DEEP LEARNING BASED APPROACH FOR IDENTIFICATION OF LOCAL FISH

BY

MASUM SHAH JUNAYED
ID: 151-15-5008

AFSANA AHSAN JENY
ID: 151-15-5278
AND

NAZMUS SADAT JISAN
ID: 151-15-5116

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

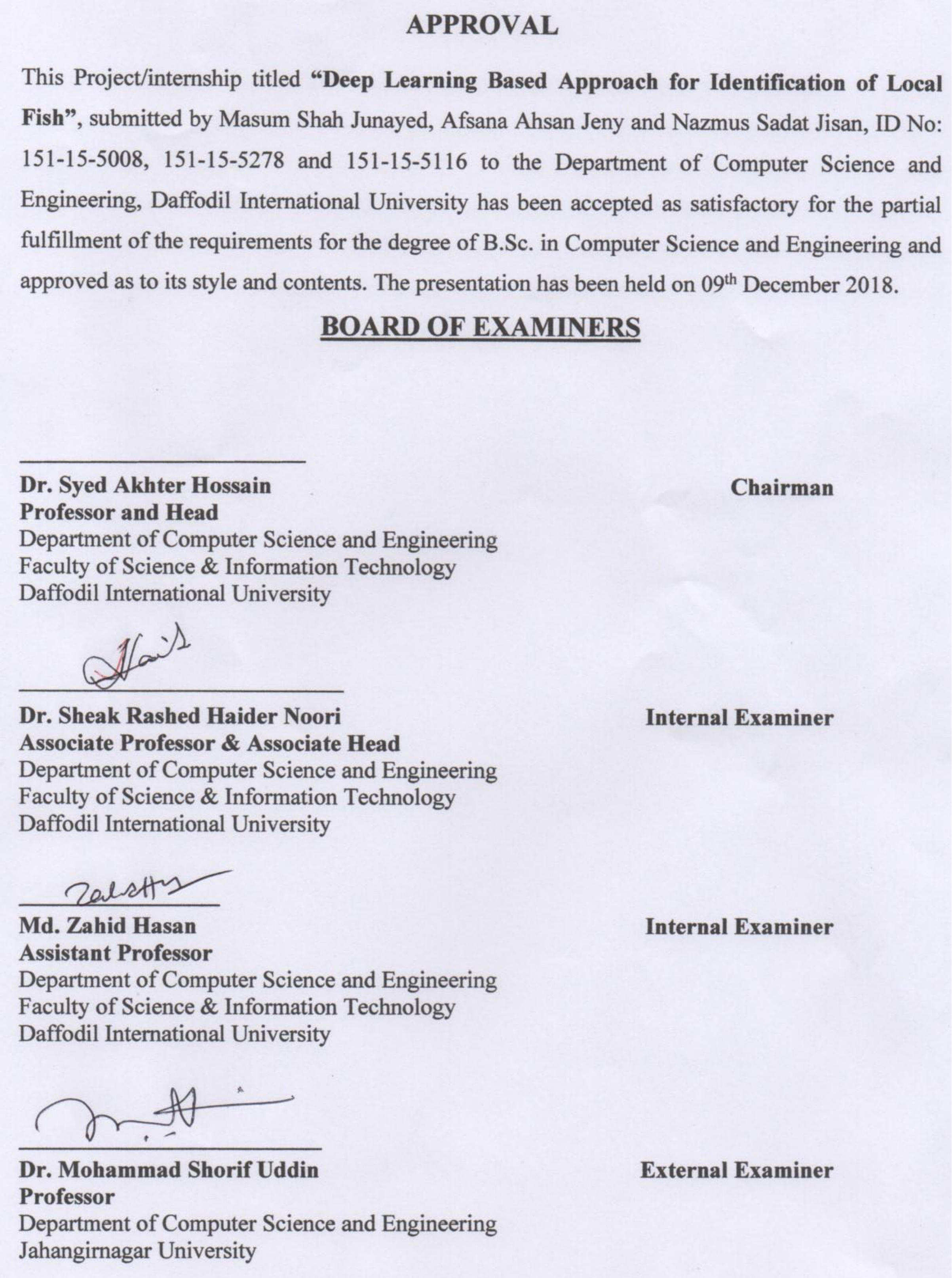
Supervised By:

Md. Tarek Habib
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2018



APPROVAL

This Project/internship titled **"Deep Learning Based Approach for Identification of Local Fish"**, submitted by Masum Shah Junayed, Afsana Ahsan Jeny and Nazmus Sadat Jisan, ID No: 151-15-5008, 151-15-5278 and 151-15-5116 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 09th December 2018.

BOARD OF EXAMINERS

____

Dr. Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Dr. Sheak Rashed Haider Noori **Internal Examiner**
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Md. Zahid Hasan **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Dr. Mohammad Shorif Uddin **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

This is to certify that the work presented in this thesis is the outcome of the analysis and experiment carried out by us under the supervision of **Md. Tarek Habib, Assistant Professor, Department of CSE,** Daffodil International University (DIU) Dhaka, Bangladesh. 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:	**CO-SUPERVISED BY:**
Md. Tarek Habib Assistant Professor Department of CSE Daffodil International University	**Md. Sadekur Rahman** Assistant Professor Department of CSE Daffodil International University

SUBMITTED BY:

Masum Shah Junayed
ID: 151-15-5008
Department of CSE
Daffodil International University

Afsana Ahsan Jeny
ID: 151-15-5278
Department of CSE
Daffodil International University

Nazmus Sadat Jisan
ID: 151-15-5116
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty of Allah for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Md. Tarek Habib**, **Assistant Professor**, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Deep Learning*" to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to the Almighty Allah 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

Bangladesh is considered one of the most suitable area for fish culture with the world's largest climate wetland with thousands of rivers and ponds and being a fish-loving nation. Learning a classification of fish can help people to identify the local fish. Various types of fish are classified based on their characteristics so that people can scientifically describe different types of fish easily. The classification of the different species of internal relationships and their consensus of an animal is specially considered. Differences between fish size and size are so much that it is very difficult to detect them. The people of new era use Mobile phones and other devices to shoot fishes but they became confused to identify fish. For this reason, we take a purpose for identifying fish. For our experiment, we have used total 6000 images of local fish with 10 categories. We have used Convolutional Neural Network models, those are Inception-V3, MobileNet, ResNet50, and Xception and they have obtained a high accuracy of 98.07%, 98.41%, 97.65%, and 95.53% respectively on our dataset.

TABLE OF CONTENTS

CONTENTS **PAGE**

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Introduction

Bangladesh has a long history of the role of fish species. Unfortunately, during the importation of this species, the quarantine system was not taken at all and consequently, these were introduced in the country without proper documentation. Information such as published and published lists of harmful effects; such scientific information is not available on the environmental and economic effects of species. It is difficult to identify the exact time of fish in the absence of research.

Bangladesh is a riverine country. In Bangladesh, there are 401 species of oceanic fishes and 251 species of local fishes (in freshwaters and brackish waters). Some kind of inland fishes or local fishes is now becoming rare. Most of the people of Bangladesh don’t know about the name of local fishes. As they don’t know about the name so they always take pictures of fish and try to recognize using the internet. But most of the time it is very time consuming and they can’t easily detect the name of those fishes. “Ilish” is the national fish of Bangladesh. Except this, there are lots of fish in Bangladesh such as Balichura, Anju, Arwari, Bagair, Baim, Bacha, Boitka, Bhetki, Balichura, Bati, Bechi, Bele, Bhola, Bhetki, Chapila, Chebli, Chela, Chitol, Pholi, Chuno, Darkina, Gagla, Gilipunti, Gojar, Gutum, Grass carp, Kakila, Kajuli, Kalibaus, Koi, Lotey, Mola, Magur, Punti, Pabda, Rui, Taki, Tengra, Topshe, Telapia, Titari, Utii and so on.

1.2 Motivation

Every fish is different from each other by size, structure, color, and outlook. In this modern era, the new generation is more attracted to the marine fish but if we can’t recognize the local fish, these will vanish from the country. On account of this, we have to know about our culture so we can say that our work plays an important role in our country.

1.3 Rationale of the Study

Here, we have used a Convolutional Neural Network (CNN) [5] which is an effective detection system that has been exhibited in current years. It is based on brain structure. It consists of a network of neurons teaching units. This network ignores the complex processing of the image, therefore, people can get the real image right way. It uses local receivable fields as brain neurons by sharing it weight and link information reduces training and restrictions compared to neural system. It has a specific degree translation, rotation and image distortion conversion.

There are many pretrained models of CNN [5] such as LeNet-5 [2], AlexNet [3], VGG 16 [4], Inception-V3 [1] [14], ResNet50 [6] [15], ResNeXt [7], DenseNet [8] and also depthwise separable convolution models [17] like MobileNet [9], Xception [10] and so on. In this paper, we have applied Inception-V3 [1] [14], ResNet50 [6] [15], MobileNet [9] and Xception [10] for identifying local fish of Bangladesh and the CNN [5] models have given a high accuracy on our dataset. We have taken 10 classes which are Baim, Bacha, Bele, Chitol, Gutum, Kakila, Koi, Punti, Taki and Tengra for our experiment.

1.4 Research Questions

1. What are the "very interesting" research topics that focus on the deep teaching community recently?
2. Do we need a lot of data to train deep learning models?
3. Does the Convolutional Neural Network make learning more interesting?
4. What are the research areas of deep learning?
5. What is the difference between Deep Learning and Machine Learning?

1.5 Expected Output

We hope that our experiment will work in the future and help the people who are intersected to deep learning. In the modern era, the deep learning is very necessary who are interested to research work. It makes learning more interesting. Though it is very necessary so that people can easily recognize the local fish of Bangladesh in a wonderful way. Hope, our experiment will help people in the future.

1.6 Report Layout

The content of this paper is organized as follows: In Section II, the similarity works have discussed with background study. In Section III, the system architecture, data collection, fish, feature and the proposed models are discussed. In Section IV, the experimental evaluations are described. In Section V, the result analysis is discussed. And finally, In Section VI, the future work and the synopsis of the conclusion are described.

CHAPTER 2
BACKGROUND STUDY

2.1 Introduction

Bangladesh is mainly an agrarian economy and naturally abundant sweet water resources and the longest continuous beach in the world. With the world's largest floodwater wetlands, behind China and India, only the third largest aquatic biodiversity of Asia is considered as the most suitable area for climate and fishery in the world. There is an inland water area of about 45,000 km 2 and about 710 km long coastal belt of the country. Given this vast water supply, it is clear that fish plays an important role in the economy and the population of the population. Fish and fish products provide 60 percent of the animal protein and about three percent of the total export earnings. By using local fish there is no identification work happened actually. We think that many people still don't know about the inland fish of Bangladesh. If they don't recognize the local fish will vanish very quickly. That's why we have to take an approach to identify the local fish.

2.2 Related Works

There is some work on Aquarium fish, marine fish, underwater fish, and river fish pictures. We have summarized some of those below.

In 2016, Sushil Kumar Mahapatra, Sumant Kumar, Sakuntala Mahapatra, Shuvendra Kumar used Multithreading Fuzzy C mean Algorithm for detecting underwater fishes in a noisy and dense condition with high detection rate. They have used 1200 video frames. Their accuracy was good and it was 98.35%. But this method is to work quickly for small displacement [18].

