

IMPROVED RANDOM CLASSIFIER FOR ENDANGERED SPECIES OF BIRDS USING GOOGLNET

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Baskar. D (38110073)

Chandru.S (38110100)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of D.BASKAR (38110073) and S.CHANDRU (38110100) who carried out the project entitled "entitled IMPROVED RANDOM CLASSIFIER FOR ENDANGERED SPECIES OF BIRDS USING GOOGLNET" under my supervision from November 2021 to April 2022

Internal Guide

Dr. J. CRUZ ANTONY MCA., Phd

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean, School of Computing** **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.J.CRUIZ ANTONY MCA.,Phd** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

DECLARATION

I **Baskar.D (Reg no:38110073)** and **Chandru.S (Reg no:38110100)** hereby declare that the Project Report entitled **IMPROVED RANDOM CLASSIFIER FOR ENDANGERED SPECIES OF BIRDS USING GOOGLNET** done by me under the guidance of **Dr.J.CRUIZ ANTONY MCA.,Phd** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in COMPUTER SCIENCE AND ENGINEERING

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ABSTRACT

Today, many species of birds are rarely found, and it is difficult to classify bird species when found. For example, for different scenarios, birds come with different sizes, forms, colors and from a human viewpoint with different angles. Indeed, the images show different differences that need to be recorded as audio recognition of bird species. It is also easier for people to identify birds in the pictures. Today, using deep convolutional neural network (DCNN) on Google Net framework bird species classification is possible. For this experiment, a bird image was converted into a gray scale format that generated the autograph. After examining each and every autograph that calculates the score sheet from each node and predicts the respective bird species after the score sheet analysis. In this experiment, the Caltech-UCSD Birds 200 [CUB-200-2011] data collection was used for both training and testing purposes. For training purpose 500 labeled data are used and 200 unlabeled data are used for testing. For classification, Deep Convolutional Neural Networks are used and parallel processing was carried out using GPU technology. Final results show that the DCNN algorithm can be predicted at 98.33% of bird species. The experimental research is performed on the linux operating systems with Tensor flow library and using a NVIDIA Geforce GTX 680 with 2 GB RAM.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iii
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
	LIST OF TABLES	ix
1.	INTRODUCTION	
	1.1 SYSTEM OVERVIEW	08
	1.2 OBJECTIVE OF THE PROJECT	08
2.	LITERATURE SURVEY	09
3.	EXISTING SYSTEM AND PROPOSED SYSTEM	
	1.4 EXISTING SYSTEM	12
	1.4.1 Drawbacks Of Existing System	12
	1.5 PROPOSED SYSTEM	12
	1.5.1 Advantages Of Proposed System	12
	1.6 ORGANIZATION OF THE REPORT	13
4	SYSTEM REQUIREMENTS AND SPECIFICATION	
	4.1 EXTERNAL INTERFACE REQUIREMENTS	14
	4.1.1 User Interface Requirements	14
	4.1.2 Hardware Interface	14
	4.1.3 Software Interface	14
	4.2 SYSTEM FEATURES	15
	4.3 NON- FUNCTIONAL REQUIREMENTS	15
5.	SYSTEM DESIGN	
	5.1 SYSTEM ARCHITECTURE	17
	5.2 UML DIAGRAM	18
	5.2.1 Use Case Diagram	18
	5.2.2 Activity Diagram	19
	5.2.3 Sequence Diagram	20
	5.3 SYSTEM STUDY	21

	5.3.1 Feasibility Study	21
	5.3.2 Economical Feasibility	21
	5.3.3 Technical Feasibility	21
	5.3.4 Social Feasibility	21
	5.4 DATA MODEL	22
6.	INPUT AND OUTPUT DESIGN	
	6.1 INPUT DESIGN	25
	6.2 OUTPUT DESIGN	25
7	IMPLEMENTATION	
	7.1 TECHNOLOGIES USED	27
	7.1.1 python	27
	7.1.2 Googlenet	29
	7.1.3 Cascading style sheet	30
	7.2 MODULE IMPLEMENTATION	31
	7.2.1 browse image	31
	7.2.2 Preprocessing	31
	7.2.3 Deep learning	32
	7.2.4 Classification	33
8	ALGORITHMS OF DEEP LEARNING	
	8.1 Introduction	34
	8.2 Convolutional Neural Network	35
	8.3 TEST CASES	37
	CONCLUSION AND FUTURE WORK	39
	APPENDIX	40
	REFERENCES	53

