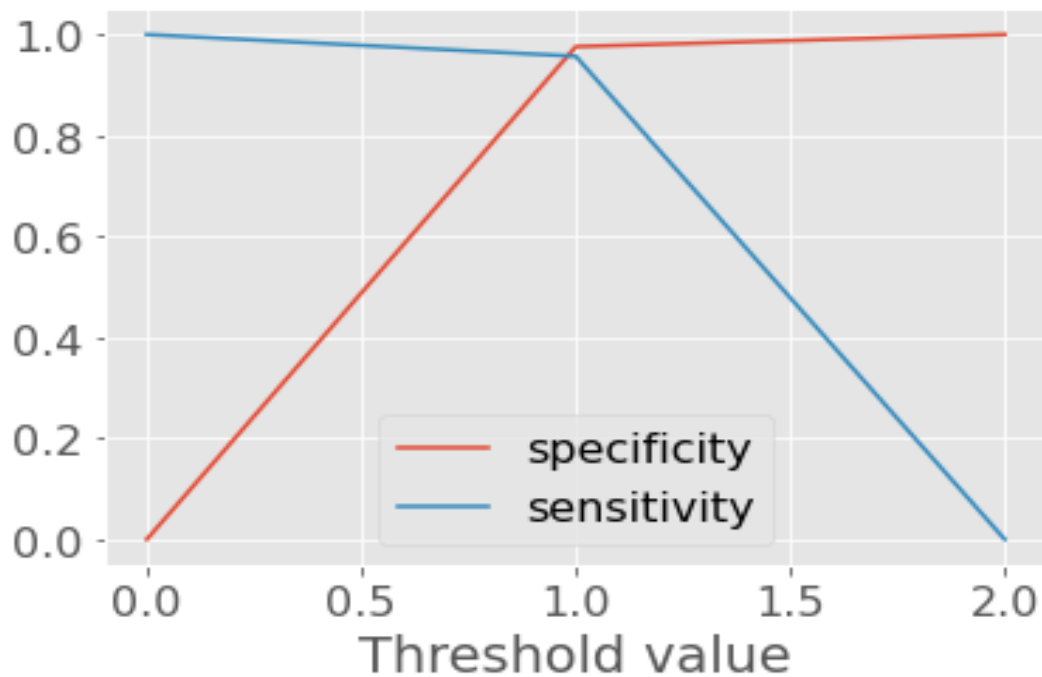


Figure 4.18: Sensitivity and Specificity for Xception

### 4.3.4 Experimental Result Analysis for InceptionV3

#### Accuracy:

Here is a record of the variation of accuracy in each epoch.

```
Epoch 1/5
187/187 [=====] - 117s 627ms/step - loss: 0.4886 - accuracy: 0.9011 - val_loss: 0.3597 - val_accuracy: 0.9869
Epoch 2/5
187/187 [=====] - 115s 613ms/step - loss: 0.4639 - accuracy: 0.9081 - val_loss: 0.2699 - val_accuracy: 0.9826
Epoch 3/5
187/187 [=====] - 125s 670ms/step - loss: 0.4377 - accuracy: 0.9238 - val_loss: 0.2610 - val_accuracy: 0.9637
Epoch 4/5
187/187 [=====] - 127s 680ms/step - loss: 0.4561 - accuracy: 0.9238 - val_loss: 0.3939 - val_accuracy: 0.9419
Epoch 5/5
187/187 [=====] - 126s 673ms/step - loss: 0.4042 - accuracy: 0.9388 - val_loss: 0.3221 - val_accuracy: 0.9501
```

Figure 4.19: Each Epochs Accuracy and Loss for InceptionV3

### Training Accuracy and Loss Vs Validation Accuracy and Loss:

Here, the accuracy and loss of Figure 4.20 are shown through the graph.