In 2017, Xiu Li, Youhua Tang, Tingwei Gao used the deep convolutional neural network to detect fish. They have used 24,277 fish images belonging to 12 classes. Their accuracy was 89.95%. But the underlying stages should be used to prevent unwanted information and to create a trade-off between accuracy and calculation, to prevent general normalization from losing too much information, they used two learned parameters [19].

In 2016, Nhat D. M. Nguyen, Kien N. Huynh, Nhan N. Vo, Tuan Van Pham used Gaussian Mixture Model and Frame-Differencing algorithm (CGMMFD) to Fish detection and tracking. They have gotten higher tracking accuracy but despite having low complexity and a high-efficiency Identification, this method shows some errors. Threshold T and S values are often set by hand according to our experience. If the selection is too large, the target of identification can also be empty. Conversely, the words will be plenty [20].

In 2014, Lars M. Wolff and Sabah Badri-Hoeher conducted acoustic-optical Underwater Fish Observatory (UFO) for Fish Detection in Shallow Waters. Their dataset was more than 50000 sonar images and accuracy was 79%. But the drawbacks of their experiment was the Sonar has a brief visual range. Therefore, the actual size and species of fish cannot be detected [21].

In 2017, R. Prados, R. García, N. Gracias, L. Neumann, and Håvard Vågstøl proposed a method which is Deep Vision system consisting a trawl to detect fish in Trawl Nets using more than 100,000 underwater images. There is a stereo camera setup connected to a trowel, which can be detected and measured if not able to bring them in the boat. But their color information is not available, and are taken from a larger distance than in the Scantrol Deep Vision system [22].

In 2017, Sukhpal Kaur, Dr. Rajinder Singh conducted Mobile Ad-Hoc Networks (MANET) to detect Jellyfish. This paper reflects the idea of attacking jellyfish and various techniques related to its detection and prevention. Their accuracy was only 70% because it can only give a review of jellyfish [23].

In 2008, M. Chambah, D. Semani, A. Renouf, P. Courtellemont, A. Rizzi used ACE model (Automatic Color Equalization) an unsupervised color equalization algorithm. They have used 12 fish species and 1346 regions. It has an open, robust, and local filtering feature that leads to more effective results. Recover images give better results when displayed or processed. But there is no universal color constancy method [24].

In 2017, Minsung Sung, Son-Cheol Yu, and Yogesh Girdhar proposed a convolutional neural network based techniques based on You Only Look Once algorithm. It has 929 fish pictures and notes. For training 829 images and 100 images for testing. Their sensitivity was 93% and specificity was 62%. A bit cannot prevent non-fish objects classified as fish classified and float in the image itself and then records x, y coordinate, width and height and height of the flats.

2.3 Research Summary

In our paper, we have used Convolutional Neural Networks [5] model to identify the local fish of Bangladesh. We have conducted Inception-V3 [1] [14], Mobilenet [9], ResNet50 [6] [15], and Xception [10]. We have experimented these model with total 6000 images of 10 classes which are Baim, Bacha, Bele, Chitol, Gutum, Kakila, Koi, Punti, Taki, and Tengra and we have gotten high accuracy.

2.4 Scope of the Problem

Our experiment is based on deep Convolutional Neural Network models those are Inception-V3 [1] [14], ResNet50 [6] [15], MobileNet [9] and Xception [10]. We have applied this model to detect local fish of Bangladesh which can't identify easily using the internet. Because those fishes we have used which are not available. People use cameras, mobile phones, and other devices for fishing, but people will also be confused because they do not know the fish species. That's why to solve this problem, we have to take an approach to detect the local fish. These classified procedures are completely artificial, big to loads, and professional staffers. Through the development of computer technology and digital image processing technology, people started to discover the method of automatic fish classification by computers. In our experiment, The CNN [5] models work well and also give a high accuracy to identify fish.

2.5 Challenges

We have faced some challenges at the time of this experiment. The followings are some of the challenges that we have faced:

1. We couldn’t get the local fish for our experiment. Because the local fish just get in the village, not to get into the city. This was so much challenging for us.
2. For getting these fishes, we went to the physical village market from the city and then collected the fish with a lot of trouble. Because we couldn’t get the fishes in the same places. We had to go to different places for collecting local fish.
3. Some of the local fish was similar in their color, shape, and structure. So it was so much difficult to identify.
4. Before starting this experiment, we thought that the CNN [5] model will be conflicted. But finally, the models worked well.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In this section, the remaining part is as follows: first, we provide a system architecture which is the procedure of our experiment; second, we describe the tools of our experiment; third, we describe the collection of our fish dataset; fourth, we discuss the feature of our dataset; fifth, we describe the Convolutional Neural Network [5] models with details description those we have used in our experiment; finally, we describe the binary and multiple class confusion matrix.

The following system architecture “Figure 3.1” represents the procedure of our experiment.

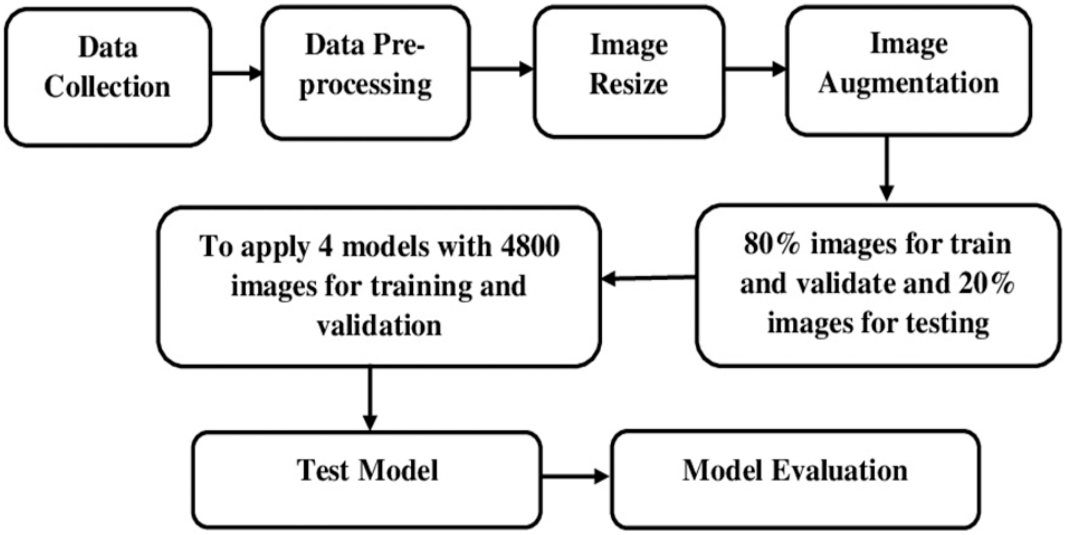

Data Collection
Data Pre-processing
Image Resize
Image Augmentation
80% images for train and validate and 20% images for testing
To apply 4 models with 4800 images for training and validation
Test Model
Model Evaluation

Figure 3.1: The system architecture of our experiment.

3.2 Research Subject and Instrumentation

In this paper, we have used transfer learning [12] technique to retrain the CNN [5] models. We have used TensorFlow [11] and Keras [13] in the backend and also used Convolutional neural Networks [5] layer.

3.2.1 TensorFlow

TensorFlow [11] is an open source software library which is tailored for numerical computation. It was developed by researchers and engineers of Google. It is useful for numerical computation with mathematical expressional using data flow graphs. In the graph, the nodes represent mathematical operations when the edges represent the multidimensional data arrays. TensorFlow [11] makes machine learning faster and easier. TensorFlow [11] together combines machine learning and deep learning bundles and algorithms together and makes them useful by a common metaphor. TensorFlow [11] reskill offers comprehensive tutorials start the initial level for new categories through transfer learning.

3.2.2 Transfer Learning

In this paper, we have used transfer learning [12] technique to retrain the CNN [5] models. Transfer learning [12] is a research problem which is focused on solving a problem and saving knowledge to be applied to a different but related problem. It is recently very famous in deep learning because it enables you to train deep neural networks with relatively little data. For example, we can use the knowledge learned for bike problem from the motorcycle problem. For this, we have to need a little data to train and get a rich accuracy within a short time compared with the traditional neural network.

3.2.3 Keras

Keras [13] is an Application Programming Interface of high-level neural networks which is written in Python. It is capable to run on top of TensorFlow [11], CNTK, or Theano. It was developed with a focus on enabling quick test. It permits for easy and quick prototyping and supports both convolutional and recurrent neural networks as well as two adjustments. It runs on the CPU and GPU. In order to work smoothly for image and text data, Keras [13] has numerous applications for common use of neural network building blocks such as layers, intentions, activation functions, optimizations, and a host of tools.

3.2.4 Convolutional Neural Network Layers

A CNN [5] is designed for reducing processing requirements that use a system like a multilayer perceptron. A CNN [5] layer has an input level, an output layer, and a hidden layer that includes multiple convolutional layers, pooling layers, activation layers, and fully connected layers or dense layers.

i. Convolutional Layer:

A convolution layer consists of neurons that are attached to the output of the input images or the output of the previous layers. When scanning through an image, the layer teaches localization of this region. When creating a layer using the Convolution2dLayer function. FilterSize, NumFilters, Name, Value are the parameters those we have sent in Convolution2dLayer function. Here FilterSize consists of Height and width of filters. And NumFilters consists of Number of filters.

For each region, the train network function calculates the weight and input of a point product and then adds the word of a bias. A set of weight applied to a region of the image is called a filter. Repeat the same calculation for each region of the filter, vertically and horizontally move along the input image. In other words, filter input convolves.

ii. Pooling Layer:

A max pooling layer divides the input into rectangular pooling areas and performs maximum computing and down-sampling in each region. For creating a max pooling layer, we have used maxPooling2dLayer function. PoolSize, Name, Value are the parameters of the maxPooling2dLayer function.

For removing the number of connections of the following layers, pooling layers follow the convolutional layers for down-sampling. They do not teach themselves, but they reduce the number of parameters to learn in the following levels. They help reduce overfitting [16]. A maximum pooling layer provides the maximum value of its input rectangular region. The size of the rectangular zone is determined by the basket of max pooling Layer. “Figure 3.2” represents a max pooling layer.

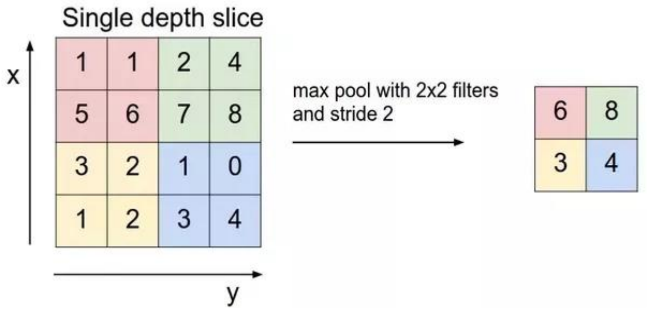


Figure 3.2: Max pooling layer.

iii. Activation Layer:

The activation level controls how the signal flows from one level to another, imitating how neurons are fired in our brain. The output signal that is strongly connected to older references, will enable more neurons, enable more prominent signal recognition for detection.