LIST OF FIGURES

S.NO	FIG.NO	TITLE	PAGE NO
1	5.1	Overall Architecture Diagram	17
2	5.2	Use Case Diagram	18
3	5.3	Activity Diagram	19
4	5.4	Sequence Diagram	20
5	5.5	Data flow Diagram-0	22
6	5.6	Data flow Diagram-1	22
7	7.1	Compilation Of Code In Python	28
8	7.2	Browse/upload image	31
9	7.3	Preprocessing	31
10	7.4	Deep learning	32
11	7.5	Classification	33
12	8.1	Neural Network Representation	35
13	8.2	Convolutional Neural Network	36
14	A1	Prediction results	51
15	A2	Classification score sheet	52

LIST OF TAB LES

Table 1 - Test cases

LIST OF ABBREVIATIONS

DCNN	-	Deep Convolutional neural network
CNN	-	Convolutional neural network
RNN	-	Residual neural network
GPU	-	Graphical processing unit
RAM	-	Random access memory

CHAPTER 1

INTRODUCTION

1.1 SYSTEM OVERVIEW

Nowadays, Identification of bird species is a difficult activity sometimes leading to uncertainty. Birds allow us to search certain organisms within the environment as they respond quickly to changes in the atmosphere (for example, the insects on which they feed) But collecting and gathering bird information requires huge efforts by humans as well as being a much more expensive method. In such situations, a robust system must be in place that will provide large-scale bird information processing and serve as a valuable resource for scholars, government agencies and so on. Consequently, naming bird species plays a significant role here for determining which species belongs to a specific image of birds. Generally; the identification of birds has done using the image, audio or video. In 2013, the IEEE International Machine Learning Workshop for Signal Processing (MLSP) declared a challenge to identify bird species.

1.2 OBJECTIVE OF THE PROJECT

In this experiment, the Caltech-UCSD Birds 200 [CUB-200-2011] data collection was used for both training and testing purposes. For training purpose 500 labeled data are used and 200 unlabeled data are used for testing. For classification, Deep Convolutional Neural Networks are used and parallel processing was carried out using GPU technology. Final results show that the DCNN algorithm can be predicted at 88.33% of bird species. The experimental research is performed on the linux operating systems with Tensor flow library and using a NVIDIA Geforce GTX 680 with 2 GB RAM.

CHAPTER 2

LITERATURE SURVEY

The author of [1] describes a convolutional neural network based deep learning approach for bird song classification that was used in an audio record-based bird identification challenge, called Bird CLEF 2016. The training and test set contained about 24k and 8.5k recordings, belonging to 999 bird species. The recorded waveforms were very diverse in terms of length and content. It converted the waveforms into frequency domain and spitted into equal segments. The segments were fed into a convolutional neural network for feature learning, which was followed by fully connected layers for classification. In the official scores our solution reached a MAP score of over 40% for main species, and MAP score of over 33% for main species mixed with background species.

The author of [2] explains one of the main problems in computer vision is the image classification problem, which is concerned with determining the presence of visual structures in an input image. Image classification analyzes the numerical properties of various image features and organizes data into categories. In recent years, many advanced classification approaches, such as artificial neural networks, fuzzy-sets, and expert systems, have been widely applied for image classification, but each of them having some problems and their accuracy level is comparatively less. One of the advanced classification approaches is to use the deep convolutional neural network (CNN) architecture of deep learning which obtains successful results in solving many machine learning problems. In this study, the image classification process is performed by using like a GOOGLE network model. The Keras library, which is regularly utilized for deep learning is used. The proposed method is trained and tested with images of cats and dogs taken from the CIFAR dataset.

The author of [3] present R-Stream TF, a polyhedral optimization tool for neural network computations. R-Stream TF transforms computations performed in a neural network graph into C programs suited to the polyhedral representation and

uses R-Stream, a polyhedral compiler, to parallelize and optimize the computations performed in the graph. R-Stream TF can exploit the optimizations available with R-Stream to generate a highly optimized version of the computation graph, specifically mapped to the targeted architecture. During our experiments, R-Stream TF was able to automatically reach performance levels close to the hand-optimized implementations, demonstrating its utility in porting neural network computations to parallel architectures.