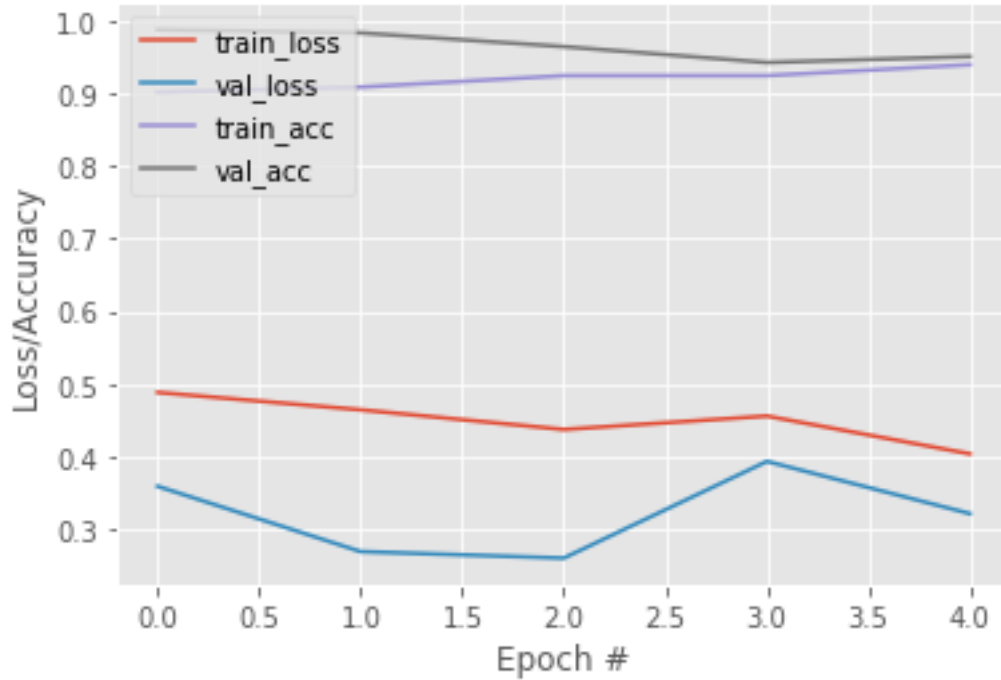


Figure 4.20: Training Accuracy and Loss Vs Validation Accuracy and Loss for InceptionV3

### Confusion Matrix:

From this Confusion Matrix, we can see that this model has performed well in terms of Fake Hilsa fish classification.

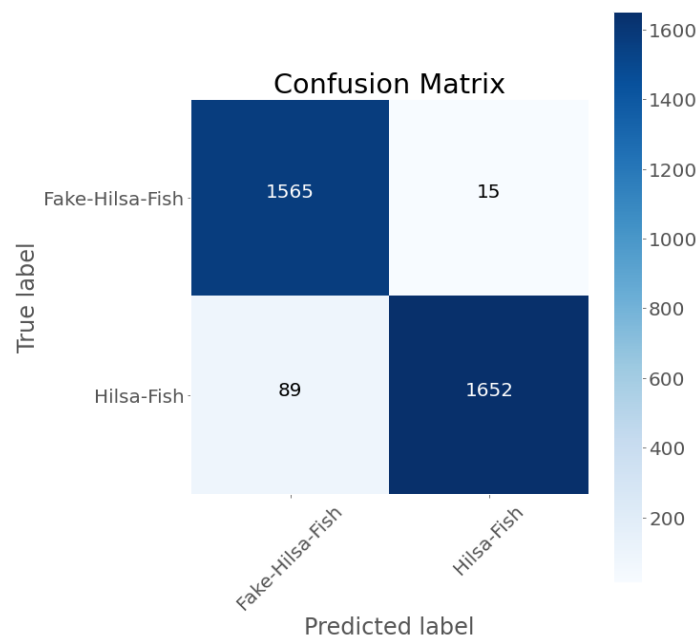


Figure 4.21: Confusion Matrix for InceptionV3

### Classification Report:

Here Precision, recall, f1-score, and accuracy for each class are shown in figure 4.22.

	precision	recall	f1-score	support
Fake-Hilsa-Fish	0.95	0.99	0.97	1580
Hilsa-Fish	0.99	0.95	0.97	1741
accuracy			0.97	3321
macro avg	0.97	0.97	0.97	3321
weighted avg	0.97	0.97	0.97	3321

Figure 4.22: Classification Report for InceptionV3

### ROC/AUC Curve:

The probability is represented by the ROC curve, and the degree or measure of separability is represented by the AUC.

