BANGLADESHI INDIGENOUS FISH CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

BY

Krishno Dey
ID: 171-15-9417

Md. Mustahid Hassan
ID: 171-15-9388

AND

Md. Masud Rana
ID: 171-15-9413

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

Supervised By

Ms. Most. Hasna Hena
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2021

APPROVAL

This Project titled **"Bangladeshi Indigenous Fish Classification using Convolutional Neural Networks**", submitted by **Krishno Dey, ID No. 171-15-9417**, **Md. Mustahid Hassan, ID No. 171-15-9388** and **Md. Masud Rana, ID No. 171-15-9413** 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 (BSc) and approved as to its style and contents. The presentation has been held on January 31, 2021.

BOARD OF EXAMINERS

____

Prof. Dr. Touhid Bhuiyan **Chairman**
Professor and Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

____

Moushumi Zaman Bonny **Internal Examiner**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

____

Md. Sazzadur Ahamed **Internal Examiner**
Senior Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

____

Dr. Md. Arshad Ali **External Examiner**
Associate Professor
Department of CSE
Faculty of Computer Science and Engineering
Mohammad Danesh Science & Technology University, Dinajpur.

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ms. Most. Hasna Hena,** Assistant Professor**, 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:

Most. Hasna Hena
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Krishno Dey
ID: -171-15-9417
Department of CSE
Daffodil International University

Md. Mustahid Hassan
ID: -171-15-9388
Department of CSE
Daffodil International University

Md. Masud Rana
ID: -171-15-9413
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/internship successfully.

We are really grateful and wish our profound indebtedness to our Supervisor **Ms. Most. Hasna Hena**, **Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Machine Learning, Deep Learning and Computer Vision*" to carry out this project. Her 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 **Prof. Dr. Touhid Bhuiyan**, Professor and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty members 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 discussion while completing the course work.

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

ABSTRACT

Fish is an important part of Bangladeshi culture and cuisine. Fish is the main and sometimes the only source of protein in the rural household of Bangladesh. Moreover, thousands of peoples of Bangladesh are directly and indirectly dependent on the fish industry. With time many traditional indigenous Bangladeshi fish have lost their existence and many of them are in danger of losing their existence. Furthermore, the young generation of Bangladesh is unable to recognize these traditional indigenous fishes besides they are also missing out on the protein and nutrition provided by these indigenous fishes. Hence an automatic fish classification system can help us not only to recognize traditional fishes but also in the production and preservation of these indigenous fishes. So, here, we propose a convolutional neural network (CNN) based automatic fish classification system. In this paper, we mainly focus on the classification of traditional indigenous fishes of Bangladesh. We used a dataset of eight classes of indigenous fish which contains 8000 images after performing 8 types of augmentation methods. We applied this dataset to 3 different CNN models of different architecture namely model M1, M2 and M3. Convolution layers of the CNN model use “Adam optimizer”, “ReLU” and “Softmax” activation functions. Finally, among our proposed CNN model M1 provides an accuracy of 99.00% on the test data with rich precision, recall, and F1 score.

TABLE OF CONTENTS

CONTENTS	PAGE

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Introduction

Fish is one of the most common foods in the world and an integral part of the different cuisines around the world. Fish industry around the world is worth billions of thousands of dollars. As time goes by many fishes are losing their existence due to global warming and pollution. We can use modern research techniques to prevent these fishes from losing their existence. With not much effective research approach not applied to this sector leaves us with tremendous scope to apply effective research operations.

These indigenous fishes are an important part of the diet of Bangladeshi people and one of the main sources of protein for people of Bangladesh [1]. Moreover indigenous fishes are irreplaceable sources of protein and nutrition. These types of fish contain omega-3, omega-6 fatty acids and different vitamins, are essential for human health [1] [2] [3]. They also lower the risk of heart attacks and strokes, prevents cancer, boosts brain health and skin health [3].

There were 260 species of freshwater indigenous fishes in Bangladesh in 1989 [4]. But many of them have already lost their existence and many of them are still fighting to keep their existence. Moreover most of us can’t really recognize these indigenous fishes. Firstly we rarely come across the types of fishes and secondly most of the fishes look exactly the same. Hence, an automatic indigenous fish recognition system would benefit in recognition, preservation and production of these indigenous fishes.

The reason we have chosen this sector as an OR (operations research) model is because it could potentially be an effective application area of modern research methodologies. Besides we feel our research would be helpful for commercial production and preservation of indigenous fishes of Bangladesh.

Here we classified 8 different indigenous fishes using classification models. Dataset we used here contains images of eight classes of indigenous fishes, a total of 8000 images. We split our data into training, validation and test sets to pull off a better result. Then we fed the dataset into three different convolutional neural networks models each with different architecture and observed the performance of each model. We found out that model M1 provides best performance with accuracy of 99.00%.

Google has developed an open-source software library TensorFlow [5] which is an example of 2G artificial intelligence. Open source library of TensorFlow[6] is written in python and helps in implementations of deep learning algorithms and especially Neural Networks. Keras [7] is an open source Application Programming Interface (API) that's run on top of TensorFlow. Keras is very effective and robust API for easy implementations of neural network models. We implemented our convolutional neural networks models with keras using Google Colab [8] environment.

1.2 **Motivation**

We want to make a difference in our society through our research. Classifying indigenous fish images manually is not very efficient and error-prone because most of the fishes look almost the same. So our aim is to build an automated indigenous fish classification system so, even those people who have no prior knowledge about indigenous fishes can identify them easily and correctly through our proposed model. Through our work, we want to help those people who are working for the production and preservation of these indigenous fishes so that our next generation can get to see these indigenous fishes. Moreover, through our work, we want to make sure that our next generation does not miss out on the nutrition and protein provided by these indigenous fishes. Using our proposed model they will also be able to identify different fishes automatically and effortlessly.

1.3 **Rationale of Study**

A lot of work has been done on image classification using image processing and machine learning algorithms throughout the last few decades. But not a great deal of work has been done in this specific domain of Bangladeshi indigenous fish identification. Most of the work has employed different traditional image processing techniques and costly algorithms. Very little work is done using deep learning, especially CNN. Moreover, overall the previous work has been proven costly. With not much effective research approach not applied to the sector leaves us with tremendous scope to apply effective research operations.

With our proposed system we can perform classification with minimal cost and runtime and yet achieve a great accuracy. Compared to other works we had to apply very minimal data pre-processing for our model to generalize on validation and test data. We combated overfitting through augmentation and our proposed deep convolutional neural networks (CNN) eliminated the need of noise removal.

1.4 **Research Questions**

- Is it feasible to classify indigenous fish images manually?
- Is it possible to collect quality raw images of indigenous fishes?
- Is the proposed method an improvement over the current methods?
- How feasible is it to implement the proposed method?
- How will people benefit from the proposed method?

1.5 **Expected Output**

- To build a Convolutional Neural Networks model that generalized on test data as well as training data.
- To be able to classify indigenous fish images correctly with the proposed CNN model.
- Improvement of the existing methods.
- People will get benefited.

1.6 **Report Layout**

In Chapter 1 we have given an overview of our work with motivation, rationale of study, research questions, expected outcome and finally this current section provides the overall layout of this report.

Chapter 2 will provide an illustration about related work done in classification of fish, research summary, scope of problems and challenges.

Chapter 3 will provide an illustration about our research methodologies.

Chapter 4 will provide an illustration about our experimental result, discussion and summary of the model.

Chapter 5 will illustrate the impact our works have on our society and environments.

Chapter 6 will provide an illustration about the summary of our research, conclusion, limitations and future works.

CHAPTER 2

Background

2.1 **Introduction**

In this chapter we will discuss works that have been done in fish image classification. We will discuss the research paper and research methodology in the following section of related work. Then we will present a summary of related work in the summary section. Finally we will discuss the scope of the problem and challenges.

2.2 **Related Work**

Cao [9] et al. used a combination of CNN and hand designed image features for new feature creation. They specially focused on low resolution images and applied the proposed method on two distinct marine animal datasets. Their proposed method achieved greater accuracy measures than just using CNN alone.

Salman et al. [10] introduced a system that uses CNN for feature extraction and SVM for classification of underwater fish images. They achieve an average accuracy of 90%.

Rachmatullah et al. [11] used 2 layered CNN on low resolution dataset and they mainly emphasized on applied augmentation. They achieved an accuracy of 99.7% on test data.

Rathi et al. [12] introduced a combination of CNN, deep learning and image processing to encounter the noises of the underwater images and achieved an accuracy of 96.29%.

Chen et al. [13] introduced a two branch based automated fish detection system. Where one branch detects, aligns fishes from input no images and passes the fish image to the classifiers. The other branch makes use of fish instances and context information of the input image to infer the type of fish.

Sung et al. [14] introduced a convolutional neural network based on you only look once (YOLO) algorithm. Their introduced method successfully identified 93% of underwater fish video images.

Li et al. [15] proposed a fast R-CNN (Regions-Based Convolutional Neural and Networks) for fast and robust classification of huge amounts of underwater fish images. They showed how their approach can be very helpful for marine biologists to identify various classes of fishes quickly.

Islam et al. [16] introduced a new feature descriptor, hybrid census transform (HCBL) for Bangladeshi indigenous fish. They used SVM for classification and achieved an average accuracy of 90%.