CNN is compatible with several complex activation functions like Rectified Linear Unit (ReLU), which is optimized for its fast training speed. A ReLU layer performs a threshold operation on each component of the input, where no value less than zero is set to zero. This operation is equivalent to

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The ReLU layer does not change its input size.

iv. Fully Connected Layer:

Completely connected to the latest layer of the network, which means that the neurons of the previous layers are connected to the next layer of each neuron. This is considered to be possible all the way from the output to inputs where imitating the high-level logic. The convolutional layers are followed by one or more fully connected layers. To create a fully connected layer using fullyConnectedLayer function. OutputSize,Name,Value are the parameters of fullyConnectedLayer function. “Figure 3.3” represents a fully connected layer.

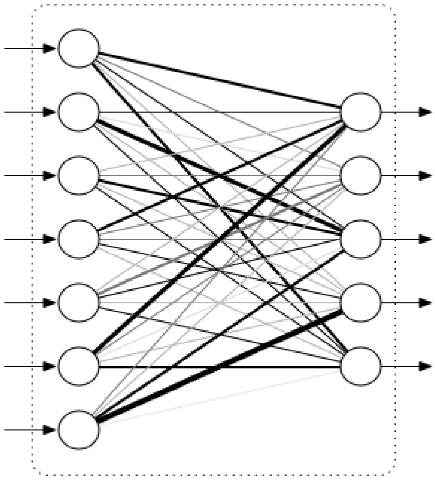


Figure 3.3: Fully connected layer.

Convolutional neural network [5] uses multilayer convolution to extract features and use much more parameters than depthwise separable convolution [17]. CNN [5] uses 29.3 million parameters but depth wise separable convolution [17] uses only 4.2 million parameters and it gives a high accuracy within a short time. Depthwise separable convolution [17] has been shown to be a successful model used for classification, in both cases, the number of parameters required to calculate previously available parameters, and to significantly reduce the number of parameters required to perform at the given level, for calculating the parameters available.

3.3 Data Collection Procedure

In our experiment, we needed a lot of images of Bangladeshi local fish. For that, we have searched on the internet for images. But unfortunately, we didn’t find the actual images of local fish. Almost fish of the internet are marine fish but we needed the local fish. On account of this, we went to a big fish market which is located into the Brahmanbaria District for collecting local fish of Bangladesh. Then we have collected 1000 real images of 10 classes which are Baim, Bacha, Bele, Chitol, Gutum, Kakila, Koi, Punti, Taki and Tengra. Each class has 100 images. It is very difficult to show all of the data. Therefore, we have given 3 images from each class of our dataset. The following “Figure 3.4” represents a sample dataset of our experiment.

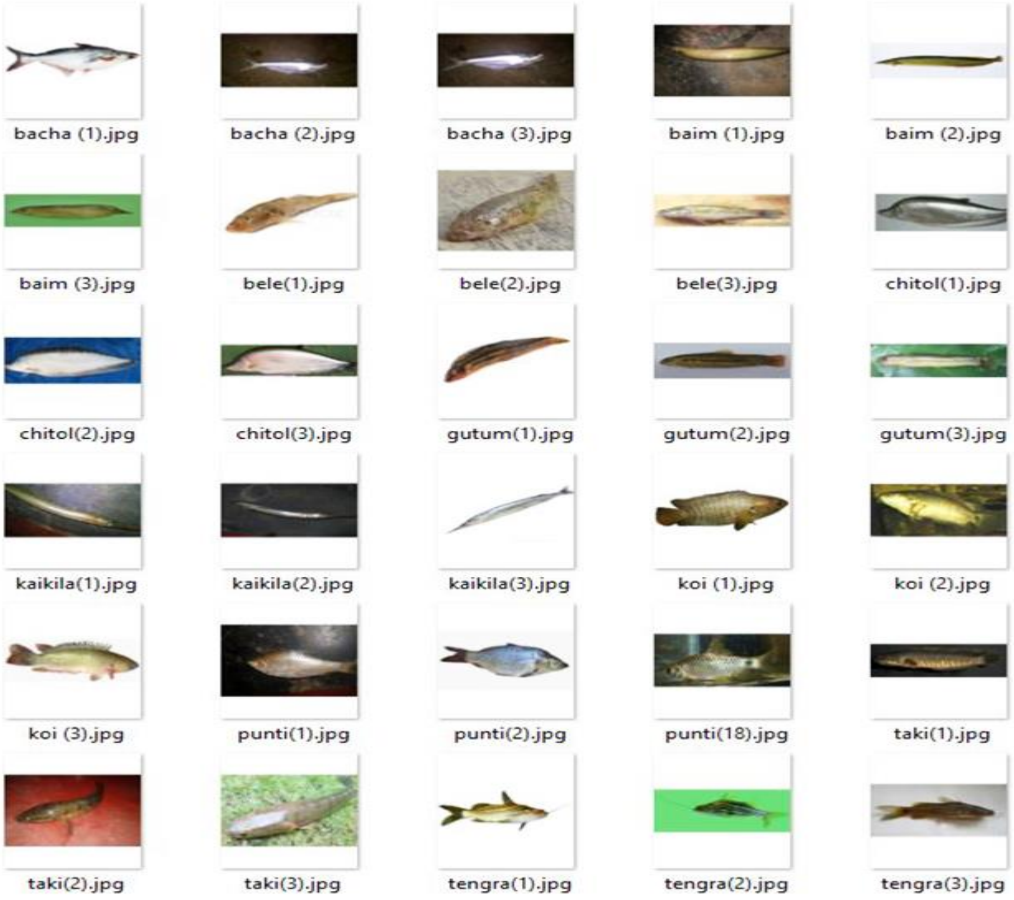


Figure 3.4: The example of the Local Fish BD Dataset of Bangladesh.

3.4 Feature Description

For our experiment, we need a lot of data indeed. But it is very difficult to collect a lot of data within a short time. But it's really true that we are facing an overfitting [16] problem for fewer data. On the other hand, large data sets can produce more accurate models, but they can extend processing time. Augmented data means when we create new information based on our existing data changes. In our experiment, we have artificially increased our dataset using augmented methods to avoid overfitting [16]. All of our images weren't an accurate size that's why we have resized our data. Then we have applied 5 different augmented methods and they are: rotate right +30 degree, rotate left -30 degree, flip horizontally, shading and translation. Rotation means move or move an axis or center round. Flip means to turn or suddenly turn with a rapid movement. Shading means an illustration of color or color is a block dark. And finally translation means a new position in a translation moves all points of an object along the same straight line path.

3.5 Implementation Requirements

3.5.1 Inception-V3

Inception-V3 [1] [14] is one of a pretrained model of TensorFlow. It is a reconsideration for the basic structure of Inception-VI, Inception-V2, in 2015. The Invention-V3 [1] [14] model is trained on the ImageNet that can detect 1000 classes in the dataset which top-5 error rate is 3.5%, error top rate I dropped 17.3%. The model is composed of a basic unit called "Inception Cell" in which it is performed a series of conglomerates at different stages and then collect the results later. To conserve the count, 1x1 convolutions are used to reduce input channel depth. For each cell, a set of 1x1, 3x3, and 5x5 filters that can learn to input features from different scales. The highest pooling is also to save the level with the "same" padding so that the output can be properly appended.

The researchers published a follow-up paper that revealed more effective alternatives to the original counting cells. Conventions with large local filters (such as 5x5 or 7x7) are useful for their specification and ability to extract features on a larger scale, but calculations are equally costly. Researchers said that the 5x5 convolution can be presented more economically by three more stacked 3x3 filters. It has been shown that 3x3 convolutions can be deconstructed further in 3x1 and 1x3 convolution. “Figure 3.5” represents the architecture of Inception-V3 [1] [14] model.

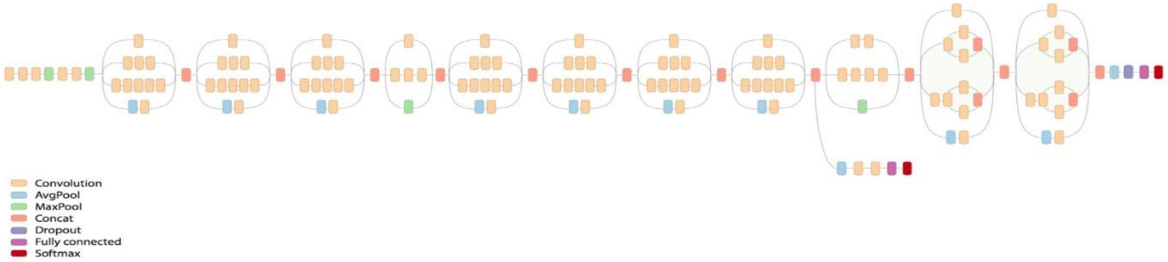


Figure 3.5: The architecture of Inception-V3 model.

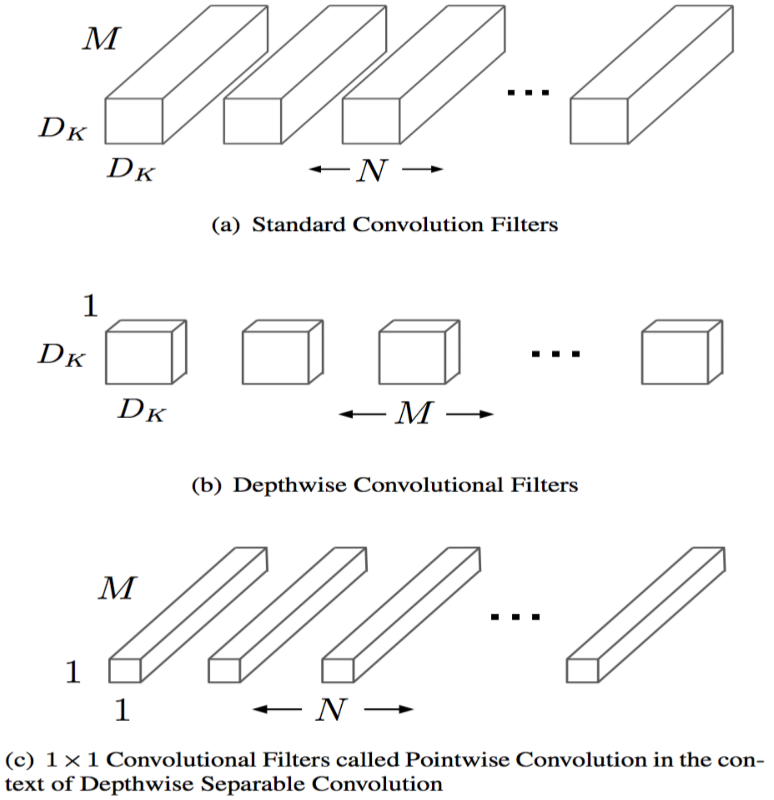
3.5.2 MobileNet

The MobileNet [9] model is a model which is based on depthwise separable convolutions [17] and it is a form of factored convolution that factors a standard resolution in depthwise separable convolutions [17]. And a 1 x 1 convolution is called a pointwise convolution. The depthwise convolution [17] applies a single filer to each input channel for MobileNets [9]. The pointwise convolution then applies 1x1 convolution to combine the output of deepwise. Deeply separable convolution divided into this two layers, a separate layer for filtering and a dividing layer for melting. This factorization has the effect of widely counting and reducing the size of the model. “Figure 3.6” displays how a depthwise convolution [17] 6(b) and a 1 x 1 pointwise convolution 6(c) is manufactured by a standard convolution.