The author of [4] Describes Now a day some bird species are being found rarely and if found classification of bird species prediction is difficult. Naturally, birds present in various scenarios appear in different sizes, shapes, colors, and angles from human perspective. Besides, the images present strong variations to identify the bird species more than audio classification. Also, human ability to recognize the birds through the images is more understandable. So this method uses the Caltech-UCSD Birds 200 [CUB-200-2011] dataset for training as well as testing purpose. By using deep convolutional neural network (DCNN) algorithm an image converted into grey scale format to generate autograph by using tensor flow, where the multiple nodes of comparison are generated. These different nodes are compared with the testing dataset and score sheet is obtained from it. After analyzing the score sheet it can predicate the required bird species by using highest score. Experimental analysis on dataset (i.e. Caltech-UCSD Birds).

The author of [5] explains in the following paper, They present an approach for fine-grained recognition based on a new part detection method. In particular, They propose a nonparametric label transfer technique which transfers part constellations from objects with similar global shapes. The possibility for transferring part annotations to unseen images allows for coping with a high degree of pose and view variations in scenarios where traditional detection models (such as deformable part models) fail. Our approach is especially valuable for fine-grained recognition scenarios where intra class variations are extremely high, and precisely localized features need to be extracted. Furthermore, They show the importance of carefully designed visual extraction strategies, such as combination

of complementary feature types and iterative image segmentation, and the resulting impact on the recognition performance. In experiments, our simple yet powerful approach achieves 35.9% and 57.8% accuracy on the CUB-2010 and 2011 bird datasets, which is the current best performance for these benchmarks.

CHAPTER 3

EXISTING AND PROPOSED SYSTEM

3.1 EXISTING SYSTEM

Consequently, naming bird species plays a significant role here for determining which species belongs to a specific image of birds. Generally; the identification of birds has done using the image, audio or video. In 2013, the IEEE International Machine Learning Workshop for Signal Processing (MLSP) declared a challenge to identify bird species. The audio processing technique allows for the detection of birds by recording the audio signal. But the processing of such information becomes more complicated because of in the environment; the mixed sounds like insects, real world objects, etc. Usually, people are more effective at find images than audios or videos. So, it is preferable to use an image over audio or video to classify birds.

3.1.1 Drawbacks of Existing System:

- They use audio file to classify the bird.
- The audio file does not give effective results.

3.2 PROPOSED SYSTEM

The proposed system classifies the bird species by using deep convolutional neural network. .For classification, deep convolutional neural network are used and parallel processing was carries out using GPU. It preprocess a developed gray scale image dataset that is used to pixel-by-pixel image recognition and image size reduction then these functions are aggregated and forward to classifier. Here In this system, google net is used to identify the bird images and predict the bird.

3.2.1 Advantages of proposed system

- This system provides accurate results.
- This system used deep convolutional neural network.
- The system uses classification score sheet to prove accuracy.

3.3ORGANISATION OF THE REPORT

The project report is organized as follows Chapter 1 deals with Introduction, System Overview, Existing System, Proposed System and Literature Survey of this project. The system requirements and hardware requirements are discussed in Chapter 2. Chapter 3 analyzes the system designs with module wise implementation. Chapter 4 discusses the implementation details. Chapter 5 explains in detail about the results produced and finally Chapter 6 gives out the conclusion along with future enhancement Chapter 7 includes screenshots and coding.

CHAPTER 4

SYSTEM REQUIREMENTS AND SPECIFICATION

4.1 EXTERNAL INTERFACE REQUIREMENTS

4.1 User Interface

The system includes a standard GUI based user interface. The user will be able to use the system to complete all expected features and the feedback information that will be displayed to the user. These are arranged as follows,

- **To browse image**

The image can be browsed from Caltech ucsc dataset .
This dataset contains more than 200 labelled data.

- **To preprocess image and deep learn**

The browsed image is given to preprocessing stage where the image is given to DCNN layers

- **To classify image**

The image is classified using the google net.