2.3 Research Summary

In this section we present summaries of related work done in this field.

TABLE 2.1: SUMMARY OF RELATED WORKS

SL	**Authors**	**Methodology**	**Description**	**Outcome**
1	Z. Cao, J.C. Principe, B. Ouyang, F. Dalgleish, A. Vuorenkoski	CNN and hand design image feature	Hand designed image features are used along with CNN to tackle low resolution underwater images.	Developed an automated underwater fish detection system that robust to low resolution images.
2	A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, E. Harvey	CNN and SVM	They used CNN for feature extraction from underwater images and used SVM as a classifier.	Developed system achieve an average accuracy of 90%.

SL	Authors	Methodology	Description	Outcome
3	M. N. Rachmatullah, I. Supriana	CNN and Image Augmentation	They used image augmentation to increase the size of the underwater fish images dataset and then applied deep learning (CNN).	A fish image classifier produces and accuracy of 99.7% low resolution test data.
4	D. Rathi ; S. Jain ; S. Indu	CNN, Deep Learning and Image Processing Techniques	They applied a combination of CNN, deep learning and image processing to tackle noisy data.	Developed system achieved an accuracy of 96.29%.
5	G. Chen ; P. Sun ; Y. Shang	CNN	Contains two branches, One branch detects fish from a given image and another branch classifies the fishes based on fish instances and contextual information.	Proposed system able to classify the fishes from noisy input Image having a lot of context or details.
6	M. Sung ; S. Yu ; Y. Girdhar	CNN and You Only Look Once (YOLO) algorithms	They proposed a CNN architecture based on YOLO algorithms for robust	Developed system classify fishes with

SL	Authors	Methodology	Description	Outcome
			underwater fish image classification.	accuracy of 93% on test data.
7	X. Li, M. Shang. H. Qin, L. Chen	R-CNN (Regions with Convolutional Neural and Networks)	They proposed the use of R-CNN on a huge dataset for fast and robust identification of underwater fish images.	Proposed method capable of classifying wide species fishes very robustly.
8	M.A. Islam, M.R. Howlader, U. Habiba, R.H. Faisal, M.M. Rahman	SVM and hybrid census transform (HCBL) feature descriptor.	They focus on classification of eight classes of Bangladeshi indigenous fishes. They proposed a new feature descriptor for feature extraction and used SVM for classification.	Developed system can classify different indigenous fishes with an accuracy of 90% on test data.

2.4 Scope of the Problem

The scope of the above problem is widespread. There used to be 160 species of indigenous fishes in Bangladesh [4]. But currently that number has decreased dramatically. A lot of research work can be done using modern research methodology. Not a lot of work has been done in this specific domain. There are not a lot of attempts being made to build automated indigenous fish detection systems. That’s mainly because of the unavailability of the proper image datasets of these fishes. It’s really challenging to find image dataset of indigenous fishes because some of the species have gone out of existence and many of the species are at brink of extinction. Islam et al. proposed a Bangladeshi indigenous fish classification an achieved an accuracy of 90% [16].

So, the door is quite open in this domain to perform quality research work to help the biologist in classification and preservation of these indigenous fishes.

2.5 Challenges

During our work we were exposed to many challenges.

- Main challenge was to collect the appropriate data and process them into the appropriate format for us to use.
- Preprocess the collected data to make it suitable to feed the data to the CNN model.
- It requires a good system to implement deep learning algorithms.
- Usually deep learning algorithms take several minutes, hours and even days to run. We had to wait patiently to evaluate the results.

CHAPTER 3

Research Methodology

3.1 Introduction

In this section we are going to discuss our research methodologies. We have divided the methodologies into several steps namely, Data Collection, Data Pre-processing, Model Installation, Training the Model, Testing the Model.

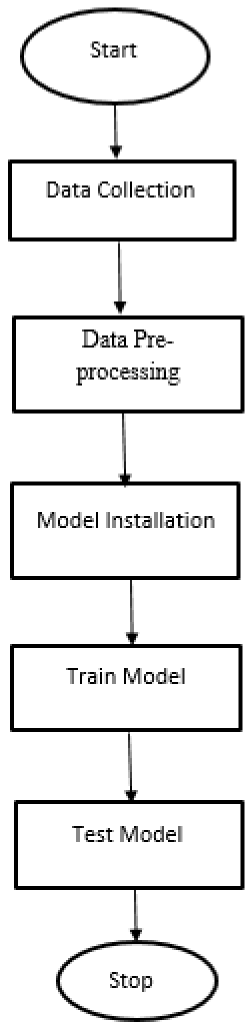


Figure 3.1: Working Steps

3.2 Data Collection

Here we used dataset namely BDIndigenousFish2019 publicly available at GitHub, created by Islam et al. [16]. Dataset contains eight classes of indigenous fish images. Dataset consists of a total 2610 images all are in .jpg format. Description dataset is given below with their frequency.

TABLE 3.1: DESCRIPTION OF THE DATASET WITH SCIENTIFIC NAMES [17].

SL	Common English Name	Scientific Name	Local Name	No. of Images
1	Lesser spiny eel	Macrognathus aculeatus	Tara baim (তারা বাইম)	500
2	Bronze featherback	Notopterus notopterus	Pholi (ফলি)	300
3	Climbing perch	Anabas testudineus	Koi (কৈ)	380
4	Stinging catfish	Heteropneustes fossilis	Shingi (শিঙি /শিঙ্গি)	400
5	Snakehead murrel	Channa striata	Shol(শৌল / শোল)	120
6	Olive barb	Puntius sarana	Sarpunti (সরপুঁটি)	200
7	Spotted snakehead	Channa punctata	Taki (টাকি)	390
8	Tyangra	Mystus tengara	Tengra (টেংরা)	320

A sample view of images of the dataset is given below. Here we have eight different species of fish.

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.2: A small slice of dataset, a. Lesser spiny eel, b. Bronze featherback, c. Climbing perch, d. Stinging catfish, e. Snakehead murrel, f. Olive barb, g. Spotted snakehead, h. Tyangra.

3.3 Data Pre-processing

Here we resized all the images to 224x244 pixels. We rescaled the value of each pixel into a range of 0 to 1 from the range of 0-255. Proper augmentation of image data is very important to increase the size of the data set which eventually helps us to achieve good performance measures by combating overfitting [18] [19] [20]. Five different types of augmentation are applied.

- Rotation of 40 degree
- Width shift range of 0.2
- Height shift range of 0.2
- Shear range of 0.2
- Zoom range of 0.2
- Horizontal flip is equal to true
- Fill mode is equal to nearest.

So, after augmentation, a total of 8000 images were created from the original 2610 images where each of eight classes contained 1000 images each. We divided our dataset into training, validation and test set for obtaining a better result. From each of eight classes 700 images (70%) used for training, 150 images (15%) used for validation, 150 images (15%) used for testing.

TABLE3.2 FREQUENCY DISTRIBUTION OF IMAGES AFTER AUGMENTATION.

Total Augmented Images	Training (70%)	Validation (15%)	Testing (15%)
8000	5600	1200	1200

3.4 Convolutional Neural Networks

Neural Network is one of the most widely used algorithms in the fields of data science, image processing, machine learning and deep learning. Neural Networks was inspired by the structure of the neurons of our brain [21]. Several variants of Neural Networks have been introduced over the years. Likes of Artificial Neural Network (ANN), Recursive Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Modular Neural Networks (MNN) are the most popular variants of Neural Networks.

Convolutional Neural Networks also known as CNN or ConvNet are the most widely used deep learning algorithms for image classification, image processing, image segmentation tasks, natural language processing etc. [22]. CNN is one the most robust and dynamic deep learning algorithms which require very minimal pre-processing.

CNN was created by Yann LeCun. It was actually inspired by the structure of the visual cortex area of our brain. CNN is a combination of a convolution layer and a fully connected layer where convolution layer extracts the feature form image and fully connected layer does the work of classification.

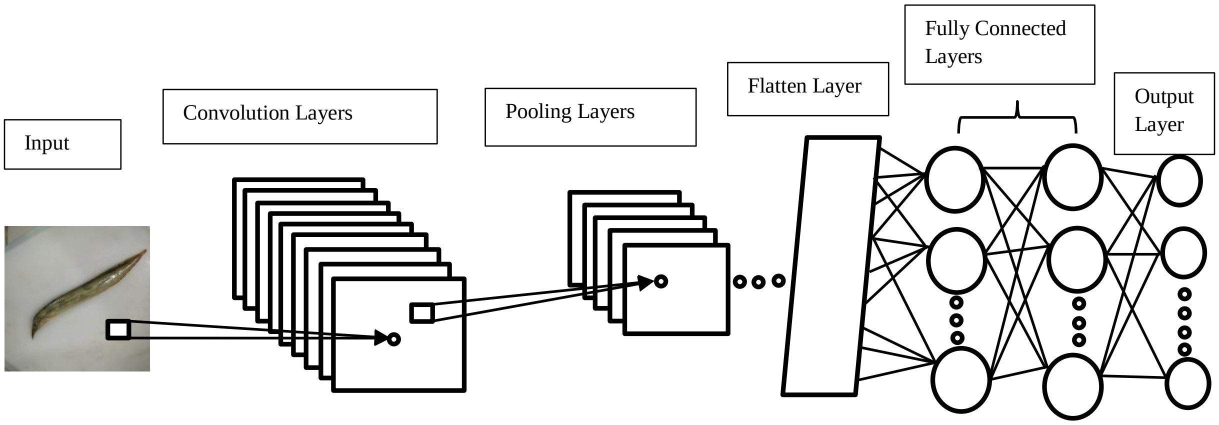


Figure 3.3: Simple architecture of CNN.