For a property map of F, the standard convolution receives D_F x D_F x M as an input and for a property map of G it manufactures D_G x D_G x N. Here $D_F$is the height and width of a spatial input feature map, M is the input channel number, D_G is the height and width of a spatial output feature map and N is the output channel number. Convolution kernel K which size is D_K x D_K x M x N is parameterized to the standard convolutional layer. Here D_K is the spatial dimension of the kernel is considered to be class and M and N are already defined. The calculation expense of standard convolution:

$$D_K . D_K . \mathrm{M. N.} D_F . D_F$$

The calculation expense of Depthwise convolution:



(a) Standard Convolution Filters

(b) Depthwise Convolutional Filters

(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 3.6: The (a) standard convolutional strainers are replaced by two layers: (b) depthwise convolution and (c) pointwise convolution to make a depthwise separable strainer.

$$D_K . D_K . \mathrm{M. N.} D_F . D_F$$

And finally, Depthwise separable convolutions expense:

$$D_K \; . \; D_K \; . \; \mathrm{M}. \; D_F \; . \; D_F + \mathrm{M}. \; \mathrm{N}. \; D_F \; . \; D_F$$

It is the sum of the depthwise and 1 x 1 pointwise convolutions. MobileNet [9] model is the address of each of these terms and their interaction. First, it uses deeply divisible solvents to break the interaction between the number of output channels and the size of the kernel. MobileNet [9] is a deeply divided split using 3 x 3, which is 8 to 9 times less than the standard calculation.

3.5.3 ResNet

Researchers have already mathematically proven that deeper neural networks have more representational power just for its identity block or shortcut connections. The plain network is mostly 3 to 3 filters and there follow the two general design rules: (i) for the same output features map size, level has the same filter number; And (ii) the size of the map of the feature map, number the time filter is double as the filter is doubled per layer. The level of conflict is that there is a stride 2. Networks end with a worldwide average pooling layer and 1000-way fully connected layer with softmax. Total number weighing level is 34.

The residual network or Resnet [6] [15] is based on the plain network. But in Resnet [6] [15], shortcut connections are inserted (Figure 3.7 bottom) which turn it on the remaining version of its reflection network. Identity shortcut can be used directly the same level output (Figure 3.7). When the level is increased (dot-line shortcuts in Figure 3.7), two options are considered: (A) the shortcut still conducts a measuring map, additional zero entries are padded for an additional level. This option introduces any additional parameters. (B) The projection shortcut matches the measurement for both options when the shortcuts are added to the map of the two-dimensional features, then it starts at the beginning of 2.

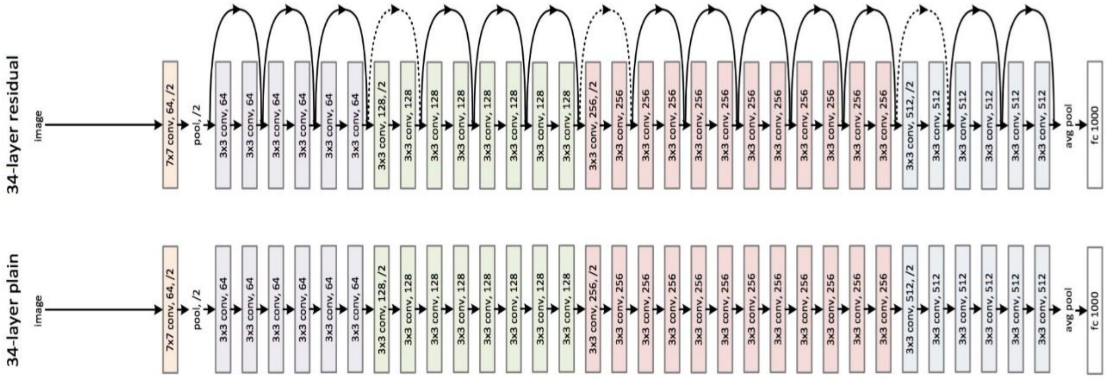


Figure 3.7: Top: 34 parameter layers of a plain network, Bottom: 34 parameter layers of a residual network.

Generally, in the plain network, the input matrix calculates two linear conversions with the RELU activation function. In a residual network, it copies the direct input matrix to the second conversion output and calculates the output in the final RELU function.

The activation of a layer in a plain network is defined as: y = f(x).

Where f (x) is the convolution, matrix quality or normalization of batch etc. When the signal is postponed, the gradient will always pass through f (x), which can cause problems due to the associated nonlinearities. Instead, apply ResNet [6] [15] to each layer: y = f(x) + x

The end "+ x" is a shortcut. It allows graduates to directly pursue chasing. By stacking this level, the gradient can theoretically "skip" all the intermediate levels and reach the bottom without decreasing. “Figure 3.8” represents the shortcut connection of ResNet.

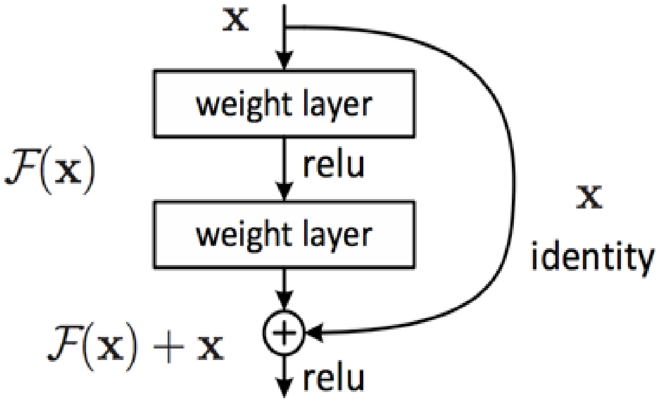


Figure 3.8: Residual connection block.

So we can say that in a Residual Network [6] [15], to add many remaining residual blocks together. The plain very deep network of flat gradually increases its energy loss, error after a certain depth. Nevertheless, ResNet [6] [15] does not face this problem.

3.5.4 Xception

The Xception [10] was proposed F. Chollet and the manufacturer Chief Executive Officer of the Keras Library in 2016. The author sees the Xception [10] model as an "extreme version" model with a maximum big tower. The introduction is first proposed by the Architecture C. It has four towers, a simple 1 × 1 convention layers, then there are towers with 3 × 3 and 5 × 5 convolutions the following 1 × 1 convention, and finally the average pool tower. In the end, the results are concatenated.

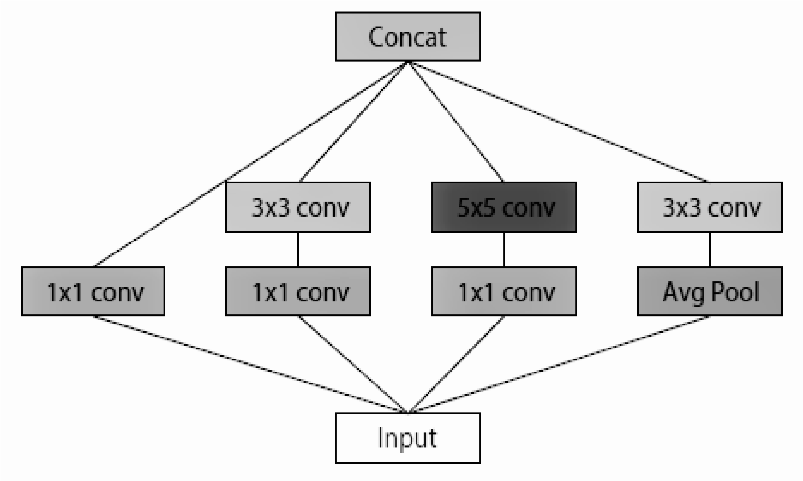


Figure 3.9: Example of Inception module.

Full details of network features are given in “Figure 3.10”. The Xception [10] architecture has 36 convolution layers that create network feature extraction centers. In our experimental evaluation, we will only verify image categories and so our traitor base will be followed by a logical response level. Formally attached layers can be entered officially before the Logical Regression Layer, which is searched in the experimental assessment section. There are 14 modules of 36 synthesizer levels, out of which there are linear residual connections around the first and the last modules.

Xception [10] model is extremely successful extreme version types start model. It has the maximum number of 9 towers in the “Figure 3.9”, the number of 1 × 1 filter is the greatest one. In other words, the expression module reaches 1 × 1 Cross-channel correlation mapped and then map the spatial relationship of each output separately channel. The planned presentation of the Xception [10] model is shown in the following diagram 10. Main Building-Block exception modules (ReLU + SeparableConv) and remaining Connection around them. The model supports images larger than 70 × 70 sizes. In case of classification Task, the final 2048-dimensional vector is followed dropout (Optional) and dense category classification.

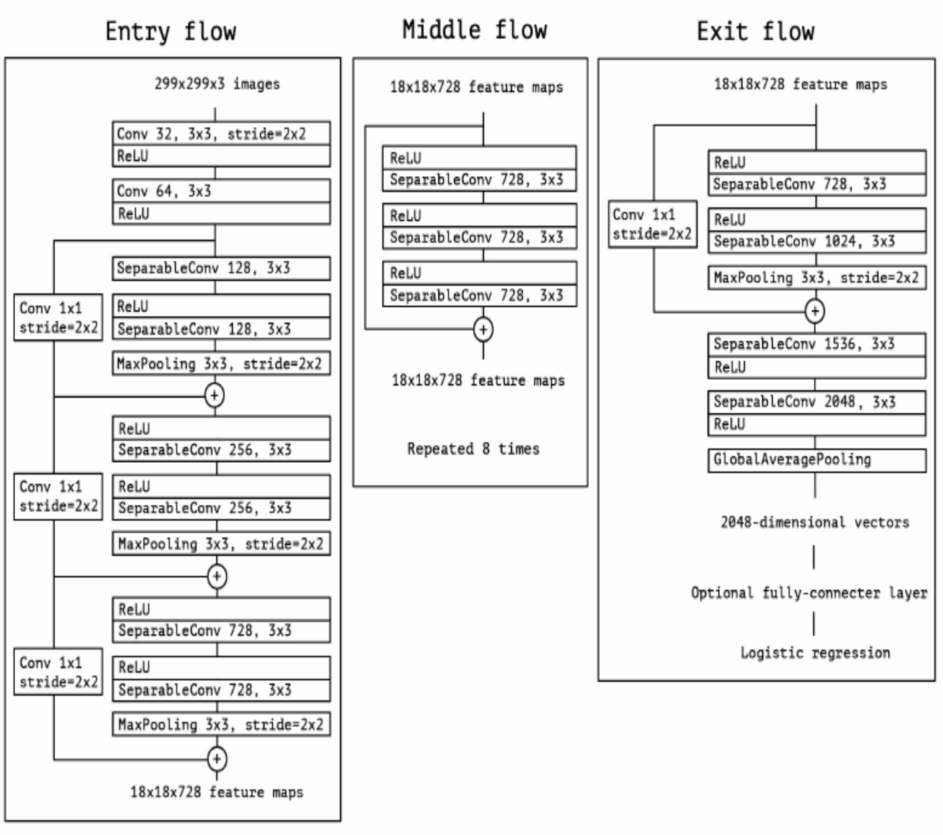


Figure 3.10: The Xception model architecture.