4.1.2 Hardware Interface

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Monitor : 15 inch VGA Color.
- Mouse : Logitech Mouse.
- Ram : 512 MB
- Keyboard : Standard Keyboard

4.1.3 Software Interface

- Operating System : Windows XP.
- Platform : PYTHON TECHNOLOGY
- Tool : Spyder, Python 3.5
- Front End : Anaconda
- Back End : python anaconda script

4.2 SYSTEM FEATURES

4.2.1 System Feature 1 Browse/upload image

Purpose Browse image from google

Stimulus/Response

Stimulus The stimulus is the application where the user can fetch image from Caltech ucsd.

Response The image fetched is saved in the computer.

Functional requirement

The process is based on chrome and Python and hence the requirement is confined to the application and database.

4.2.2 System Feature 2 Deep learning and classification

Purpose The purpose of Deep learning and classification is used to classify the bird species.

Stimulus/Response

Stimulus The stimulus is the application which classifies the input data to identify which bird species.

Response Then it predicts which bird species.

Functional requirement

The process is based on Googlenet and hence the requirement is confined to the application and database.

4.3 NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. **Non-functional requirements** add tremendous value to business analysis. It is commonly

misunderstood by a lot of people. It is important for business stakeholders, and Clients to clearly explain the requirements and their expectations in measurable terms. If the non-functional requirements are not measurable then they should be revised or rewritten to gain better clarity. For example, User stories help in mitigating the gap between developers and the user community in Agile Methodology.

- **Usability**

Prioritize the important functions of the system based on usage patterns. **Frequently used functions should be tested for usability**, as should complex and critical functions. Be sure to create a requirement for this.

- **Reliability**

Reliability defines the trust in the system that is developed after using it for a period of time. It defines the likeability of the software to work without failure for a given time period.

The number of bugs in the code, hardware failures, and problems can reduce the reliability of the software.

Your goal should be a long MTBF (mean time between failures). It is defined as the average period of time the system runs before failing.

Create a requirement that data created in the system will be retained for a number of years without the data being changed by the system.

It's a good idea to also include requirements that make it easier to monitor system performance.

- **Performance**

What should system response times be, as measured from any point, under what circumstances?

Are there specific peak times when the load on the system will be unusually high?

Think of stress periods, for example, at the end of the month or in conjunction with payroll disbursement.

- **Supportability**

The system needs to be **cost-effective to maintain**.

Maintainability requirements may cover diverse levels of documentation, such as system documentation, as well as test documentation, e.g. which test cases and test plans will accompany the system.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

A System Architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the **system**.

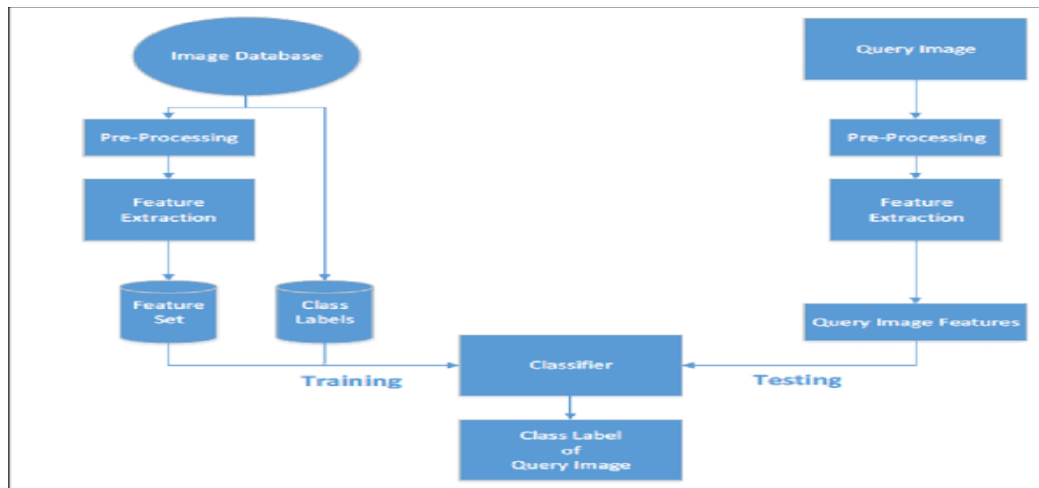


Figure 5.1 Overall architecture diagram

Figure 5.1 describes the birds classification using google net .first of all datasets are obtained from Caltech ucsc .This Caltech ucsc has 500 labelled birds data and 200 Un-labelled birds data after fetching it from this these data are given to preprocessing where the data is cleaned and removed the noisy data from it after the preprocessing the data they are given to deep learning phase here the data are learned or trained using Deep Convolutional Neural Network Techniques after that this given to google net frame work where the data are classified by using tensor-flow techniques. After that the exact bird is predicted.