3.4.1 Convolutional Layers

Convolutional layer is the layer that actually learns and explores the feature of an image received from the input layer and then passes it to the fully connected layer for further processing. Arguably it is the most important part of a CNN model. In CNN input images are represented as tensor or multidimensional arrays. The word convolution means sliding a NxN kernel or filter over an image. Dot multiplication is performed between images and filters to create a feature map.

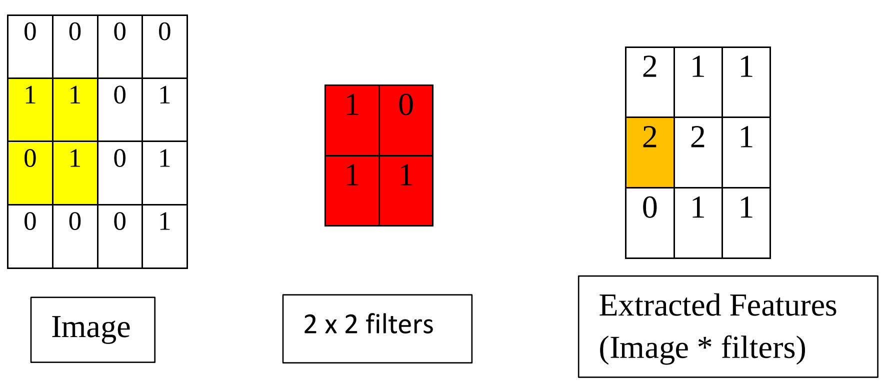


Figure 3.4: Convolution of an image using 2 x 2 filters.

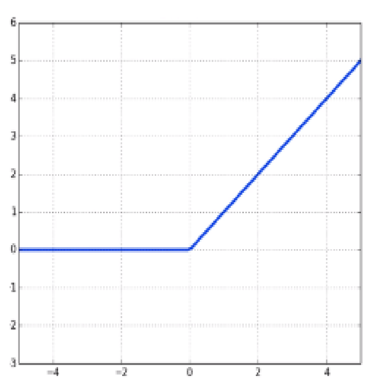
Here a 2 x 2 filter or kernels convolve or slide over an image and perform dot product between image and filter. This feature extraction if performed k times is k number of filters is used in a convolution layer.

A convolutional layer also contains an activation layer and subsequently followed by a pooling layer. A CNN can contain one or multiple convolutional layers and each convolution layer may have one or more filters depending on the requirements and complexity of the problem.

3.4.1.1 Activation Function

An activation function of a node takes inputs and performs some kind of operation on the inputs, passing it to the next layer. Input here is the multiplication of inputs and weights plus the bias factors. An activation function takes the input and represents it into a non-linear format and passes to the next layers. Activation function helps us to do classification and calculate the error rate so that we can apply backpropagation to relearn the weights.

CNN usually uses ReLU (Rectified Linear Unit) activation function [22]. It is one of the most popular activation functions used in deep learning because of its simplicity and better optimization capability compared to sigmoid or tanh activation function. It also learns faster than other activation functions. But it is only used for hidden layers. It converts an input into either zero or one. If the input value is less than zero it returns zero and when input value is greater than zero it outputs a linear slope of one.



$$RELU(x) = \begin{cases} 0 \; if \; x < 0 \\ x \; if \; x >= 0 \end{cases}$$

Figure 3.5: ReLU Activation Function.

3.4.1.2 Pooling Layers

Pooling layers actually summarizes the feature map of the image by reducing the dimensionality. It employs a down sampling to get rid unnecessary features or details of an image to avoid overfitting. One of the problems of feature creation is that it captures every tiny detail of an image. Sometimes unnecessary features are captured that lead to overfitting of the model. Pooling helps us to overcome this issue. There are many pooling techniques available like, average pooling, max polling and min pooling. Here we only used max pooling with stride of 2 and a 2 x 2 filter. That means a 2 x2 filter on the feature map extracts the max value and then we slide over by 2 columns and repeat the same process until we reach the end of the feature map.

Max pooling actually extracts max value from some specific dimension of the matrix. We illustrate the concept with an example where we apply max pooling with 2 x 2 filters and stride 2 on the feature map. Dimension of 4x4 feature map reduced to 2x2 dimension.

Figure 3.6: Max Pooling Operation.

3.4.2 Flatten Layer

This layer receives the inputs from the convolutional layer and converts it into one dimension so these inputs can be used in fully connected layers. This layer actually works as an input layer for fully connected layers. Then finally feed that one dimensional structure to the fully connected layers.

3.4.3 Fully Connected Layers

Fully connected layer is basically a feed forward artificial neural network. Where each node in one layer is connected with the each node of the next layer and so on. These layers actually perform the classifications for us. This layer is also called hidden layers and followed by an output layer. The output produces some quantity of probability of each class label. An image is classified as a label that has the highest probability. This layer may contain one or multiple hidden layers depending on the complexity of the problems.

These layers also contain activation layers (shown in section 3.2.1.1). In hidden layers RELU activation function (fig: 3.4) is used and for the output layer softmax activation function is used. Dropout layers are also used to combat overfitting in fully connected networks.

3.4.3.1 Softmax Activation Function

Softmax activation function is also known as normalized exponential function [23]. It takes an input vector for k labels and normalizes the vector into probability distribution of k labels. It is widely used in the output layer of a neural network model to draw out a probability form the input data.

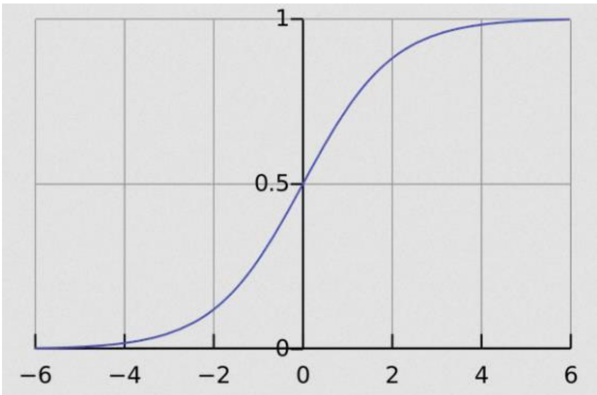


Figure 3.7: Softmax Activation Function.

3.4.3.2 Dropout Layers

Dropout layers actually randomly drop a few of inputs form the previous layer of the neural network. This helps the model to avoid learning unnecessary features during training [24]. As a result it can generalize well on the validation and test data.

3.5 Model Installations

Here we proposed three different convolutional neural networks namely model M1, M2 and M3. Each of them is different in terms of architecture and learning parameters. For actual implementation was done using Keras Application Programming Interface (API) which runs on top of Google's introduced Tensorflow python library.