Machine learning and especially the problem of statistical classification, a confusion matrix, also known as an error matrix. A Confusion Matrix is a table that is often used to describe the performance of a classification model in test data for which the actual values are known. A confusion matrix for a binary, such as the number of false positive numbers (FPs), false negative (FNs), true positive (TPs), and true negative (TNs) of 2-class problems. In the case of multiple class, such as more than 2-class problem, there can be confusion matrix n × n (n > 2) levels. It contains n rows, n columns and total n × n entries in confusion matrix. From these matrices, the number of FPs, FNs, TPs, and TNs cannot be counted directly. According to this method, classes for FPs, FNs, TPs, and TN are calculated for the class:

$$TP_i = a_{ii}.$$

$$FP_i = \sum_{j=1, j\neq i}^{n} a_{ji}..$$

$$FN_i = \sum_{j=1, j\neq i}^{n} a_{ij}..$$

$$TN_i = \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq i}^{n} a_{jk}.$$

We have made Confusion Matrix for each model. By using, the Confusion Matrix, we have calculated Accuracy, Precision, Recall, F1- Score, Sensitivity or False positive rate (FPR), Specificity, and False negative rate (FNR). Accuracy is the most intuitive performance measurement and it is simply the ratio of forecast observation accurately to total observations.

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

Precision is the predictive positive observation of predicted positive observations.

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

A recall of all the genuine categories of accuracy accurately predicted the positive monitoring ratio.

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

The F1 scores are the weighted and the weighted average of the recall.

$$\text{F1 scores} = 2 * \frac{Recall * Precision}{Recall + Precision}$$

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

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

$$\text{FPR} = 1\text{- Specificity}$$

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

CHAPTER 4

EXPERIMENTAL EVALUATION

4.1 Introduction

Deep networks need a large amount of training information to achieve better performance. To create a strong image classifier using very little training data, image addition is usually needed to increase the performance of deep networks. The image combination creates an image of artificially processing methods by combining various processes or a combination of multiple processes, such as random rotation, transit, shear, and flip etc. But we have used 5 augmented methods which we already described in feature description point. In our experiment, our models need a fixed pixel for all data images. That’s why we have resized all of the data in a fixed resolution which is 300 x 300 pixels. After augmentation, we have gotten total 6000 images indeed. Now each class has 600 images. For testing, we have used 1200 images and 4800 images used for training and validation. We are given a name of our dataset which name is “Local Fish BD Dataset”.

4.2 CNN Models Implementation

In a neural network, neurons are organized into the layer. Different layers can perform different conversions in their inputs. In Inception-V3 [1] [14], the signals travel from the first level (input) to the last one (output), alternatively, the layers often travel after traveling. As the last hidden layer, the "bottleneck" provides enough brief information to provide the actual classification work to the next level. In the retrain.py script, we removed the old top layer and we handle a new train in the downloaded images. In the preprocessing step, there are 10 unique labels in our dataset and we have only 1000 training images, we have to increase the data to prevent our models from over-fitting. Then we have retrained the bottleneck and fine-tuning the model. The retrain script is the key component of our algorithm and any custom image classification that uses the intention of learning to transfer. It was designed by TensorFlow [11] writers for this particular purpose.

Keeping the depth and breadth of balance, the networks are carefully prepared, the maximum data flow into the network. Before each pooling, increase the feature map. When the depth is increased, the number or height of the level increases systemically. Before the next level, using the increase in the coordinates of the properties increase the width of each level. If we were to filter 5x5 and 7x7 by splitting multiple 3x3, we only used 3x3 convolution. In Inception-V3, we have used 21 million parameters, 48 layers, 3 input channels and the image size was 299 x 299. In the end, the final graph “Figure 4.1” of Inception-V3 [1] [14] is generated in TensorBoard. It is very difficult to train this model in a low figured computer that’s why it takes 8 hours to run. Then we test our images.

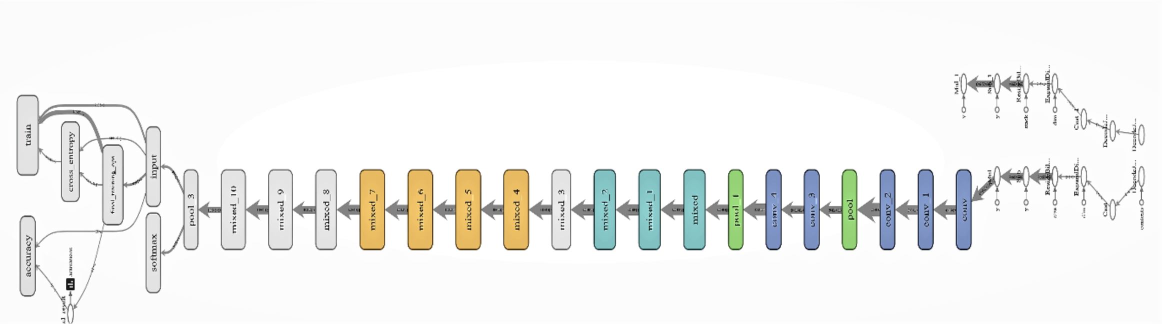


Figure 4.1: The main graph of Inception-V3.

MobileNet [9] has many versions. We have used 224 1.0 version. Here 224 is the input image resolution and 1.0 is the relative model size. The higher resolution images may take longer but there is a possibility of providing better classification accuracy. Since we are only training the final level and our datasets are not very big, that’s why we have kept this value as 224. If the size of the model is bigger, then we can get a more accurate result. For this purpose, we have taken 1.0 model size. If the image resolution and model size is small then our model accuracy will be lower. Here, we have taken the image size is 224 x 224. Then we have run the retrain python script which is provided by TensorFlow [11]. We have to command set up the command directory paths for the bottleneck, model and summary files where our image is stored. We have applied Batch Normalization (BN) and ReLU after each convolutional (Conv) layer.

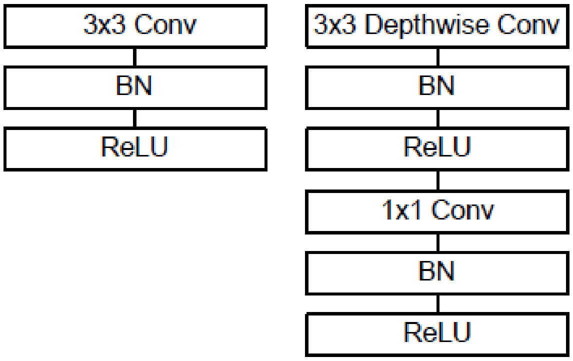


Figure 4.2: Left: Standard Convolution, Right: Depthwise separable convolution.

In this model, we have used only 4.24 million parameters, 28 layers, and 3 input channels. MobileNet [9] is small in size with so much faster than the other three models. It takes only 4 hours to run. If we use Conv MobileNet [9] then we have to be used 29.3 million parameters. Then it will be a longer process. At last, the final graph “Figure 4.3’ is generated from the TensorBoard and then we test our images for MobileNet [9].

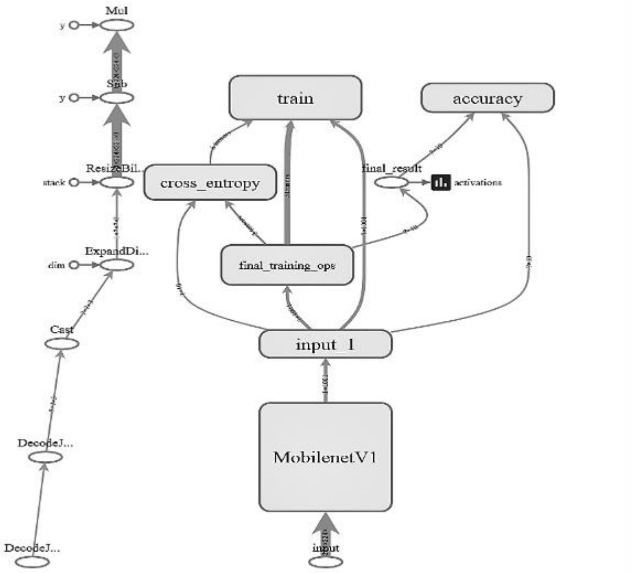


Figure 4.3: The main graph of MobileNet.

The Convolutional Neural networks [5] have Conv layers, fully connected layers for classification like Inception-V3 [1] [14], ResNeXt [7], DenseNet [8] and so on, without any shortcut connections. That's why CNN [5] is called a plain network. When plain networks go deeper, the problem of vanishing gradient occurs. To solve the problem of vanishing gradient, a skip or shortcut connection is added to add output x after some weight level as "Figure 4.4". Even if there is an invisible gradient for weight layers, there is always an identity x to move the previous layers in ResNet [6] [15]. There are 3 types of shortcut connections when input levels are smaller than the output dimensions. (A) Performs shortcut measurement mapping, with extra zero padding for increased magnitude. So, no additional parameters. (B) Projection Shortcuts are used only to increase levels, other shortcut identities. Additional parameters are required. (C) All shortcuts are approximate, more than the additional parameters (B).

In our experiment, we have used ResNet50 [6] [15] which has 50 layers of the residual network. We have conducted this because we don't face vanishing gradient problem due to skip connections. For the ResNet50[6] [15] model, we only substitute the remaining block with three layers of bottlenecks which reduce the 1x1 conventions and later restore channel depth, allow for the less computational load when calculating the 3x3 convolution. The ResNet50 [6] [15] can be seen as both parallel and serial modules, just thinking of inputs as multiple modules in parallel, each module outputs connected to the series.

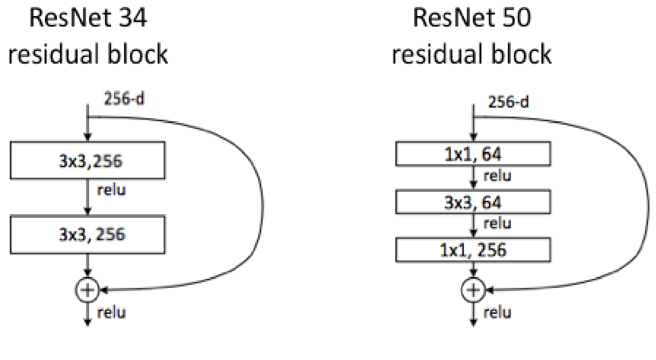


Figure 4.4: Left: ResNet34 residual block, Right: ResNet50 residual block.

ResNet50 [6] [15] with 50 layers which started to use a bottleneck layer like above. This layer reduces the number of features in each layer using a 1x1 resolution with a small output (typically input 1/4) and then drops to 3x3 levels and then again, a 1x1 resolution reduces to a larger number of features. For Inception-V3 [1] [14] modules, it allows keeping the countdown when providing a rich combination of features. Just for this bottleneck design “Figure 4.4”, 34-layer ResNet has become 50-layer ResNet.

In this model, the image size was 224 x 224. We have used 23.5 million parameters and 3 input channels. It takes almost 38 hours to run which was a huge longer process because of many layers. Then we test our images for ResNet50 [6] [15].