5.2 UML DIAGRAMS

5.2.1 Use case diagram

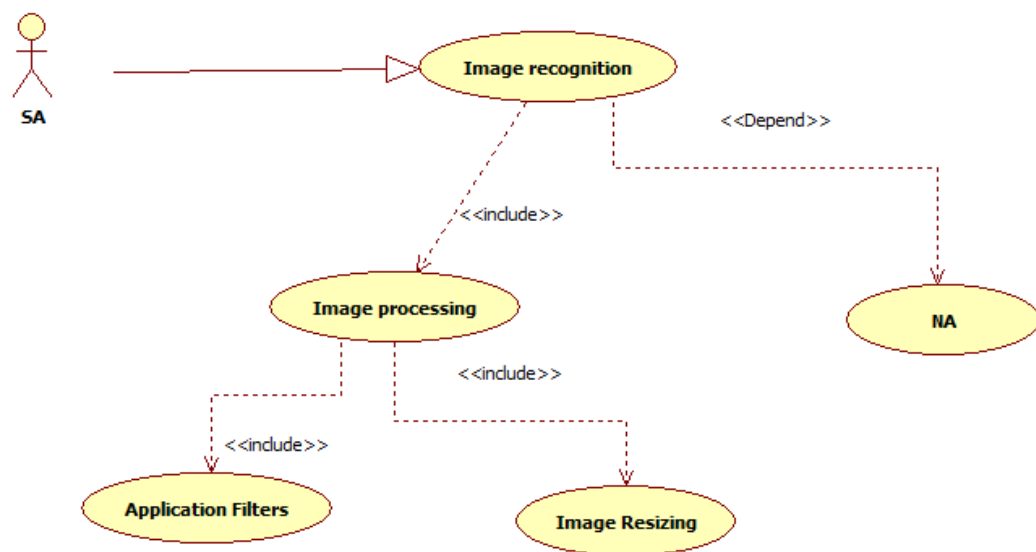


Figure 5.2 Use Case Diagram

Figure 5.2 describes that use case diagram. Here the actors are user and admin. A use case diagram in the Unified Modeling Language is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actor, their goals (represented as use cases), and any dependencies between those use cases. The use case diagram is shown in Figure 3.2