3.5.1: Model M1

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 128)     3584
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 128)     0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 112, 112, 512)     590336
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 512)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 56, 56, 256)       1179904
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 256)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 64)        147520
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 14, 14, 128)       73856
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 128)               802944
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               16640
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 2056
=================================================================
Total params: 2,825,096
Trainable params: 2,825,096
Non-trainable params: 0
```

Figure 3.8: Summary of Model M1

We used 5 convolutional layers in our first model. Each convolution layer uses RELU activation function and is followed by a 2 x 2 Max Pooling Layer of stride 2. Model M1 has 2.82 million of trainable parameters. We have used Conv2D(), MaxPool2D() to build convolutional layers, then a Flatten() to flatten the input and finally Dense() and Dropout() for fully connected layers. RELU activation function for each of the hidden layers and softmax activation function for the output layer. Kernel of size 3x3 used in each convolutional layer. The Input shape for the model is 224x224. Figure 3.8 shows the whole summary of model M1.

3.5.2: Model M2

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 222, 222, 128)     3584      
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 111, 111, 128)     0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 109, 109, 64)      73792     
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 64)        0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 52, 52, 128)       73856     
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 128)       0         
_________________________________________________________________
flatten (Flatten)            (None, 86528)             0         
_________________________________________________________________
dense (Dense)                (None, 64)                5537856   
_________________________________________________________________
dropout (Dropout)            (None, 64)                0         
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 520       
=================================================================
Total params: 5,689,608
Trainable params: 5,689,608
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.9: Summary of Model M2

We used 3 convolutional layers in our second model. Each convolution layer uses RELU activation function and is followed by a 2 x 2 Max Pooling Layer of stride 2. Model M1 has 5.68 million of trainable parameters. We have used Conv2D(), MaxPool2D() to build convolutional layers, then a Flatten() to flatten the input and finally Dense() and Dropout() for fully connected layers. RELU activation function for each of the hidden layers and softmax activation function for the output layer. Kernel of size 3x3 used in each convolutional layer. The Input shape for the model is 224x224. Figure 3.9 shows the whole summary of model M2.

3.5.3: Model M3

We used 2 convolutional layers in our first model. Each convolution layer uses RELU activation function and is followed by a 2 x 2 Max Pooling Layer of stride 2. Model M1 has 12.24 million of trainable parameters.

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 222, 222, 256)     7168
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 111, 111, 256)     0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 109, 109, 128)     295040
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 54, 54, 128)       0
_________________________________________________________________
flatten_4 (Flatten)          (None, 373248)            0
_________________________________________________________________
dense_8 (Dense)              (None, 32)                11943968
_________________________________________________________________
dropout_4 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_9 (Dense)              (None, 8)                 264
=================================================================
Total params: 12,246,440
Trainable params: 12,246,440
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.10: Summary of Model M3

We used Conv2D(), MaxPool2D() to build convolutional layers, then a Flatten() to flatten the input and finally Dense() and Dropout() for fully connected layers. RELU activation function for each of the hidden layers and softmax activation function for the output layer. Kernel of size 3x 3 used in each convolutional layer. The Input shape for the model is 224x224. Figure 3.10 shows the whole summary of model M3.

3.6 Train Model

In this section we train our three proposed models with our train data set and learn train, validation accuracy and loss.

3.6.1 Model M1

We used 5600 images (1000 for each class) for training and 1200 images for both validation and testing on our model M1. Model M1 was run for 20 epoch since 20 epoch was quite enough for our model to converge and give a more stable accuracy measure.

For compilation of the model we use Adam optimizer with learning rate of 0.0001 and for loss function we use categorical-crossentropy as we are dealing with a multiclass classification problem. Figure 3.11 shows train and validation loss graph over the number the epoch ran.

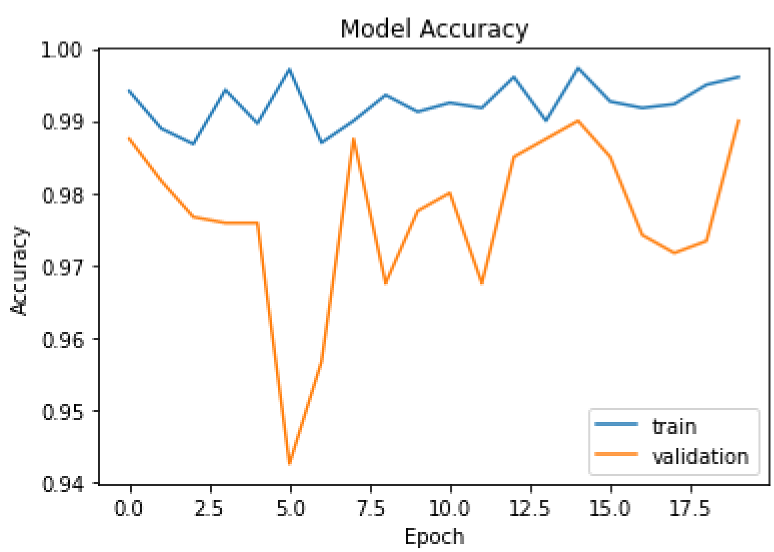


Figure 3.11: M1 Accuracy Graph

From Figure 3.11 we can see both training and validation accuracy started off at quite high as the number of epochs increasing both training and validation accuracy are converging to a more stable value. At 20th epoch accuracy seems to be stalling and training accuracy is 99.61% and validation accuracy is 99.00%. As there is very little difference between train and validation accuracy, there is no overfitting or underfitting.

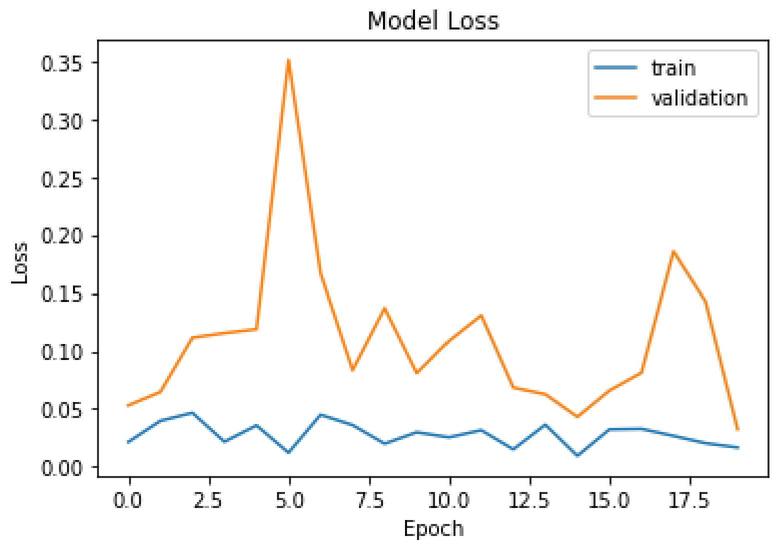


Figure 3.12: M1 Loss Graph.

From Figure 3.12 we can see that both training loss started at quite low. Validation loss started off at quite high but significantly decreased as the number of epochs increased. Both training and validation converges to a stable value with epoch. Very little difference between training and validation loss ensures there is no overfitting or underfitting.

3.6.2 Model M2

We used 5600 images (1000 for each class) for training and 1200 images for both validation and testing on our model M2. M2 was run for 20 epoch since 20 epochs was quite enough for our model to converge and give a more stable accuracy measure.