Xception [10] stands for an extreme version of Inception which is created by Google with a modified depthwise separable convolution network [17]. Xception architecture is a linear stack of depthwise separable convolution layer with the remaining connection. It makes architecture very easy to define and correct. By using a high-level library such as Kersh [13] or Tensorflow [11], it only takes 30 to 40 line codes, in contrast to architecture such as the VGG-16 [4], rather than the Inception V2 architecture or V3 [1] [14] which defines a lot of complex Keras [13] and TensorFlow [11] are supplied as a part of the Kears [13] Application module, under the MIT license of an Open Source implementation. The Xception [10] model is only available for the Tensorflow [11] backend, so the Theano or CNTK will not working in the backend. We have used OpenCV which is a python library for image reading, Numpy for data preparation & preprocessing.

In Inception [1] [14], we started two little separations. We used the 1x1 resolution to input the original inputs in different, small input spaces, and the main input to each input spaces, we used different types of filter to convert those small 3D block data. Xception [10] takes one step forward. Rather than splitting input data into different narrow areas, it maps to local relationships separately in each output channel and 1x1 performs the convolution deeply to capture cross-channel relationships. It is basically equivalent to an existing activity known as a " depthwise separable convolution", which is depthwise convolution by a pointwise convolution. We can first consider it as a search of correlation between 2D spaces, then find a correlation in 1D space. Intuitively, this 2D + 1D mapping is easier to learn than a complete 3D mapping in “Figure 4.5”.

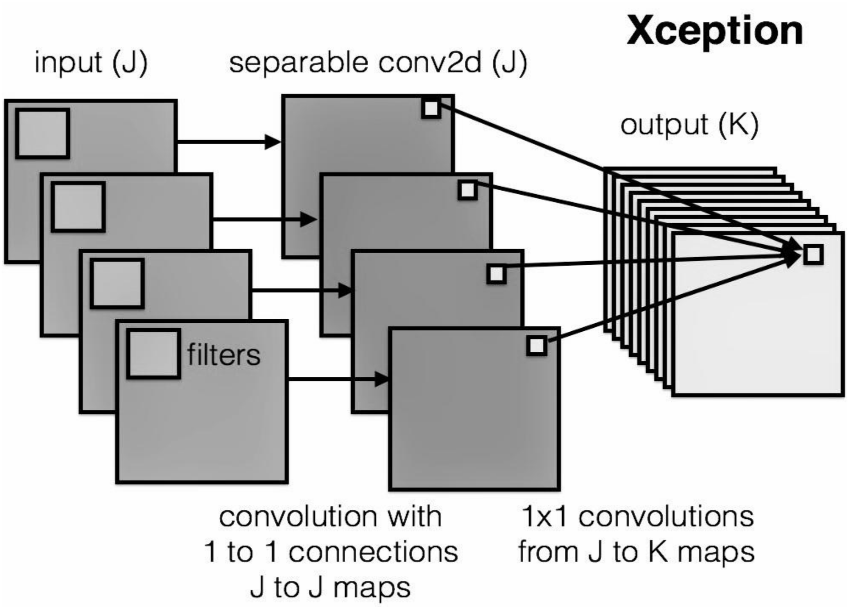


Figure 4.5: The cross-channel correlation of Xception.

The input image size was 299x299 in Xception [10]. We have used conv layers, separable conv layers, max-pooling layers, global and average max-pooling layers, and the total parameter is 22.9 million with 3 input channels. There are 14 modules in 36 convolutional layers. Xception [10] is a marginally slower model than the others. It took almost 42 hours to run indeed. Then we test our images for Xception [10].

CHAPTER 5

RESULT ANALYSIS

5.1 Introduction

This chapter involves the graph analysis, multiclass confusion matrix, and discussion of the result of the study. This chapter has also some different tables and different graph of CNN [5] models.

5.2 Experimental Results

Here, Tables 5.1, 5.2, 5.3 and 5.4 are the Confusion Matrix of Inception-V3 [1] [14], MobileNet [9], ResNet50 [6] [15], and Xception [10] models respectively.

Table 5.3: The multiclass confusion matrix of Inception-V3.

Fish	Baim	Bacha	Bele	Chitol	Gutum	Kakila	Koi	Punti	Taki	Tengra
Baim	107	0	3	0	5	2	0	2	1	0
Bacha	0	109	0	5	1	3	0	2	0	0
Bele	0	2	108	0	4	1	3	0	2	0
Chitol	0	4	0	111	0	0	1	0	1	3
Gutum	1	1	5	0	109	2	1	0	0	1
Kakila	4	5	0	0	0	111	0	0	0	0
Koi	0	0	0	3	0	5	109	0	0	3
Punti	0	0	0	2	4	0	2	112	0	0
Taki	1	0	2	0	4	0	8	0	105	0
Tengra	0	10	0	1	4	0	1	0	0	104

Table 5.4: The multiclass Confusion Matrix of MobileNet.

Fish	Baim	Bacha	Bele	Chitol	Gutum	Kakila	Koi	Punti	Taki	Tengra
Baim	110	0	0	0	2	0	0	8	0	0
Bacha	0	110	0	0	0	5	0	0	0	5
Bele	0	2	108	0	4	1	3	0	2	0
Chitol	0	4	4	112	0	0	0	0	0	0
Gutum	1	1	5	0	109	2	1	0	0	1
Kakila	4	5	0	0	0	111	0	0	0	0
Koi	0	0	0	0	0	0	113	1	6	0
Punti	0	0	0	2	4	0	2	112	0	0
Taki	0	0	0	0	3	0	7	0	110	0
Tengra	0	4	0	4	0	0	0	0	0	112

Table 5.5: The multiclass Confusion Matrix of ResNet-50.

Fish	Baim	Bacha	Bele	Chitol	Gutum	Kakila	Koi	Punti	Taki	Tengra
Baim	105	0	7	0	7	0	0	1	0	0
Bacha	3	107	0	7	0	0	3	0	0	0
Bele	8	0	105	0	0	0	7	0	0	0
Chitol	0	0	5	111	0	0	4	0	0	0
Gutum	6	0	2	0	109	0	0	0	0	3
Kakila	0	2	0	0	4	106	0	8	0	0
Koi	0	0	0	3	0	0	107	0	10	0
Punti	4	4	0	2	0	8	0	102	0	0
Taki	0	0	0	0	3	0	8	0	106	3
Tengra	2	0	0	4	0	0	5	0	5	104

Table 5.6: The multiclass Confusion Matrix of Xception.

Fish	Baim	Bacha	Bele	Chitol	Gutum	Kakila	Koi	Punti	Taki	Tengra
Baim	89	1	0	11	9	2	3	0	3	2
Bacha	0	98	2	10	0	1	3	1	0	5
Bele	0	0	80	7	15	8	0	0	0	10
Chitol	0	0	0	100	0	3	7	2	0	8
Gutum	8	10	0	10	78	4	0	0	0	10
Kakila	6	0	0	8	14	89	0	0	0	3
Koi	0	0	1	2	0	0	102	0	7	8
Punti	0	6	0	8	0	0	3	102	0	1
Taki	1	4	3	2	4	0	4	0	99	3
Tengra	4	8	2	0	3	0	3	0	3	97

The following “Figure 5.1” and “Figure 5.2” display the performance of accuracy and cross entropy of Inception-V3 [1] [14] model which is generated from TensorBoard. Training accuracy is the percentage of images used in the current training batch were labeled in the right category. The Cross-entropy is loss function, which gives a glimpse of how well the learning process is working (lower number is good). We can see from “Figure 5.1” and “Figure 5.2”, the Inception-V3 [1] [14] model has done 5000 iterations.

In “Figure 5.1”, the orange line represents the accuracy of the model on the training data. When the blue line represents the accuracy on the validation set. And X-axis shows a function of training progress while Y-axis shows accuracy. If the training accuracy is increasing while decreasing the verification accuracy, the model is considered "overfitting". When the model starts to remember the training set instead of understanding the general pattern, it is overfitting [16]. So we can tell that our Inception-V3 [1] [14] model has happened a little overfitting [16].

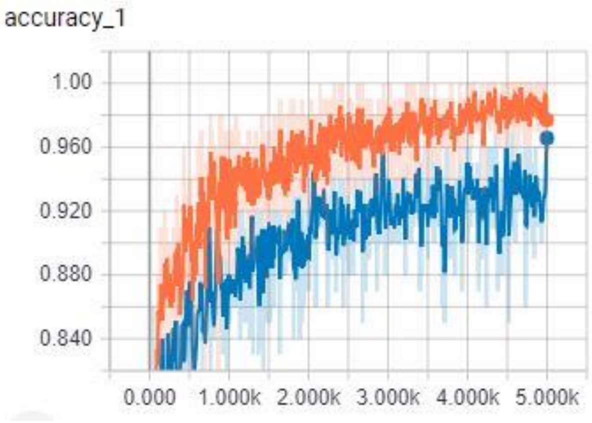


Figure 5.1: The variation of accuracy on Local Fish BD Dataset.

In “Figure 5.2”, the orange line represents the cross-entropy of the model on the training data. When the blue line represents the cross-entropy on the validation set. And X-axis shows a function of training progress while Y-axis shows cross-entropy.

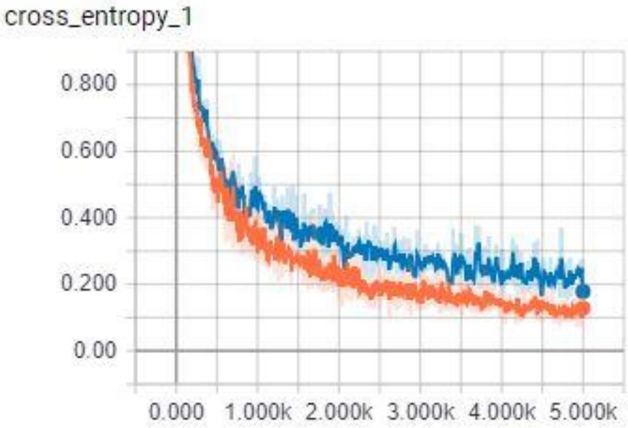


Figure 5.2: The variation of cross entropy on Local Fish BD Dataset.

Table 5.5 shows the description of the above two figures. For our Local Fish BD Dataset, the training accuracy can reach 98-100%, and the validation accuracy can be sustained at around 96%. On the other hand, the training cross-entropy is 0.10 and the validation cross-entropy is 0.15.

Table 5.7: Description of above two figures.

Dataset	Index	Performance
Dataset	The accuracy of training set	98-100%
	The accuracy of validation set	96%
	The cross entropy of training set	0.10
	The cross entropy of validation set	0.15

The following "Figure 5.3" and "Figure 5.4" are the accuracy and cross-entropy graph of MobileNet [9] which is also generated from TensorBoard. The MobileNet [9] model has also completed 5000 iterations. "Figure 5.3" and "Figure 5.4" are the same category graph like Inception-V3 [1] [14]. So, the blue line, the orange line, X-axis and Y-axis which thing those are represented, I have already described at the time of representation in Inception-V3 [1] [14].