5.2.2 Activity Diagram

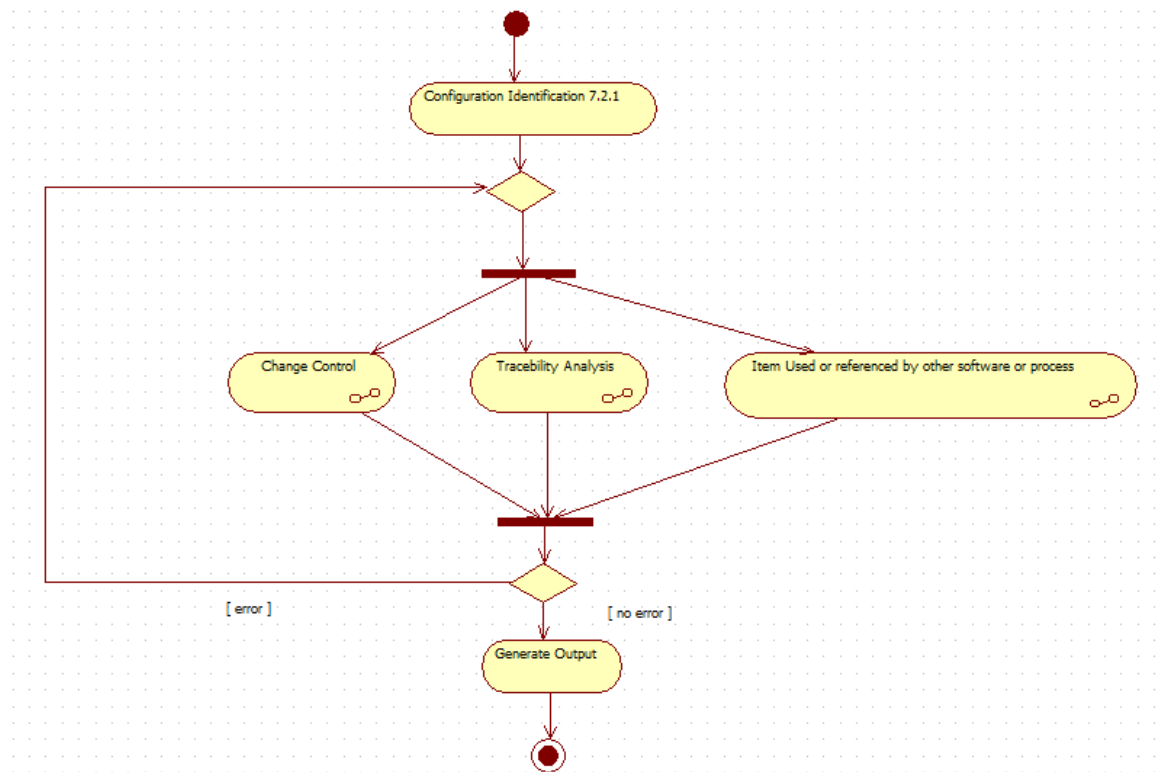


Figure 5.3 Activity Diagram

Figure 5.3 describes that activity diagram. Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to both computational and organizational process (i.e. workflows). Activity diagram shows the overall flow control. Activity diagram are constructed from a limited number of shapes connected with arrows.

5.2.3 Sequence Diagram

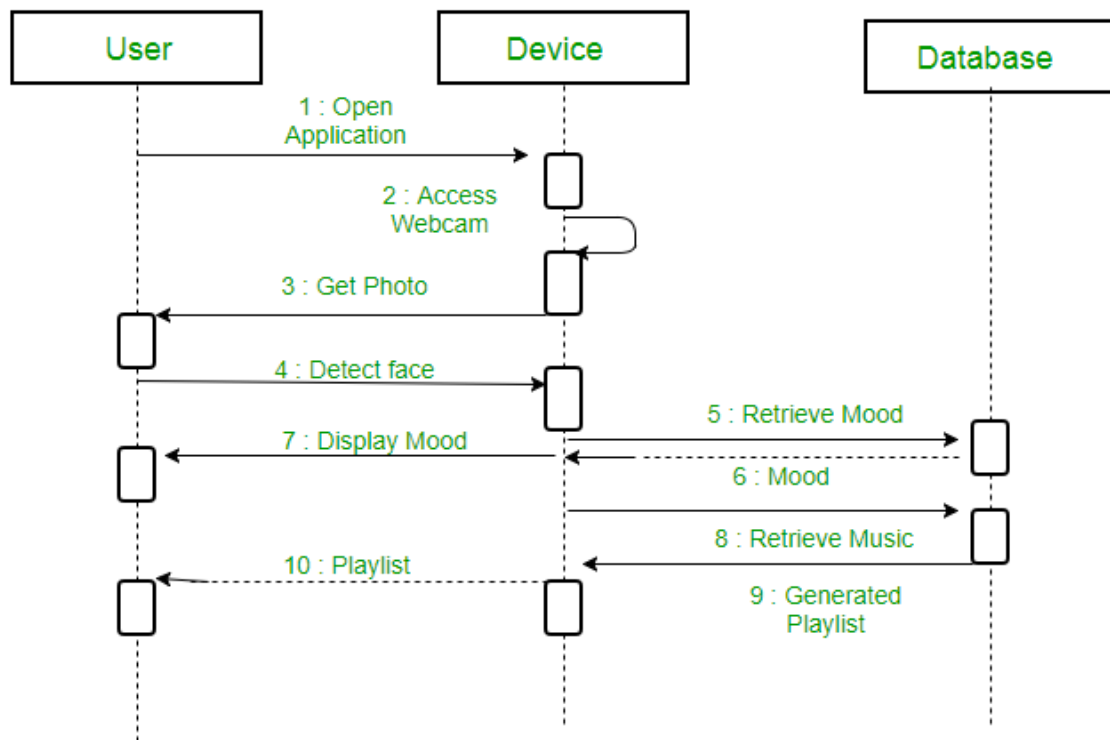


Figure 5.4 Sequence Diagram

A sequence diagram (Figure 5.4) shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or object that live simultaneously, and as horizontal arrows, the messages exchanged between them, on the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

5.3 SYSTEM STUDY

5.3.1 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. There key considerations involved in the feasibility analysis are economical feasibility, technical feasibility and social feasibility.