For compilation of the model we use Adam optimizer with learning rate of 0.00001 and for loss function we use categorical-crossentropy as we are dealing with a multiclass classification problem.

Figure 3.13 shows train and validation loss graph over the number the epoch ran .Here we can see both training and validation accuracy started off at quite high as the number of epochs increasing both training and validation accuracy are converging to a more stable value. At 20th epoch accuracy seems to be stalling and training accuracy is 88.55% and validation accuracy is 88.33%. As there is very little difference between train and validation accuracy, there is no overfitting or underfitting.

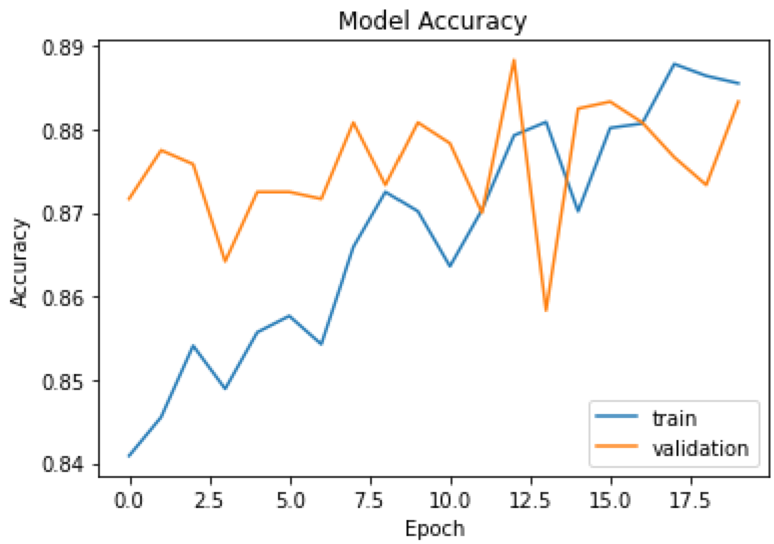


Figure 3.13: M2 Accuracy Graph.

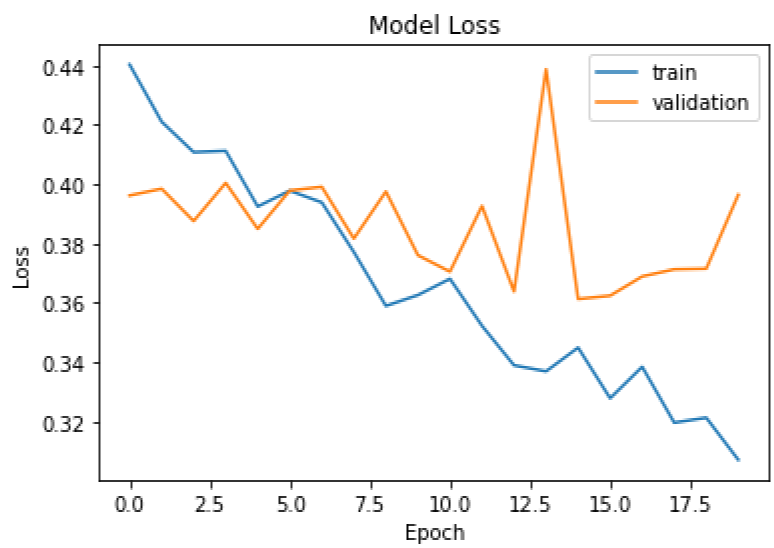


Figure 3.14: M2 loss Graph.

Figure 3.14 shows train and validation loss graph over the number the epoch ran. Here we can see that training loss started at higher between two but decreased with the number of epochs. On the other hand validation accuracy started off low between two but remained somewhat stable as the number of epochs increased. Very little difference between training and validation accuracy ensure there is no overfitting.

3.6.3 Model M3

We used 5600 images (1000 for each class) for training and 1200 images for both validation and testing on our model M3. M3 was run for 20 epoch since 20 epoch was quite enough for our model to converge and give a more stable accuracy measure.

For compilation of the model we use Adam optimizer with learning rate of 0.001 and for loss function we use categorical-crossentropy as we are dealing with a multiclass classification problem.

Figure 3.15 shows train and validation accuracy graphs. Figure 3.16 shows train and validation loss graph over the number the epoch ran.

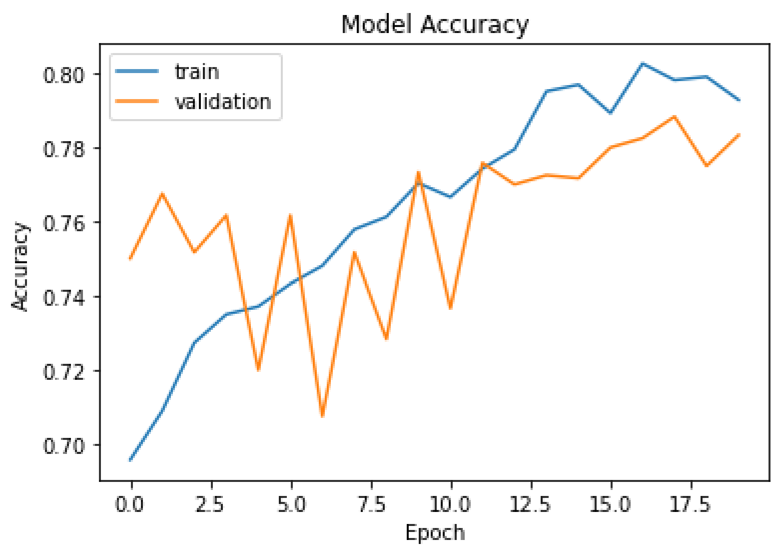


Figure 3.15: M3 Accuracy Graph

From Figure 3.15 we can see both training and validation accuracy started at very little but as the number of epochs increases both training and validation accuracy is also increasing. At 20th epoch accuracy seems to be stalling and training accuracy is 79.29% and validation accuracy is 78.33%. As there is quite a small difference between train and validation accuracy, so there is no overfitting or underfitting.

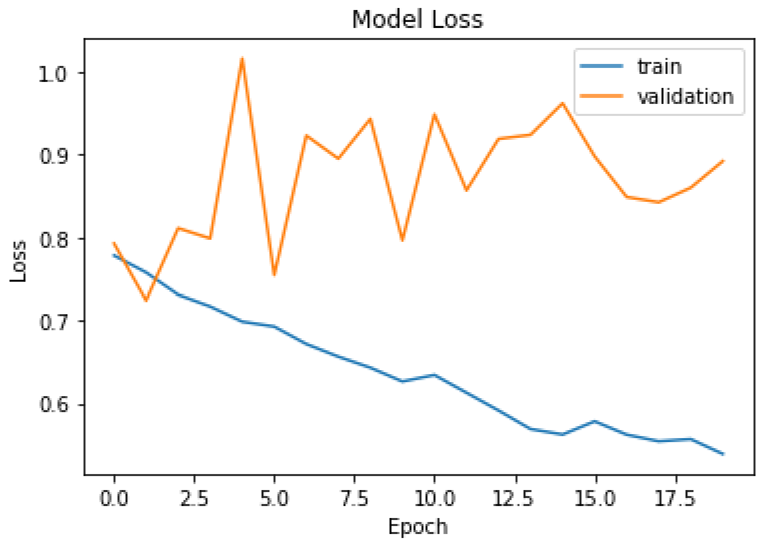


Figure 3.16: M3 loss Graph

From Figure 3.16 we can see that both training and validation loss started off at high but significantly decreased as the number of epochs increased. Very little differences between training and validation loss ensure there is no overfitting or underfitting.

3.7 Test Model

Here we test our proposed all three models to check how much the model is generalizing on test data after training. We trained our model with 5600 images for all three models and tested these three models of a test dataset of 1200 images. For measuring performances of our proposed models we used, confusion matrix and calculated different performance measures like accuracy, precision, recall, f1 score and ROC curve. A detailed description of performance measures and comparison among proposed models is given in chapter 4.

3.7.1 Model M1

Upon completion of testing we get accuracy of 99.00% which is not surprising since validation accuracy was also 99.00% and training accuracy was 99.61% for model M1. So, we can fairly say that model M1 is generalizing very well on test data of 1200 images. It correctly classified 1188 images out of 1200 images, and 12 images were incorrectly classified.

3.7.2 Model M2

Upon completion of testing we get accuracy of 88.58% which is not surprising since validation accuracy was also 88.33% and training accuracy was 88.55% for model M2. We can see that model M2 is generalizing very well on test data of 1200 images. It correctly identified 1063 images out of 1200 images and 137 images were incorrectly classified.

3.7.3 Model M3

Upon completion of testing we get accuracy of 78.58% which is not surprising since validation accuracy was also 78.33% and training accuracy was 79.29% for model M3. We can see that model M3 is generalizing on test data of 1200 images. It correctly identified 943 images out of 1200 images and 257 images were incorrectly classified.

3.8 Implementation Requirements

We've done most of our training and testing on Google Colaboratory (Google Colab) [8]. It is an online deep learning development platform. It's environment is the same as the jupyter notebook environment except it's a cloud service.

Based on the complexity of our problem, which we have discussed throughout this report, the probable Implementation requirements would be the following.

Hardware and Software Requirements

- Any Operating System
- Any Internet Browser
- RAM (2GB or more)
- HDD (100GB or more)
- CPU with a decent speed

Development Requirements

- Stable Internet Connection
- Google Colab(Use GPU or TPU)

CHAPTER 4

Experimental Results and Discussion

4.1 Introduction

In this section we discuss the result of our three proposed models. As mentioned in table [3.2] we used a total 1200 images (150 images of each class) for testing. We have used confusion matrix to determine accuracy, precision, recall and f1 score. Later we have used Receiver Operating Characteristics (ROC) curve analysis for each of three models. Then we analyzed and compared the results of proposed three models. Finally we concluded this section with the summary of the results.

4.1.1 Confusion Matrix

Confusion matrix is very popular in the field of machine learning for visualization and performance evaluation of algorithms [25] [26]. It helps us to check the correctness of a model or algorithms. We cannot evaluate a model with just accuracy, we also need precision, recall, f1 score, receiver operating characteristics curve etc. and we can determine all this through a confusion matrix. Some of the common terms associated with the confusion matrix are the following.

- TP (True Positive): When an event occurs and the model predicts true.
- TN (True Negative): When an event doesn't occur and the model predicts false.
- FP (False Positive): When an event doesn't occurs but the model predicts true.
- FN (False Negative): When an event occurs but the model predicts false.

With four terms we can determine accuracy, precision, recall, f1 score, true positive rate, false positive rate as follows.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{Recall / True Positive Rate} = \frac{TP}{TP + FN}$$

$$\text{Pr}ecison = \frac{TP}{TP + FP} \qquad \text{F1 Score } = \frac{2 * Accuracy * \text{Pr}ecision}{Accuracy + \text{Pr}ecision}$$

4.2 Experimental Results

In this section we described the results of our proposed model 1(M1), model 2 (M2), model 3 (M3) respectively. We discussed the performance measures model of all three models with confusion matrix and ROC curves.