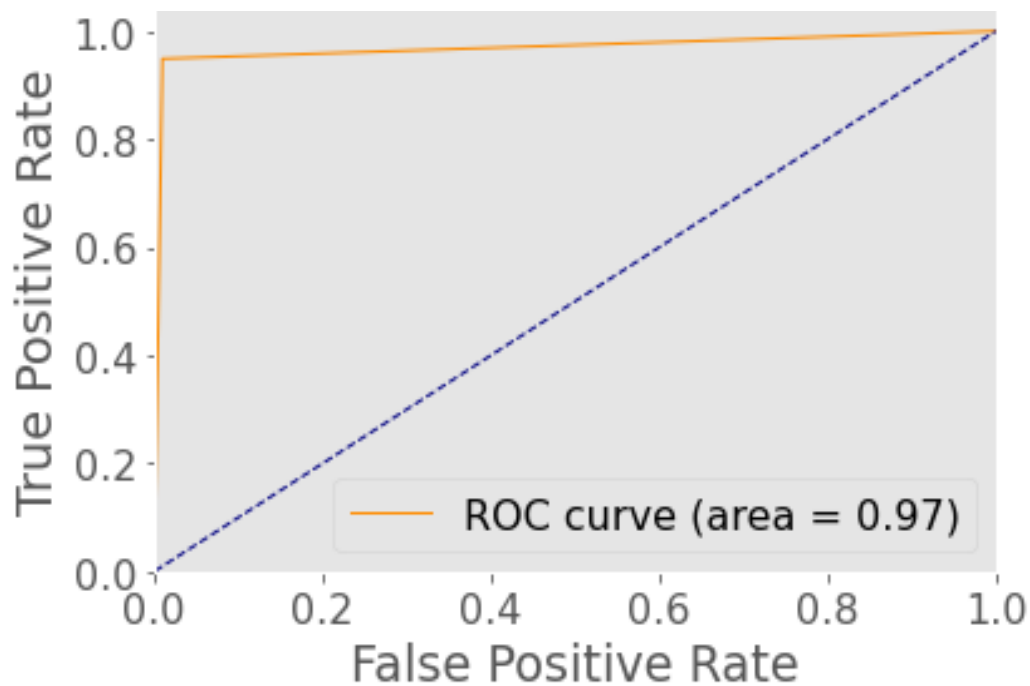


Figure 4.23: ROC/AUC for InceptionV3

### Visualized Sensitivity and Specificity:

Here the values of Sensitivity and Specificity are shown through graphs.