5.3.2 Economical Feasibility

This study is carried out to check the economic impact the system will have in the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.3.3 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. The system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will led to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.3.4 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user and this includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. This level of acceptance by the user solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. The level of confidence must be raised so that he is also able to make some constructive criticism, as he is the final user of the system.

5.4 DATA MODEL

Level 0

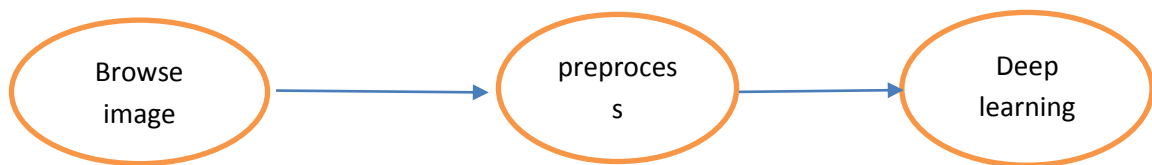


Figure 5.5 Data flow diagram – DFD0

Figure 5.5 represents that browse image from Caltech ucsc it has 500 labelled data and 200 un-labelled data. These images are taken as input and then preprocessed. In this phase, they develop a gray scale image and is used to pixel by pixel image recognition and image size reductions after that this send to deep learning.

Level 1

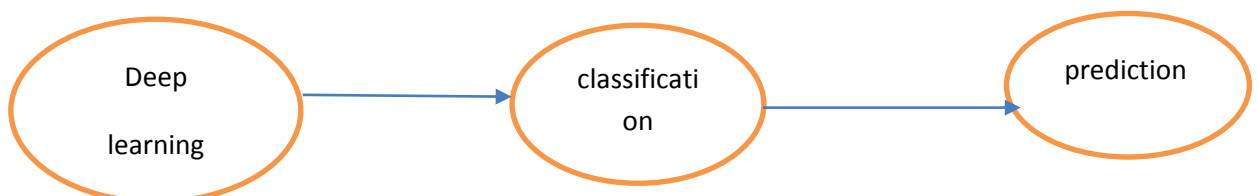


Figure 5.6 Data flow diagram – DFD1

Figure 5.6 represents that the preprocessed image is pushed into deep learning state. Here this sent to 27 different layers of convolutional neural network to gain the classification accuracy. This alignment is constructed through a deep network to extract multi functions. After deep learning stage this push to analysis

phase where Googlenet is used to identify the image. Tensor-flow produces an autograph consisting of sequence of processing nodes and retaining the dataset to achieve greater recognition accuracy.

5.4.1 Data flow diagram

The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects represented labeled arrows and transformations are represented by circles also called as bubbles. DFD is presented in hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

The DFD enables the software engineer to develop models of the information domain and functional domain at the same time. As the DFD is refined into greater levels of details, the analyst performs an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embodies the applications. A context-level DFD for the system, the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

5.4.2 Rules for DFD

- Fix the scope of the system by means of context diagrams.
- Organize the DFD so that the main sequence of the actions.
- Reads left to right and top to bottom.
- Identify all inputs and outputs.
- Identify and label each process internal to the system with rounded circles.
- The process is required for all the data transformation and transfers. Therefore, never connect a data store to a data source or the destinations or another data with just a flow arrow.

- Do not indicate hardware and ignore control information.
- Make sure the names of the processes accurately convey everything the process is done.
- There must not be unnamed process.
- Indicate external sources and destination of the data with Squares.
- Number each occurrence of repeated entities.
- Identify all data flows for each process step, except simple record retrievals.
- Labels data flows on each arrow.
- Use details flow on each arrow.
- Use the details flow arrow to indicate data movements.

CHAPTER 6

INPUT AND OUTPUT DESIGN

6.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the s.
- Methods for preparing input validations and steps to follow when error occur.
-

6.1.2 OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maze of instant. Thus the objective of input design is to create an input layout that is easy to follow

6.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

CHAPTER 7

IMPLEMENTATION

7.1 TECHNOLOGIES USED

7.1.1 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas the other languages use punctuations. It has fewer syntactical constructions than other languages.