4.2.1 Model M1 Result

TABLE 4.1: CONFUSION MATRIX OF MODEL M1

		Predicted							
		Lesser spiny eel	**Bronze featherba**	**Climbing perch**	**Stinging catfish**	**Snakehead murrel**	**Olive barb**	**Spotted snakehea**	**Tyangra**
Actual	**Lesser spiny eel**	148	0	0	0	0	0	1	1
	Bronze featherback	0	150	0	0	0	0	0	0
	Climbing perch	0	0	147	0	0	0	3	0
	Stinging catfish	0	1	0	149	0	0	0	0
	Snakehead murrel	0	0	0	1	149	0	0	0
	Olive barb	0	0	0	0		150	0	0
	Spotted snakehead	0	0	5	0	0	0	145	0
	Tyangra	0	0	0	0	0	0	0	150

From table [4.1] we can plot graphical representations of performance measures. Accuracy of the M1 model is 99.00% on test data. We achieve excellent recall, precision and f1 scores shown in figure 4.1

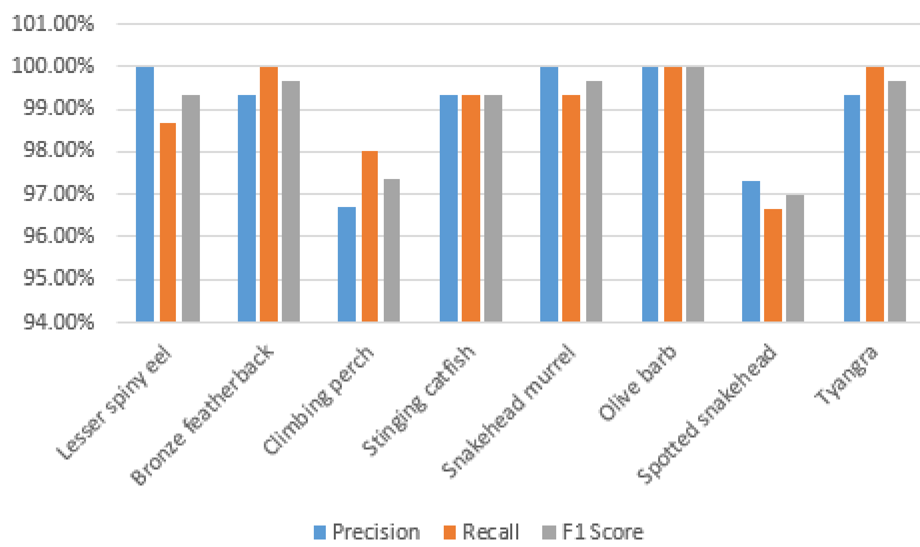


Figure 4.1: Precision, Recall and F1 Score of model M1

The ROC Curve of Model M1 shows us 100% is area under the ROC curve for all 8 classes of images.

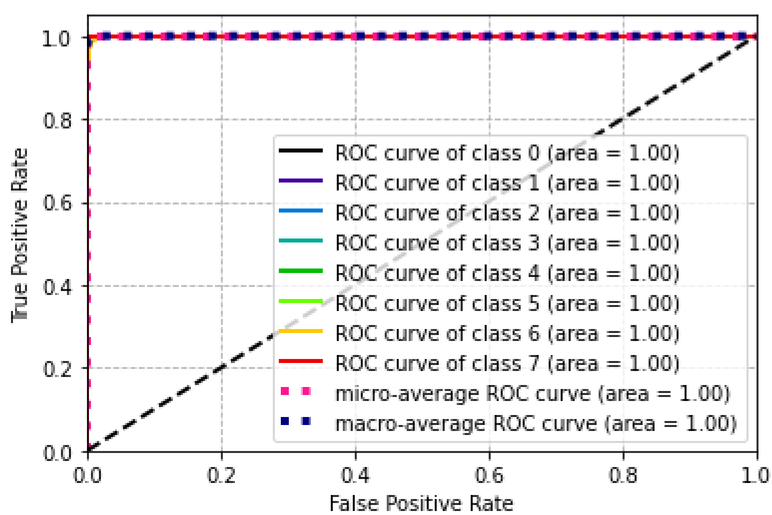


Figure 4.2: ROC Curve of model M1

4.2.2 Model M2 Result

TABLE 4.2: CONFUSION MATRIX OF MODEL M2

		Predicted							
		Lesser spiny eel	**Bronze featherback**	**Climbing perch**	**Stinging catfish**	**Snakehead murrel**	**Olive barb**	**Spotted snakehead**	**Tyangra**
Actual	**Lesser spiny eel**	134	2	2	4	0	1	4	3
	Bronze featherback	0	145	0	0	1	2	0	2
	Climbing perch	4	1	113	7	0	2	22	1
	Stinging catfish	1	3	1	132	3	0	1	9
	Snakehead murrel	0	0	0	0	150	0	0	0
	Olive barb	0	1	1	0	6	141	0	1
	Spotted snakehead	11	0	14	5	3	0	116	1
	Tyangra	1	9	0	4	0	4	0	132

From table [4.2] we can plot graphical representations of performance measures. Accuracy of the M2 model is 88.58% on test data. We achieved excellent recall, precision and f1 scores shown in figure 4.3.

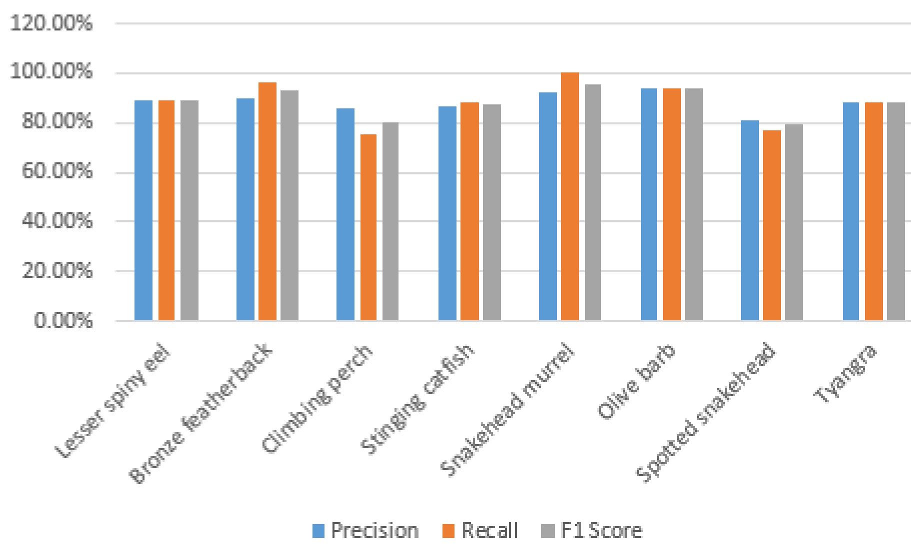


Figure 4.3: Precision, Recall and F1 Score of model M2

The ROC Curve of model M2 shows us that the average of 98.91% area is under the ROC curve for all 8 classes of images.

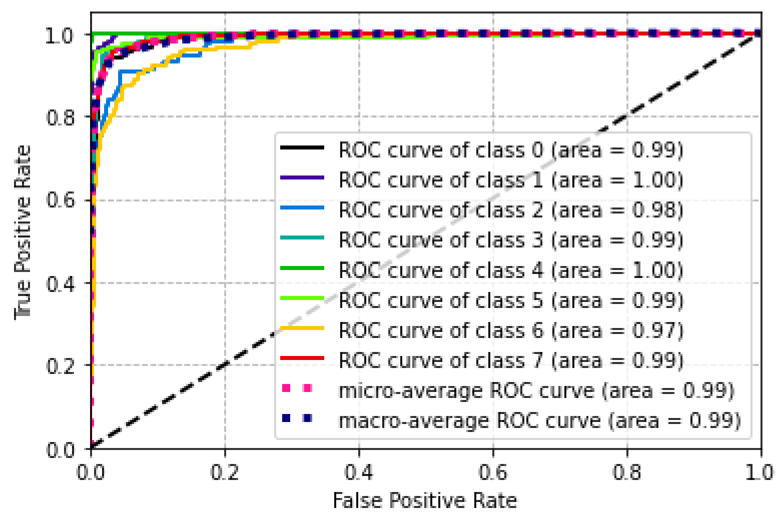


Figure 4.4: ROC Curve of model M2

4.2.3 Model M3 Result

TABLE 4.3: CONFUSION MATRIX OF MODEL M3

		Predicted							
		Lesser spiny eel	**Bronze featherback**	**Climbing perch**	**Stinging catfish**	**Snakehead murrel**	**Olive barb**	**Spotted snakehead**	**Tyangra**
Actual	**Lesser spiny eel**	109	0	3	4	6	16	10	2
	Bronze featherback	0	105	0	31	2	0	0	12
	Climbing perch	6	0	106	1	0	1	35	1
	Stinging catfish	0	7	0	142	1	0	0	0
	Snakehead murrel	0	0	0	3	144	1	2	0
	Olive barb	26	0	2	2	3	111	3	3
	Spotted snakehead	9	0	29	2	1	1	108	0
	Tyangra	1	10	1	13	0	7	0	118

From table [4.3] we can plot graphical representations of performance measures. Accuracy of M3 model is 78.58%. We achieved excellent recall, precision and f1 scores shown in figure 4.5.

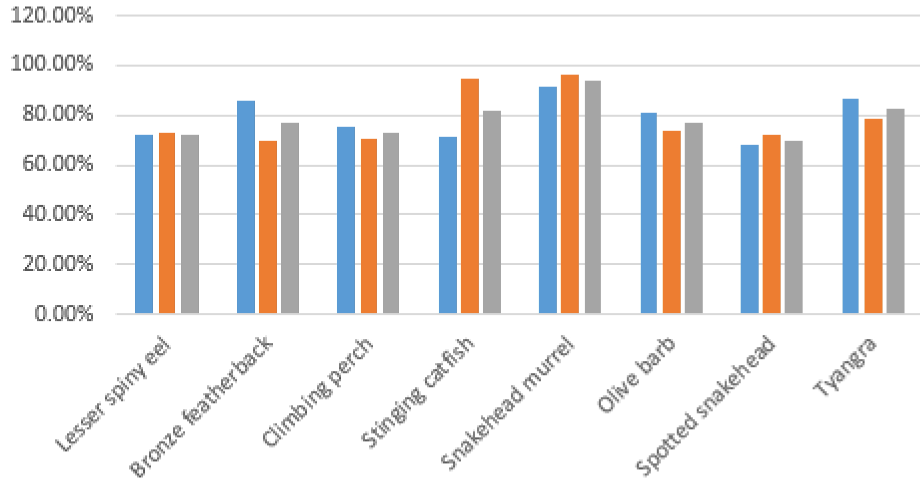


Figure 4.5: Precision, Recall and F1 Score of model M3

The ROC Curve of model M3 shows us that the average of 96.75% area is under the ROC curve for all 8 classes of images.

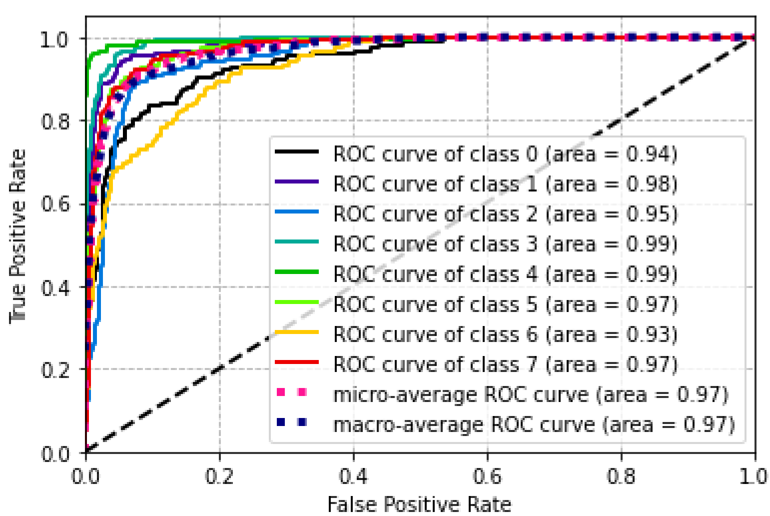


Figure 4.6: ROC Curve of model M3

4.3 Descriptive Analysis

In this section we compared the M1, M2, M3 models based on the average performance measures. Firstly we calculated the average accuracy, precision, recall and f1 scores. From these performance measures and ROC curve analysis it is clearly proved that the M1 model gives the best performance than the other two models. Figure 4.7 shows the comparison among M1, M2, M3 models. M1 is superior to both M2 and M3 in all respects of the performance measures and provides accuracy of 99.00% on test data. On the other hand M2 is superior to only M3 in all respects of performance measures and provides an accuracy of 88.58% on test data. Although model M3 not either superior to M1 or M3 but it still provides 78.58% accuracy with decent precision, recall, f1 scores and area under ROC is also decent. Additionally none of the models is overfitted or underfitted. All three models are proved to provide decent performance. Finally model M1 is the best and should be deployed for further applications.

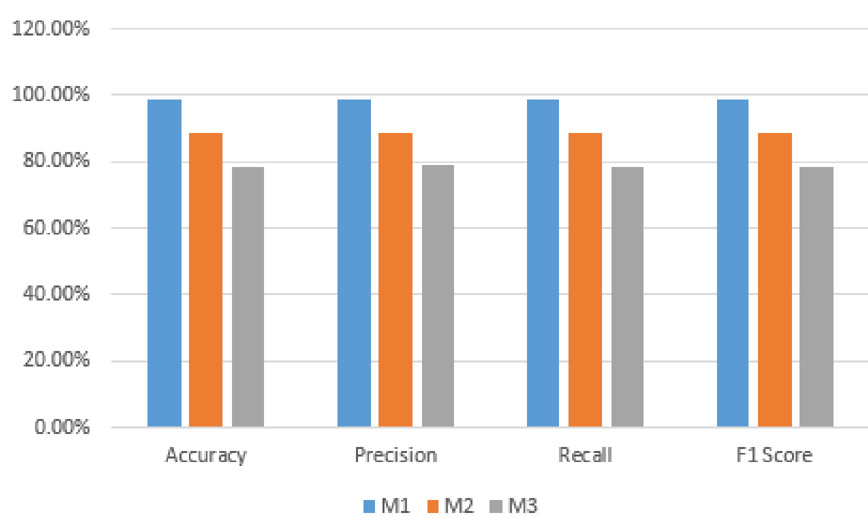


Figure 4.7: Performance Comparison of models M1, M2 and M3.

4.4 Summary

So far we have described all our experimental results. We compared among all three of our proposed models. Based on the performance measures, 5 convolutional layer based model M1 turn out to be the best model among all three proposed models. On the other hand 4 convolutional layer based model M2 provide a better performance than 2 convolutional layer based model M3. But it is fair to say all three proposed models provide decent results.

CHAPTER 5

Impact on Society, Environment and Sustainability

5.1 Impact on society

Our work will have a huge positive impact on our society in the identification, preservations, and production of indigenous fishes. As a result, the yearly production of these fishes will increase and satisfy the need for proteins in Bangladeshi households. So our next generations will be able to see and get proteins from these fishes. We won't have to depend on meat to satisfy the needs of proteins. Fish contains final vitamins that are essential to children's development. Moreover, fishes are healthier and cheaper sources of proteins, so everyone will be able to afford it. As production increases, this will create a huge number of job opportunities for the peoples of Bangladesh in rural areas, Thus it will help to reduce unemployment. Fish Fisherman's will especially be benefitted from this, ensuring them a stable source of income. Using our system young fish cultivators will be able to select the right and healthy species of baby fishes for cultivation. Thus our system will help in the development of the socio-economic conditions of our society.

5.2 Impact on Environments

Preservations of different indigenous fishes will help us to maintain the balance of the ecosystem of our rivers and ponds. Moreover, the cultivation of diverse species of fishes will help in the proper utilization of food in each different level of water resources. Our proposed system will not have any negative impact on society, as it is an automatic system that identifies fishes from their pictures. Thus our system will ensure a stable environment, by preserving fishes that are in danger of losing their existence.

Robots will take over agriculture in the future, proper fish identification will help agricultural robots to make better decisions. The agricultural sector will revolutionize in a few decades, where everything will be automated, to increase the production of foods for gigantic populations of the earth. Thus automatic classification systems will play an important role. So, our automatic fish classification system will also be a part of the agricultural revolutions.

5.3 Ethical Aspects