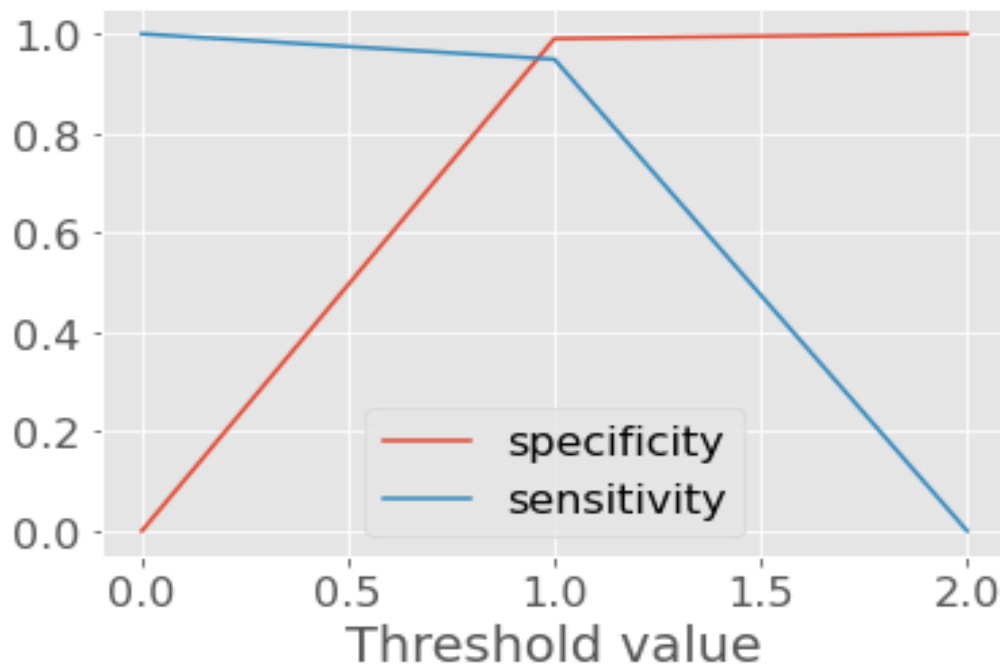


Figure 4.24: Sensitivity and Specificity for InceptionV3

### 4.3.5 Experimental Result Analysis for DenseNet201

#### Accuracy:

Here is a record of the variation of accuracy in each epoch.

```
Epoch 1/5
156/156 [=====] - 1124s 7s/step - loss: 1.0085 - accuracy: 0.6206 - val_loss: 0.4968 - val_accuracy: 0.8629
Epoch 2/5
156/156 [=====] - 1082s 7s/step - loss: 0.7315 - accuracy: 0.7728 - val_loss: 0.6043 - val_accuracy: 0.9496
Epoch 3/5
156/156 [=====] - 1105s 7s/step - loss: 0.6178 - accuracy: 0.8277 - val_loss: 0.2870 - val_accuracy: 0.9718
Epoch 4/5
156/156 [=====] - 1082s 7s/step - loss: 0.5280 - accuracy: 0.8716 - val_loss: 0.2528 - val_accuracy: 0.9496
Epoch 5/5
156/156 [=====] - 1109s 7s/step - loss: 0.4616 - accuracy: 0.8962 - val_loss: 0.2636 - val_accuracy: 0.9536
```

Figure 4.25: Each Epochs Accuracy and Loss for DenseNet201

### Training Accuracy and Loss Vs Validation Accuracy and Loss:

Here, the accuracy and loss of Figure 4.26 are shown through the graph.

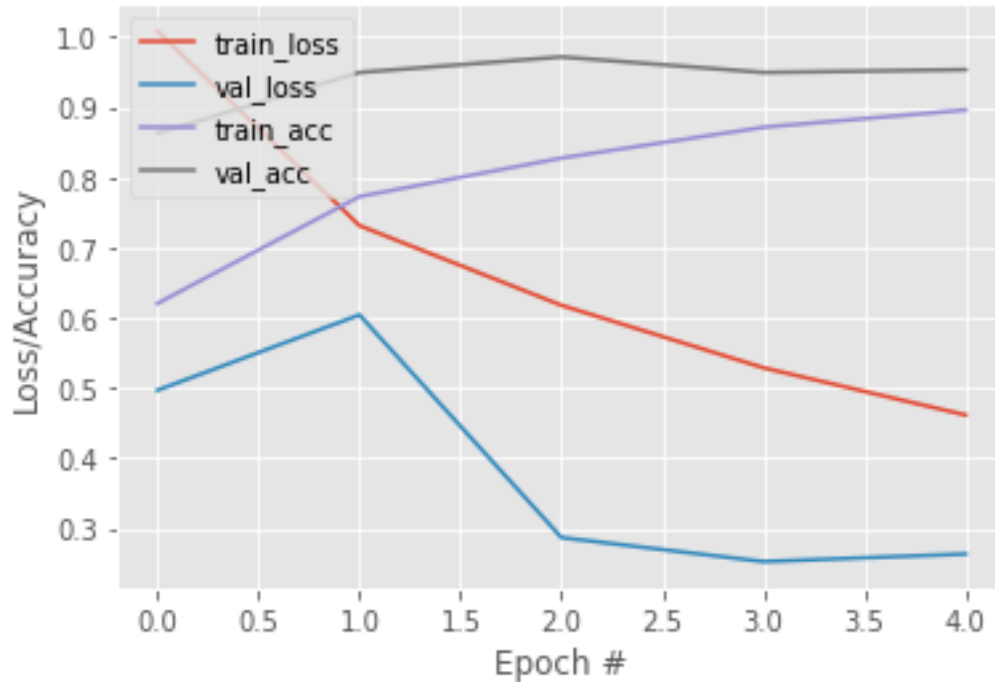


Figure 4.26: Training Accuracy and Loss Vs Validation Accuracy and Loss for DenseNet201

### Confusion Matrix:

From this Confusion Matrix, we can see that this model has performed well in terms of Fake Hilsa fish classification.