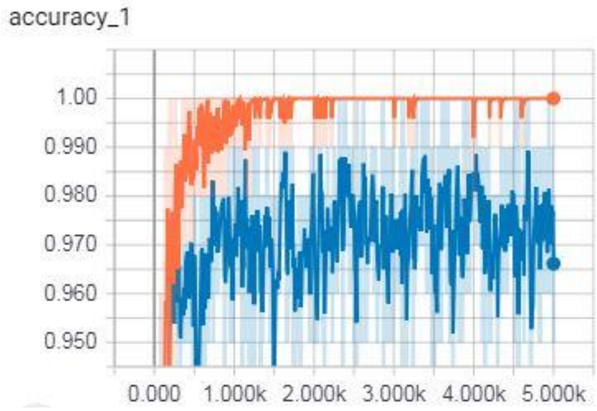


Figure 5.3: Accuracy as for training and validation of MobileNet.

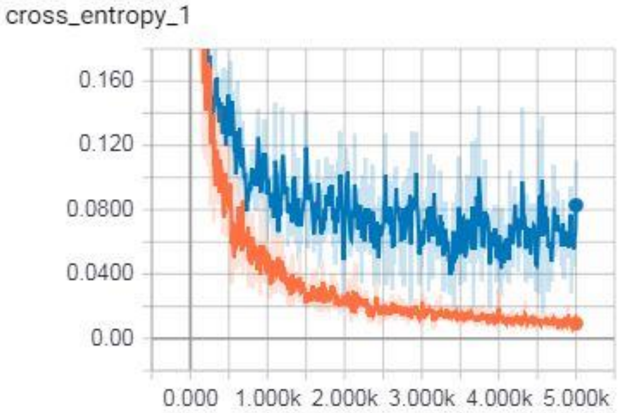


Figure 5.4: Cross entropy as for training and validation of MobileNet.

The following "Figure 5.5" is the training loss vs validation loss of ResNet50 [6] [15]. In "Figure 5.5", the blue line represents the performance of the training loss while the orange line represents the performance of the validation loss. And X-axis represents a number of epochs while Y-axis represents the loss. In ResNet50 [6] [15], we have used 20 epochs. From the following plot of losses, we can see that both the model and its validity datasets have comparable performance (labeled) performance. If these parallel plots start to leave consistently, it can be a mark of closing training in a previous era.

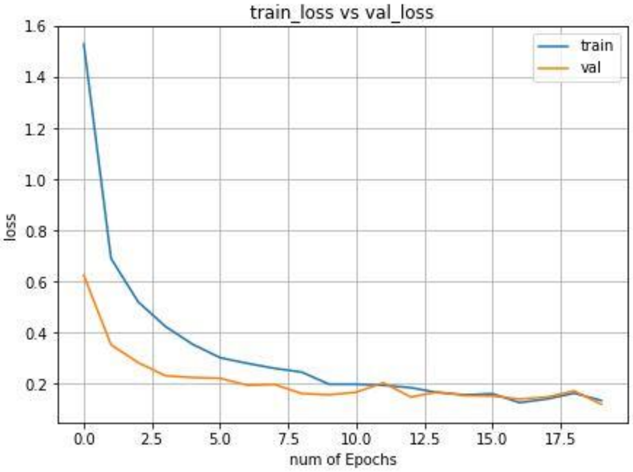


Figure 5.5: Training loss vs Validation loss curve of ResNet-50.

The following “Figure 5.6” is the training accuracy vs validation accuracy of ResNet50 [6] [15]. In Figure 5.6”, the blue line represents the performance of the training accuracy while the green line represents the performance of the validation accuracy. And X-axis represents a number of epochs while Y-axis represents the accuracy. From the plot of accuracy, we can see that the model may be slightly trained because the accuracy trend of both datasets is still growing for the past few decades. We can also see that the model is not yet able to learn more than the training dataset, which shows comparative skills in both datasets.

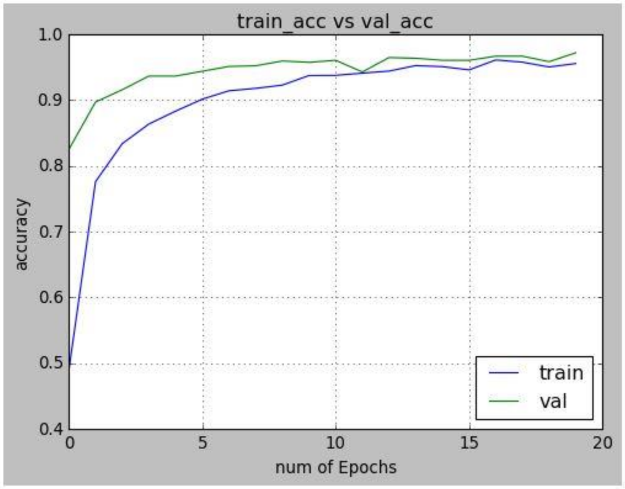


Figure 5.6: Training accuracy vs Validation accuracy curve of ResNet-50.

In Xception [10], we have used 10 epochs. The following “Figure 5.7” and “Figure 5.8” are the Training loss vs Validation loss and Training accuracy vs Validation accuracy graph of Xception [10]. “Figure 5.7” and “Figure 5.8” are the same category graph like ResNet50 [6] [15]. So the blue line, the orange line, the green line, X-axis and Y-axis which thing those are represented, I have already described at the time of representation in ResNet50 [6] [15].

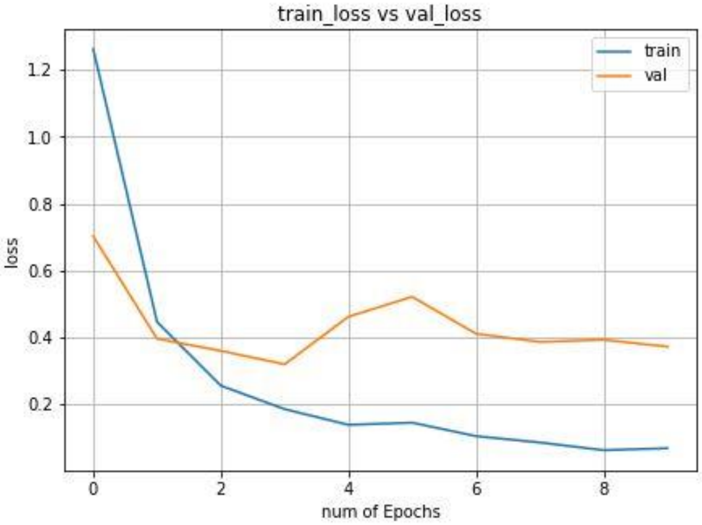


Figure 5.7: Training loss vs Validation loss curve of Xception.

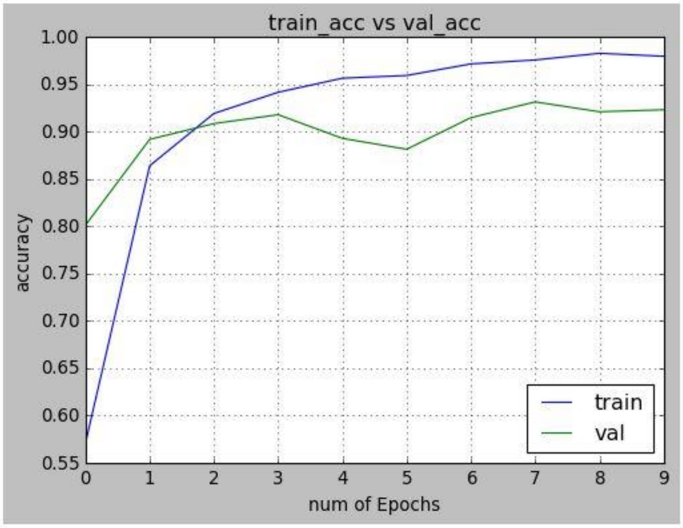


Figure 5.8: Training accuracy vs Validation accuracy curve of Xception.

Now here, Tables 5.6, 5.7, 5.8 and 5.9 represent the Precision, Recall, Accuracy and F1-score of Inception-V3 [1] [14], MobileNet [9], ResNet50 [6] [15], and Xception [10] models respectively. The calculation of Precision, Recall, Accuracy and F1-score, we have already described in Research Methodology part.

Table 5.8: The precision, recall, accuracy and f1-score for Inception-V3.

Fish	Precision	Recall	Accuracy	F1-Score
Baim	94.6%	89.1%	98.4%	91.7%
Bacha	83.2%	90.8%	97.3%	86.8%
Bele	91.5%	90.0%	98.1%	90.7%
Chitol	90.9%	92.5%	98.3%	91.6%
Gutum	83.2%	90.8%	97.3%	86.8%
Kakila	95.2%	92.5%	98.2%	90.9%
Koi	87.2%	90.8%	97.8%	88.9%
Punti	96.5%	93.3%	99.0%	94.8%
Taki	96.3%	87.5%	98.4%	91.6%
Tengra	93.6%	86.6%	98.1%	89.9%

Table 5.9: The precision, recall, accuracy and f1-score for MobileNet.

Fish	Precision	Recall	Accuracy	F1-Score
Baim	95.6%	91.6%	98.7%	93.5%
Bacha	87.3%	91.6%	97.8%	89.3%
Bele	92.3%	90.0%	98.2%	91.1%
Chitol	94.9%	93.3%	98.8%	94.0%
Gutum	89.3%	90.8%	98.0%	90.0%
Kakila	93.2%	92.5%	98.5%	92.8%
Koi	89.6%	94.1%	98.3%	91.7%
Punti	92.5%	93.3%	98.5%	92.8%
Taki	94.0%	91.6%	98.5%	92.7%
Tengra	94.9%	93.3%	98.8%	94.0%

Table 5.10: The precision, recall, accuracy and f1-score for ResNet-50.

Fish	Precision	Recall	Accuracy	F1-Score
Baim	82.0%	87.5%	96.8%	84.6%
Bacha	94.6%	89.1%	98.4%	91.7%
Bele	88.2%	87.5%	97.5%	87.8%
Chitol	87.4%	92.5%	97.9%	89.8%
Gutum	88.6%	90.8%	97.9%	89.6%
Kakila	92.9%	88.3%	98.1%	90.5%
Koi	79.8%	89.1%	96.6%	84.1%
Punti	91.8%	85.0%	97.7%	88.2%
Taki	87.6%	88.3%	97.5%	87.9%
Tengra	94.5%	86.6%	98.1%	90.3%

Table 5.11: The precision, recall, accuracy and f1-score for Xception.

Fish	Precision	Recall	Accuracy	F1-Score
Baim	82.4%	74.1%	95.8%	78.0%
Bacha	77.1%	81.6%	95.7%	79.2%
Bele	90.9%	66.7%	96.0%	76.9%
Chitol	63.2%	83.3%	93.5%	71.8%
Gutum	63.4%	65.0%	92.7%	64.1%
Kakila	83.1%	74.1%	95.9%	78.3%
Koi	81.6%	85.0%	96.5%	83.2%
Punti	97.1%	85.0%	98.2%	90.6%
Taki	88.3%	82.5%	97.1%	85.3%
Tengra	65.9%	80.8%	93.9%	72.5%

For the classification problem, we can rely on an AUC - ROC curve. When we have to evaluate or visualize the effectiveness of our multi-class classification problem, we use the AUC (area under curve) ROC (receiver operating properties) curve. This is one of the most important evaluation matrices to test the performance of any category model. It is also written as AUROC (area under the receiver operating character). AUC - ROC curve is a performance measurement for classified problems at different threshold settings. ROC represents a potential curve and measure of AUC degree or isolation. It tells how many models it is able to differentiate between classes. Higher AUC is well predicting as 0s and 1s as models.