Our proposed system does not cause any harm to the environments, rather it helps in maintaining the stability of the environments. Moreover, our system does not invade anyone's privacy or deprive anyone's freedom of speech. From data collection to classification, our system does violate any ethical codes. Data used in our system is fish images that were collected with permissions. We maintained complete privacy for any additional information that we collected for our system.

Our system will help the preservation of indigenous fishes, thus will have a positive impact in society and maintains all ethical aspects.

5.4 Sustainability Plan

We can deploy our system to Web applications or Android applications, which will ensure instant automatic identifications of indigenous fishes of Bangladesh. People will be able to identify fish species by just taking a picture on smartphones. Buyers will cross-check knowledge about fish species during buying fish. The fish cultivator can identify the correct species of fish for cultivations. Finally, our system will help people in different ways.

CHAPTER 6

Summary, Conclusion and Implication for Future Research

6.1 Summary of Study

A lot has been done of fish classification using convolutional neural networks (CNN). But most of the work has been on underwater fish images. There is very little work being done on the classification of Bangladeshi indigenous fishes. Among those few works none of them used convolutional neural networks.

So we proposed a system to classify indigenous fishes of Bangladesh using convolutional neural networks. After different experimentation and analysis we found out the model M1 is the best. M1 is capable of classifying fish images with an accuracy of 99.00%.

6.2 Conclusions

We have illustrated our work throughout this report. We proposed three architectures (M1, M2, M3 models). We fed out the prepared dataset into three modes and trained the models and observed training and validation accuracies. Then we tested the models with test data and evaluated the models based on various different performance measures. Finally, we discovered M1 gives the best results although the other two (M2, M3) models also classified test data with decent accuracy. So we can say all three models are great classifiers for our Bangladeshi indigenous fish images dataset. If we can generate a bigger dataset we can apply MobileNet, DenseNet, VGG16, Inception v3, etc. to yield even better results.

6.3 Implication for Further Study

The challenge of our work was the collection of data. Indigenous fish images are hard to find. Mostly because a lot has gone out of existence and others are on brink of extinction. We also contacted Bangladesh Fisheries Research Institute (BFRI) even, they don’t have enough image data for all indigenous fishes probably because there has not been a lot of research conducted on indigenous fish classifications.

We feel if we can manage to collect a lot of images of these fishes with the collaboration of BDRI we can build a classification system for a large-scale indigenous fish classification. That would greatly benefit BFRI, they would no longer be required to detect each fish manually. The classification system would detect fishes automatically, easily, and accurately. So in the future, we want to build a classifier for large scale classification of Bangladeshi indigenous fishes.

REFERENCES:

[1] N. Roos; M.M. Islamyand; S.H. Thilsted "Small Indigenous Fish Species in Bangladesh: Contribution to Vitamin A,Calcium and Iron Intakes", Animal Source Foods and Nutrition in Developing Countries, vol. 6, pp 1-6, June 24–26, 2002.

[2] Learn about Health Benefits of Fish, available at <<https://www.doh.wa.gov/communityandenvironment/food/fish/healthbenefits>>, last accessed 07-09-2020at01:07am

[3] Learn about 14 benefits of fish, available at << https://www.organicfacts.net/fish.html/>>, last accessed 07-09-2020 at01:07am

[4] Learn about indigenous fishes of Bangladesh, available at << http://en.bdfish.org/2009/11/small-indigenous-species-sis-of-fishes/>>, last accessed 07-09-2020 at01:07am

[5] M. Abadi, A. Agarwal, et al, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems", CoRR abs/1603.04467, 2016

[6] Learn about TensorFlow, available at <<https://www.tensorflow.org/>>, last accessed 07-09-2020 at01:07am.

[7] Learn about keras, available at << https://keras.io/>>, last accessed 12-10-2020 at 07:57pm

[8] Learn about Google Colab availblae at << https://colab.research.google.com/notebooks/intro.ipynb# >>, last accessed 12-10-2020 at 07:57pm

[9] Z. Cao, J.C. Principe, B. Ouyang, F. Dalgleish, A. Vuorenkoski, "Marine animal classification using combined CNN and hand-designed image features", OCEANS 2015 - MTS/IEEE Washington, vol. 6, 19-22 Oct. 2015.

[10] A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, E. Harvey 'Fish species classification in unconstrained underwater environments based on deep learning". Limnology and Oceanography: Methods, vol.16, 2016.

[11] M. N. Rachmatullah, I. Supriana, "Low Resolution Image Fish Classification Using Convolutional Neural Network", 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), vol.6, 14-17 Aug. 2018.

[12] D. Rathi ; S. Jain ; S. Indu,"Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning", 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), vol. 6, 27-30 Dec. 2017.

[13] G. Chen ; P. Sun ; Y. Shang, "Automatic Fish Classification System Using Deep Learning", 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), vol 6, 6-8 Nov. 2017.

[14] M. Sung ; S. Yu ; Y. Girdhar, "Vision based real-time fish detection using convolutional neural network", OCEANS 2017 – Aberdeen, vol. 6, 19-22 June 2017.

[15] X. Li, M. Shang. H. Qin, L. Chen, "Fast accurate fish detection and recognition of underwater images with Fast R-CNN", OCEANS 2015 - MTS/IEEE Washington, vol. 5, 19-22 Oct. 2015.

[16] M.A. Islam, M.R. Howlader, U. Habiba, R.H. Faisal, M.M. Rahman, "Indigenous Fish Classification of Bangladesh using Hybrid Features with SVM Classifier", 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), vol. 4, pp. 1-4, 11-124, July 2019.

[17] Learn about indigenous fish of Bangladesh and their scientific names, available at << https://en.wikipedia.org/wiki/List_of_fishes_in_Bangladesh>>, last accessed 12-10-2020 at 07:57pm

[18] A. Mikołajczyk, M. Grochowski, "Data augmentation for improving deep learning in image classification problem", 2018 International Interdisciplinary PhD Workshop (IIPhDW), vol. 6, 9-12 May 2018

[19]A. Fawzi ; H. Samulowitz ; D. Turaga ; P. Frossard, "Adaptive data augmentation for image classification", 2016 IEEE International Conference on Image Processing (ICIP), vol. 5, 25-28 Sept. 2016

[20] C. Shorten; T.M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", *Journal Big Data* **6,** 60 (2019). https://doi.org/10.1186/s40537-019-0197-0