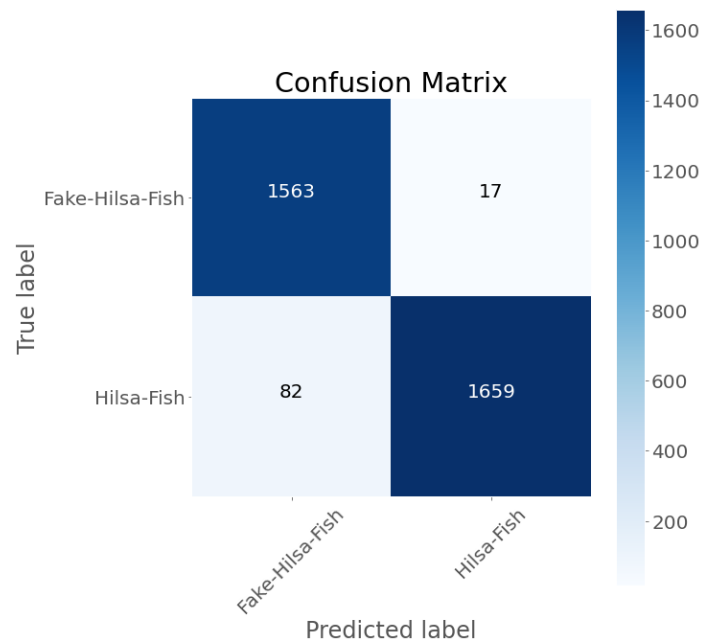


Figure 4.27: Confusion Matrix for DenseNet201



### Classification Report:

Here Precision, recall, f1-score, and accuracy for each class are shown in figure 4.28.

	precision	recall	f1-score	support
Fake-Hilsa-Fish	0.95	0.99	0.97	1580
Hilsa-Fish	0.99	0.95	0.97	1741
accuracy			0.97	3321
macro avg	0.97	0.97	0.97	3321
weighted avg	0.97	0.97	0.97	3321

Figure 4.28: Classification Report for DenseNet201

### ROC/AUC Curve:

The probability is represented by the ROC curve, and the degree or measure of separability is represented by the AUC.