Our ROC curve “Figure 5.9” is plotted with TPR (True Positive Rate) against the FPR (False Positive Rate) where the y-axis represents TPR and x-axis represents FPR.

A nice model has 1 nearby AUC, which means it's a good measure of isolation. A poor model has around 0 AUC, which means it is the worst measure of isolation. In fact, this means that the result of recipes It is predicted as 0s and 1s and 1s as 0s. And when AUC is 0.5 that means there is no class separation capability of the model. In our experiment, all the AUC value comes near to 1. Here, the pink, the purple, the turquoise, and the yellow lines represent the ROC curve of Inception-V3 [1] [14], Mobilenet [9], ResNet50 [6] [15] and Xception [10] respectively.

Sensitivity and specificity are inversely proportional to each other. When we decrease threshold, we achieve a more positive value so that it increases sensitivity and reduces specificity. Similarly, when we increase the threshold, we achieve more negative values so that we get higher specificity and less sensitivity. As we know FPR = 1 - specificity. So when we increase TPR, the FP increases, and vice versa.

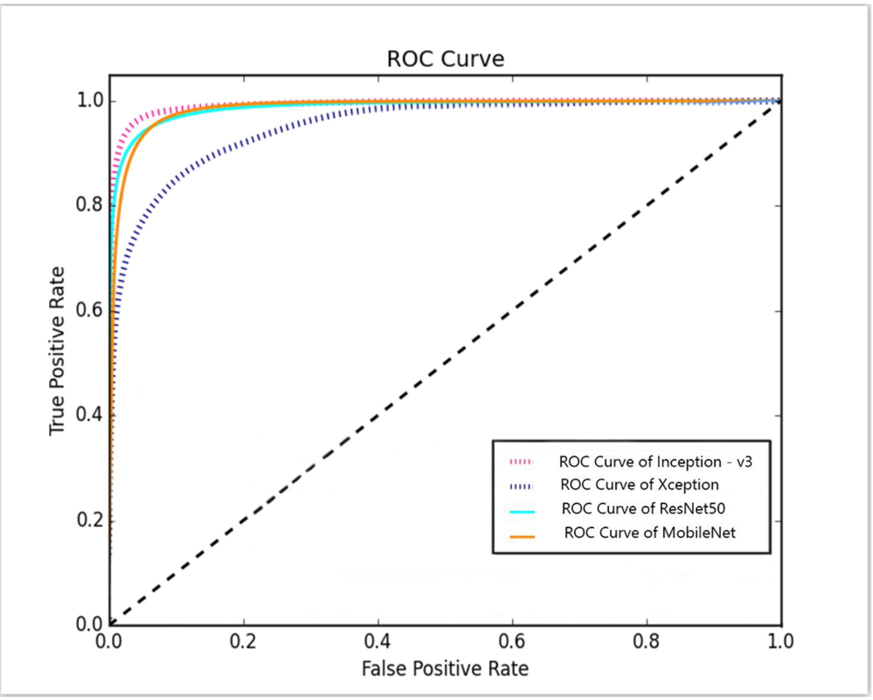


Figure 5.9: The AUC - ROC Curve for 4 CNN models.

4.3 Summary

To predict local fish with pictures, we study methods for various types of deep neural network architecture and learning supervision features. From the results we get from this study, it shows that the CNN [5] models have worked very well and given a higher accuracy on our dataset. And we try to give the graph of every model with a clear explanation. Hopefully, it will be benefited for people.

CHAPTER 6

FUTURE WORK AND CONCLUSION

6.1 Summary of the Study

In our paper, we have used Convolutional Neural Networks [5] model to identify the local fish of Bangladesh. We have conducted Inception-V3 [1] [14], Mobilenet [9], ResNet50 [6] [15], and Xception [10]. We have experimented these model with total 6000 images of 10 classes which are Baim, Bacha, Bele, Chitol, Gutum, Kakila, Koi, Punti, Taki, and Tengra and we have gotten high accuracy. From Inception-V3 [1] [14], Mobilenet [9], ResNet50 [6] [15] and Xception [10], we have gotten 98.07%, 98.41%, 97.65% and 95.53% respectively.

6.2 Limitation and Conclusion

For our experiment, we have used total 6000 images of local fish with 10 categories. We have used Convolutional Neural Network models, those are Inception-V3, MobileNet, ResNet50, and Xception and they have obtained a high accuracy of 98.07%, 98.41%, 97.65%, and 95.53% respectively on our dataset.

Though I have a small dataset, it has basically shown outstanding performance. Among the four models, the MobileNet [9] is done better than others because it is small in size and run faster with a high accuracy. On the other hand, the Xception [10] is performed with slow speed and also took a huge time. We have tried to provide a different experiment so that it will be benefited for the new generation people. Though it is very necessary so that people can easily recognize the local fish of Bangladesh in a wonderful way. Hopefully, it will be useful in the future.

6.3 Implication for Further Study

As we have used the pretrained model of Convolutional Neural Networks [5] which are Inception-V3 [1] [14], Mobilenet [9], ResNet50 [6] [15], and Xception [10]. Mathematically CNN [5] is a cross-relation rather than a conflict. From our experiment, we can say that the models of CNN [5] are performed very high and have also provided high accuracy. Though in Inception-V3 [1] [14] and MobileNet [9] have occurred a little overfitting [16] so we will try to remove the overfitting [16] in future using to add more data, more augmentation methods, regularization methods and also try to consume time complexity. We would like to make a model in future so that we can use that model and can get also a high accuracy like CNN [5] models. And we want to use a large data set of images to cover a wide range of local fish.

APPENDICES

Abbreviation:

CNN = Convolutional Neural Network

ROC = Receiver Operating Characteristic

AUC = Area under the ROC Curve

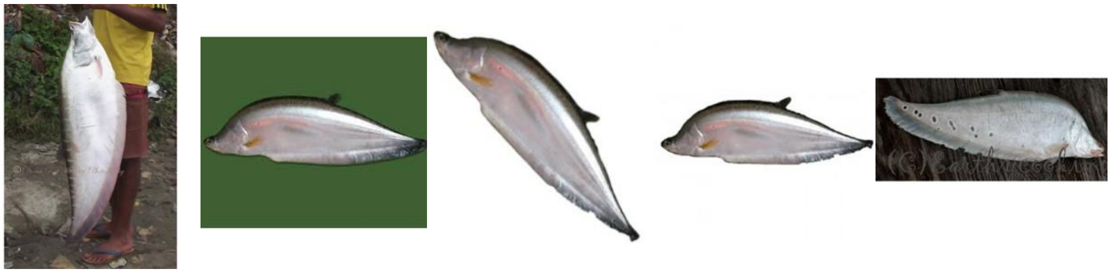
Sample of our Dataset:

Baim:

Bacha:

Bele:

Chitol:



Gutum:

Kakila:

Koi:

Punti:

Taki:

Tengra:

REFERENCES

1. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions", arXiv:1409.4842v1 [cs.CV] 17 Sep 2014.
2. Yann LeCun, Leon Botton, Yoshua Bengio, and Parteick Haffner, "Gradient –Based Learning Applied to Document Recognition", Proceedings of the IEEE (Volume: 86 , Issue: 11 , Nov 1998)
3. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Proceeding NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems
4. Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv:1409.1556v6 [cs.CV] 10 Apr 2015
5. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Proceeding NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems
6. Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, "Deep Residual Learning for Image Recognition", arXiv:1512.03385v1 [cs.CV] 10 Dec 2015
7. Saining Xie Ross Girshick Piotr Doll´ar Zhuowen Tu Kaiming He UC San Diego, "Aggregated Residual Transformations for Deep Neural Networks", arXiv:1611.05431v2 [cs.CV] 11 Apr 2017
8. Gao Huang, Zhuang Liu, Laurens van der Maaten and Kilian Q. Weinberger, "Densely Connected Convolutional Networks", arXiv:1608.06993v5 [cs.CV] 28 Jan 2018
9. Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv:1704.04861v1 [cs.CV] 17 Apr 2017
10. Franc¸ois Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions", correarXiv: 1610.02357v3 [cs.CV] 4 Apr 2017
11. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain, "TensorFlow: A System for Large-Scale Machine Learning", 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16).
12. Sinno Jialin Pan and Qiang Yang Fellow, "A Survey on Transfer Learning", IEEE Transactions on Knowledge and Data Engineering, 16 October 2009.

13. Vassili Kovalev, Alexander Kalinovsky, and Sergey Kovalev, “Deep Learning with Theano, Torch, Caffe, TensorFlow, and Deeplearning4J: Which One Is the Best in Speed and Accuracy?” Conference Paper · October 2016
14. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, and Jonathon Shlens, “Rethinking the Inception Architecture for Computer Vision”, arXiv:1512.00567v3 [cs.CV] 11 Dec 2015
15. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Identity Mappings in Deep Residual Networks”, arXiv:1603.05027v3 [cs.CV] 25 Jul 2016
16. Andrew Y. Ng, “Preventing “Overfitting’ of Cross-Validation data”, Proceeding ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning
17. Łukasz Kaiser, Aidan N. Gomez, and François Chollet, “Depthwise separable convolutions for neural machine translation”, Published as a conference paper at ICLR 2018
18. Sushil Kumar Mahapatra, Sumant Kumar, Sakuntala Mahapatra, Shuvendra Kumar, “A Proposed Multithreading Fuzzy C-Mean Algorithm for Detecting Underwater Fishes”, 2016 International Conference on Computational Intelligence and Networks
19. Xiu Li, Youhua Tang, Tingwei Gao, “Deep but Lightweight Neural Networks for Fish Detection”, OCEANS 2017 – Aberdeen
20. Nhat D. M. Nguyen, Kien N. Huynh, Nhan N. Vo, and Tuan Van Pham, “Fish Detection and Movement Tracking”, 2015 International Conference on Advanced Technologies for Communications (ATC)
21. Lars M. Wolff and Sabah Badri-Hoeher, “Imaging Sonar-Based Fish Detection in Shallow Waters”, 2015 International Conference on Advanced Technologies for Communications (ATC)
22. R. Prados, R. García, N. Gracias, L. Neumann, and Håvard Vågstøl, “Real-time Fish Detection in Trawl Nets”, OCEANS 2017 – Aberdeen
23. Sukhpal Kaur and Dr. Rajinder Singh, “Review On Jelly Fish Detection and Prevention Schemes in MANETS”, IJARIIT
24. M. Chambah, D. Semani, A. Renouf, P. Courtellemont, A. Rizzi, “Underwater Color Constancy: Enhancement of Automatic Live Fish Recognition”, 16th Annual symposium on electronic imaging, 2004, Inconnue, United States. Pp.157-168, 2004. <Hal-00263734>
25. Minsung Sung and Son-Cheol Yu, Yogesh Girdhar, “Vision based Real-time Fish Detection Using Convolutional Neural Network”, OCEANS 2017 – Aberdeen.