[21] Learn about Neural Networks and it's variants, available at << https://en.wikipedia.org/wiki/Neural_network>>, last accessed 07-09-2020 at01:07am.

[21] Learn about Convolutional Neural Networks, available at << https://en.wikipedia.org/wiki/Convolutional_neural_network>>, last accessed 07-10-2020 at01:07am.

[22] Learn about ReLU activation function, available at << https://en.wikipedia.org/wiki/Rectifier_(neural_networks)>>, last accessed 07-09-2020 at01:07am.

[23] Learn about Softmax activation function, available at << https://en.wikipedia.org/wiki/Softmax_function>>, last accessed 07-09-2018 at01:07am.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", vol. 30, pp 1-30, 2014

[25] Learn about Confusion Matrix, available at << https://en.wikipedia.org/wiki/Confusion_matrix>>, last accessed 13-012-2020 at10:22am.

[26] Learn about Confusion Matrix, available at << https://medium.com/greyatom/performance-metricsforclassification-problems-in-machine-learning-part-i-b085d432082b >>, last accessed 13-12-2020 at10:22am.

APPENDICES

Abbreviations:

ANN = Artificial Neural Networks

RNN = Recurrent Neural Network

MNN = Modular Neural Networks

CNN = Convolutional Neural Networks

ConvNet = Convolutional Neural Networks

RCNN = Regions-Based Convolutional Neural and Networks

SVM = Support Vector Machine

ReLU = Rectified Linear Unit

ROC = Receiver Operating Characteristic

API = Application Programming Interface

GPU = Graphics Processing Unit

TPU = Tensor Processing Unit

M1 = Proposed CNN model 1

M2 = Proposed CNN model 2

M3 = Proposed CNN model 3

Appendix: Research Reflections and Acknowledgements

We started this research project with great enthusiasm. In the beginning, we had very little knowledge of Machine Learning, Deep Learning, and Computer vision. But our respected supervisor always encouraged and helped us to learn about those concepts. She was always

very kind and patient with us. Basically, she guided us thoroughly throughout this research project.

We are so grateful to her for all the help and guidance. We would also like thank Islam et al. for providing the data set that we have used for our work.

Finally, we have learned a lot during this project and it really encouraged us to conduct more research work in the future.

PLAGARISM REPOT

Turnitin Originality Report

Processed on: 25-Jan-2021 20:51 +06
ID: 1494053946
Word Count: 6992
Submitted: 1

Krishno__ By Hasna Hena

Similarity Index	Similarity by Source	
7%	Internet Sources:	5%
	Publications:	5%
	Student Papers:	3%

1% match (publications)
"Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)", Springer Science and Business Media LLC, 2019

1% match (Internet from 02-Feb-2019)
http://smalldatabrains.com/python-tensorflow/

< 1% match (student papers from 22-May-2017)
Submitted to University of Surrey on 2017-05-22

< 1% match (publications)
"Information and Communications Security", Springer Science and Business Media LLC, 2020

< 1% match (Internet from 16-Jul-2020)
https://people.cs.umass.edu/~amir/papers/Whitebox-SP19.pdf

< 1% match (student papers from 03-Oct-2020)
Submitted to TechKnowledge on 2020-10-03

< 1% match (publications)
Md. Aminul Islam, Md. Rasel Howlader, Umme Habiba, Rahat Hossain Faisal, Md. Mostafijur Rahman. "Indigenous Fish Classification of Bangladesh using Hybrid Features with SVM Classifier", 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), 2019

< 1% match (student papers from 03-Jun-2019)
Submitted to University of Essex on 2019-06-03

< 1% match (publications)
Mojtaba Sadeghi, Ata Akbari Asanjan, Mohammad Faridzad, Phu Nguyen, Kuolin Hsu, Soroosh Sorooshian, Dan Braithwaite. "PERSIANN-CNN: Precipitation Estimation from Remotely Sensed Information Using Artificial Neural Networks–Convolutional Neural Networks", Journal of Hydrometeorology, 2019

< 1% match (student papers from 04-Nov-2019)
Submitted to Daffodil International University on 2019-11-04

< 1% match (Internet from 05-Aug-2020)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/4093/P15437%20%2828_%29_.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 08-Dec-2020)
https://publications.waset.org/abstracts/search?page=565&q=molecular+genetic+analysis

< 1% match (publications)
Rouhan Noor, Kazi Mejbaul Islam, Md. Jakaria Rahimi. "Handwritten Bangla Numeral Recognition Using Ensembling of Convolutional Neural Network", 2018 21st International Conference of Computer and Information Technology (ICCIT), 2018

< 1% match (Internet from 06-Aug-2020)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/4095/P15433%20%2829_%29_.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 09-Nov-2020)
https://www.seebo.com/machine-learning-ai-manufacturing/

< 1% match (Internet from 04-Jan-2021)
https://openaccess.thecvf.com/content_CVPRW_2019/papers/AAMVEM
/Pedersen_Detection_of_Marine_Animals_in_a_New_Underwater_Dataset_with_CVPRW_2019_paper.pdf

< 1% match ()
http://hdl.handle.net/10344/7849

< 1% match (Internet from 23-Aug-2020)
https://mafiadoc.com/gmdh-based-networks-for-intelligent-intrusion-_59c6c8351723ddb171dadb4e.html

< 1% match (Internet from 31-Dec-2019)
https://upcommons.upc.edu/bitstream/handle/2117/127536/136621.pdf?isAllowed=y&sequence=1

< 1% match (publications)
Chi Ding, Zheng Cao, Matthew S. Emigh, Jose C. Principe et al. "Algorithmic Design and Implementation of Unobtrusive Multistatic Serial LiDAR Image", OCEANS 2019 MTS/IEEE SEATTLE, 2019

< 1% match (Internet from 01-May-2020)
https://link.springer.com/content/pdf/10.1007/978-3-030-16837-7.pdf

< 1% match ()
https://scholar.uwindsor.ca/etd/5328

< 1% match (Internet from 18-Apr-2012)
http://eusoils.jrc.ec.europa.eu/projects/scape/uploads/79/Moreno_etal.pdf

< 1% match (Internet from 12-May-2018)
https://csusmchronicle.com/7856/sports/the-heart-beat-your-guide-to-better-brain-health/

< 1% match (Internet from 17-Jan-2021)
https://academic.oup.com/jamia/article/25/10/1419/5035024?login=true

< 1% match (Internet from 29-Aug-2020)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/4181/P15391%20%2828_%29_.pdf?isAllowed=y&sequence=1

< 1% match (publications)
Long Chen, Zhihua Liu, Lei Tong, Zheheng Jiang, Shengke Wang, Junyu Dong, Huiyu Zhou. "Underwater object detection using Invert Multi-Class Adaboost with deep learning", 2020 International Joint Conference on Neural Networks (IJCNN), 2020

< 1% match (publications)
Venubabu Rachapudi, G. Lavanya Devi. "Improved convolutional neural network based histopathological image classification", Evolutionary Intelligence, 2020

< 1% match (student papers from 08-Apr-2019)
Submitted to University of Southern California on 2019-04-08

< 1% match (publications)
"Image Analysis for Moving Organ, Breast, and Thoracic Images", Springer Science and Business Media LLC, 2018

< 1% match (publications)
Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh. "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, 2006