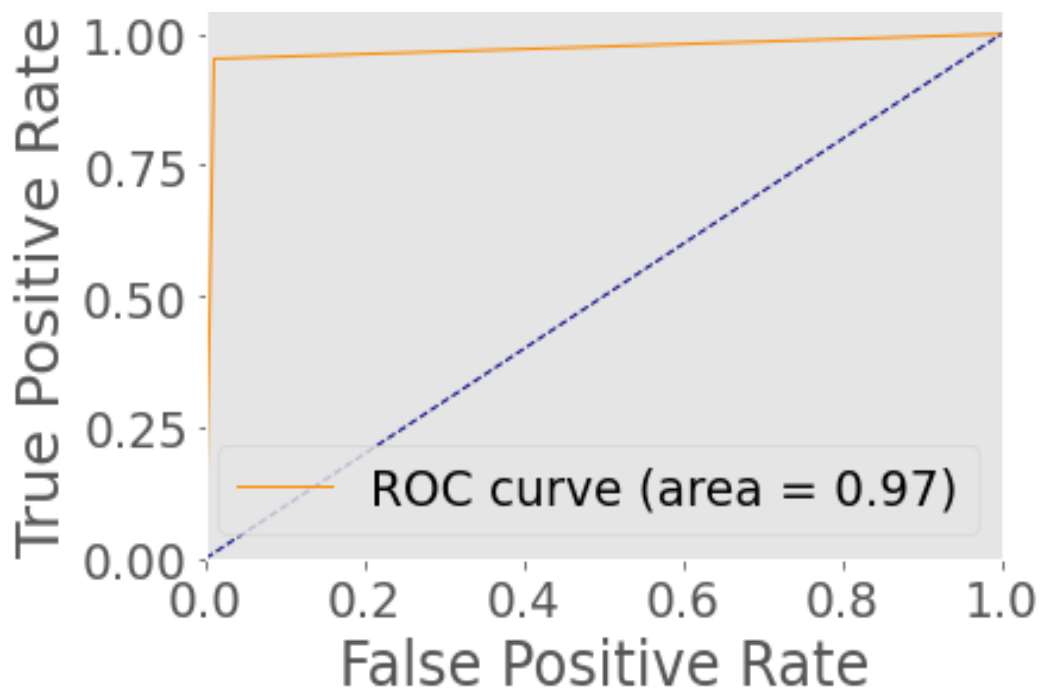


Figure 4.29: ROC/AUC for DenseNet201

### Visualized Sensitivity and Specificity:

Here the values of Sensitivity and Specificity are shown through graphs.

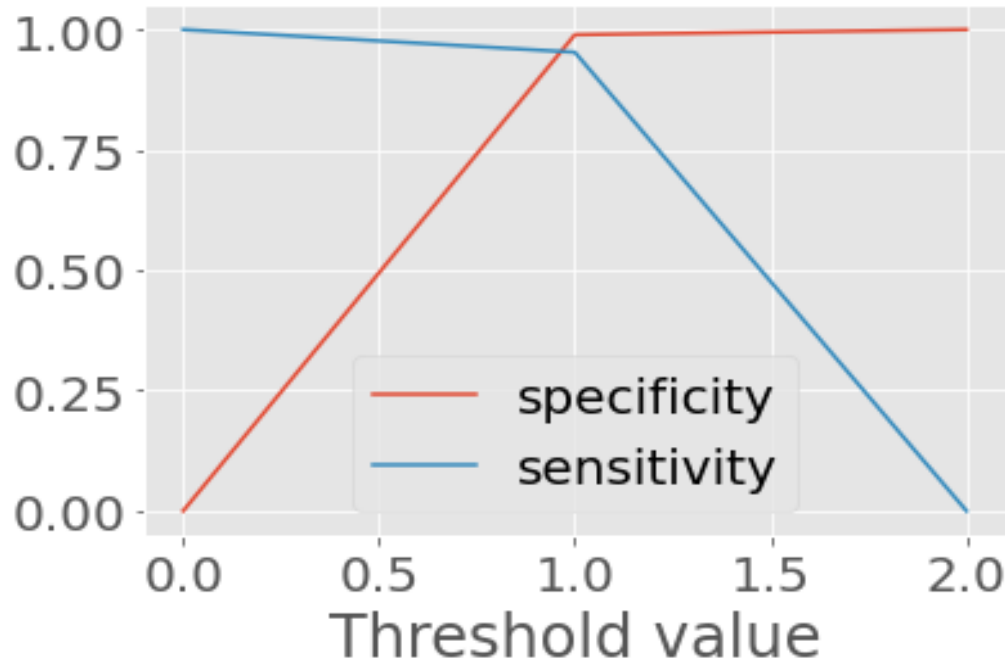


Figure 4.30: Sensitivity and Specificity for DenseNet201

## 4.4 Comparative Analysis

Table 4.1 lists the results of the confusion matrix for the five CNN models. According to Table 4.2, we found in NasnetMobile that the difference between the false-positive rate and the false-negative rate is high. DenseNet201 shows a significant improvement because the difference between FP and TP rate is low rather than the NasnetMobile. The accuracy measure of Xception, Inceptionv3, and DenseNet201 are very close to each other. The accuracy of Xception is 96.60% and InceptionV3 is 96.87%. Here the accuracy measured difference between them is very low which is 0.27%. The accuracy of DenseNet201 is 97.02%, the difference of its accuracy with InceptionV3 is only 0.15%. Based on the data in Table 4.1, it can be stated that the DenseNet201 is giving better classification performance than the other model. From Table 4.2, we observe that the F1 score is increasing along with the accuracy from top to bottom. Here one noticeable thing is that NASNetMobile has a difference between sensitivity and specificity.

TABLE 4.1: CONFUSION MATRIX RESULT FOR FIVE CNN MODELS.

Model	True Positive(TP)	True Negative(TN)	False Positive(FP)	False Negative(FN)	Accuracy (%)
NASNetMobile	1331	1550	30	410	86.75
VGG16	1568	1578	6	173	94.61
Xception	1666	1542	38	75	96.60
InceptionV3	1652	1565	15	89	96.87
DenseNet201	1659	1563	17	82	97.02

TABLE 4.2: CLASSIFICATION REPORT FOR FIVE CNN MODELS.

Model	Accuracy (%)	Precision (%)	F1 (%)	Sensitivity (%)	Specificity (%)	FPR (%)	FNR (%)
NASNetMobile	86.75	97.80	85.82	79.08	97.79	1.89	23.54
VGG16	94.61	99.62	94.60	90.09	99.61	0.37	9.93
Xception	96.60	97.77	96.72	95.36	97.76	2.40	4.30
InceptionV3	96.87	99.10	96.95	94.61	99.10	0.94	5.11
DenseNet201	97.02	98.99	97.10	95.01	98.98	1.07	4.79

From the intensive result analysis, we can conclude that DenseNet201 gives a better classification performance in our experiment.

## **CHAPTER 5**

### **IMPACT ON SOCIETY, ENVIRONMENT, AND SUSTAINABILITY**

#### **5.1 Impact on Society**

The identification of fake Hilsa fish using a Deep Conventional Neural Network would have a practical effect on society. It is a fact that the selling of fake Hilsa fish has risen at an alarming pace, and this must be stopped immediately. Since a large portion of the population buys Hilsa Fish on a daily basis, it is important to recognize the real Hilsa from the fake ones. Many unscrupulous traders are exporting these fake fish to other countries, tarnishing the country's image. This research study can be used to stop the syndicate from exporting fake Hilsa fish to another country. Our society will be spared from this dark scam in this way.

#### **5.2 Impact on Environment**

Our study would have no negative impact on the environment. We didn't use any materials that could be harmful to the environment. Our research study would help the entire human race

#### **5.3 Ethical Aspects**

This proposed methodology isn't unethical, and it doesn't violate fundamental freedoms in any way. Personal details, such as a person's name and personality, are not harmed by the methodology. As a result, there would be no problem with privacy. Using this methodology and Learning Techniques to introduce automatic fake Hilsa fish detection can be done with ease.

#### **5.4 Sustainability Plan**

Any solid project idea that can be implemented in real life requires a sustainability plan. Smart mobile applications can be developed to instantly identify Fake and Real Hilsa fish native to a specific area, as well as obtain valuable information such as nutrient information.

## **CHAPTER 6**

### **CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH**

#### **6.1 Summary of the Study**

To complete our research, we used deep learning-based binary classification to detect original Hilsa fish automatically. Image processing is an introspective topic for researchers in the research sector. For our study, we collected a large number of raw images and labeled them according to their class. Then we pre-processed them using different processing techniques. For classification, we used five different Keras Applications, which are deep learning models that come with pre-trained weights.

#### **6.2 Conclusions**

The main objective of this research describes the proposed methodology to identify original Hilsa fishes. This is the first study of original Hilsa and fake Hilsa identification using convolutional neural network (CNN). We used five CNN models in our experiments and observed their performance. Then, we have also performed an observational comparative analysis of our obtained results. Among them, NASNetMobile achieved the least performance with an accuracy of 86.75%, VGG16 achieved 94.61%, Xception achieved 96.60%, InceptionV3 achieved 96.87% accuracy, and DenseNet201 produce a very good performance with an accuracy of 97.02%. We hope that this study will be beneficial for researchers who will research related topics.

#### **6.3 Implication for Further Study**

Since we have performed binary classification on our research, we have placed the images of the original Hilsa fish into one class and some other types of fake Hilsa fish into another class. In the future, we will keep each Fake Hilsa fish in a separate class and performed a multi-class classification. With this, we want to enrich our dataset with more images and develop a mobile application that can be used to identify the original Hilsa fish in real time.

## REFERENCES

- [1] Mredul, M. M. H., Uddin, M. E., Pervez, A. K. M. K., Yesmin, F., and Akkas, A. B.: Food aid program during restricted hilsa fishing period: effectiveness and management perspective. *Journal of Fisheries*, 8(1), pp. 752–761 (2020).
- [2] Urbanova P., Bozhynov V., Čiřá P., and Zelezný M.: Classification of Fish Species Using Silhouettes. In: Rojas I., Valenzuela O., Rojas F., Herrera L., Ortuño F. (eds), *Bioinformatics and Biomedical Engineering. IWBBIO 2020. Lecture Notes in Computer Science*, vol 12108. Springer, Cham (2020).
- [3] Sharmin, I., Islam, N.F., Jahan, I., Joye, T.A., Rahman, M.R. and Habib, M.T.: Machine vision based local fish recognition. *SN Applied Sciences*, 1(12), pp.1529, (2019).
- [4] Chhabra H.S., Srivastava A.K., Nijhawan R.: A Hybrid Deep Learning Approach for Automatic Fish Classification. In: Singh P., Panigrahi B., Suryadevara N., Sharma S., Singh A. (eds), *Proceedings of ICETIT 2019. Lecture Notes in Electrical Engineering*, vol. 605, pp. 427–436, Springer, Cham (2020).
- [5] R. M. Hernandez, A. A. Hernandez.: Classification of Nile Tilapia using Convolutional Neural Network, In 9th International Conference on System Engineering and Technology (ICSET), IEEE, pp. 126-131, Shah Alam, Malaysia, (2019).
- [6] Hafiz, T.R., M., Ikram, U., L., Saliha., Syed, Z., H., S., Rehman, A., U., Bukhari, S., A., C. : Visual features based automated identification of fish species using deep convolutional neural networks. *Computers and Electronics in Agriculture*, 167, pp. 105075,(2019).
- [7] Yu, X., Zeng, N., Liu, S.et al.: Utilization of DenseNet201 for diagnosis of breast abnormality. *Machine Vision and Applications* 30, 1135–1144 (2019).
- [8] Hu J., Li D., Duan Q., Han Y., Chen G., Si X.: Fish species classification by color, texture and multi-class support vector machine using computer vision.*Computers and Electronics in Agriculture*, 88, pp. 133–140 (2012).
- [9] Ogunlana, S.O., Olabode, O., Oluwadare, S., & Iwasokun, G.: Fish Classification Using Support Vector Machine. ISSN 2006-1781 Vol 8. No. 2 (2015).
- [10] Fouad, M., Zawbaa, H., El-Bendary, N., & Hassanien, A.: Automatic Nile Tilapia fish classification approach using machine learning techniques. 13Th International Conference On Hybrid Intelligent Systems (HIS 2013).
- [11] Keras, <https://keras.io/api/applications/>, last accessed May 2020.
- [12] Raghuveer, L.VS., Kumar, C.S., Sri, G.S., Habeeb, S.: A review on Practical Diagnostic of Tomato Plant Diseases. *Suraj Punj Journal For Multidisciplinary Research*, vol. 9, pp. 432435, (2019).
- [13] Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, p. 1251–1258 (2017).
- [14] ImageNet, <http://www.image-net.org>, last accessed May 2020.

- [15] K. Simonyan and A. Zisserman.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv :1409.1556, (2014).
- [16] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: Going deeper with convolutions: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); pp. 1–9, Boston, MA,(2015).
- [17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna.: Rethinking the Inception Architecture for Computer Vision, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.. 2818–2826, Las Vegas, NV, (2016).
- [18] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le.: Learning Transferable Architectures for Scalable Image Recognition, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8697–8710, Salt Lake City, UT, (2018).
- [19] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger.: Densely Connected Convolutional Networks, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269, Honolulu, HI, (2017).
- [20] Siddiqui, S. A., Salman, A., Malik, M. I., Shafait, F., Mian, A., Shortis, M. R., and Harvey, E. S.: Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data. – ICES Journal of Marine Science, 75(1), pp. 374–389 (2017).
- [21] Keras, [https://keras.io/api/layers/convolution\\_layers/convolution2d/](https://keras.io/api/layers/convolution_layers/convolution2d/), Last accessed May 2020.

## Fake Hilsa Fish Detection Using Machine Learning Algorithm

### ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

5%

PUBLICATIONS

5%

STUDENT PAPERS

### PRIMARY SOURCES

Exclude quotes On

Exclude bibliography On

Exclude matches < 5%