- **Python is Interpreted**-Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive**-You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented**-Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language**-Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

7.1.1.1 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.
- Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).
- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

- Python 1.0 was released in November 1994. In 2000, Python 2.0 was released. Python 2.7.11 is the latest edition of Python 2.
- Meanwhile, Python 3.0 was released in 2008. Python 3 is not backward compatible with Python 2. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules so that "There should be one -- and preferably only one -- obvious way to do it." Python 3.5.1 is the latest version of Python 3.

7.1.1.2 Python Architecture

Python is an object oriented programming language like Java. Python is called an interpreted language. Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages. The standard implementation of python is called "c python". It is the default and widely used implementation of the Python. Python doesn't convert its code into machine code, something that hardware can understand. It actually converts it into something called byte code. So within python, compilation happens, but it's just not into a machine language. It is into byte code and this byte code can't be understood by CPU. So They need actually an interpreter called the python virtual machine. The python virtual machine executes the byte codes.

7.1.1.3 Compilation of code

The Python interpreter performs following tasks to execute a Python program :

- **Step 1** : The interpreter reads a python code or instruction. Then it verifies that the instruction is well formatted, i.e. it checks the syntax of each line. If it encounters any error, it immediately halts the translation and shows an error message.
- **Step 2** : If there is no error, i.e. if the python instruction or code is well formatted then the interpreter translates it into its equivalent form in intermediate language called "Byte code". Thus, after successful execution of Python script or code, it is completely translated into Byte code.
- **Step 3** : Byte code is sent to the Python Virtual Machine(PVM). Here again the byte code is executed on PVM .If an error occurs during this execution then the execution is halted with an error message.

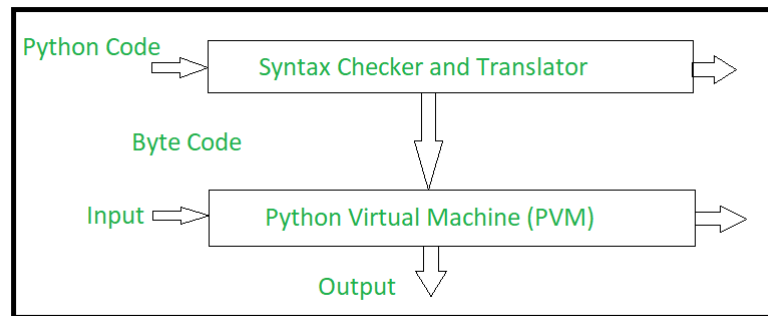


Figure 7.1 Compilation of code in Python

7.1.2 GOOGLE NET

Google Net is a 22-layer deep convolutional neural network that's a variant of the Inception Network, a Deep Convolutional Neural Network developed by researchers at Google. The Google Net architecture presented in the Image Net Large-Scale Visual Recognition Challenge 2014(ILSVRC14) solved computer vision tasks such as image classification and object detection — find out how well it performed at the conclusion section of this article. Today Google Net is used for other computer vision tasks such as face detection and recognition, adversarial training etc.

Characteristics and features of Google Net

- The input layer of the Google Net architecture takes in an image of the dimension 224 x 224.
- **Type:** This refers to the name of the current layer of the component within the architecture
- **Patch Size:** Refers to the size of the sweeping window utilized across convolutional and pooling layers. Sweeping windows have equal height and width.
- **Stride:** Defines the amount of shift the filter/sliding window takes over the input image.
- **Output Size:** The resulting output dimensions(height, width, number of feature maps) of the current architecture component after the input is passed through the layer.
- **Depth:** Refer to the number of levels/layers within an architecture component.
- **#1x1 #3x3 #5x5:** Refers to the various convolutions filters used within the inception module.
- **#3X3 reduce #5x5 reduce:** Refers to the numbers of 1x1 filters used before the convolutions.

- **Pool Proj:** This is the number of 1x1 filters used after pooling within an inception module.
- **Params:** Refers to the number of weights within the current architecture component.
- **Ops:** Refers to the number of mathematical operations carried out within the component.

Auxillary Classifiers

Before bringing the exploration of the Google Net architecture to a close, there's one more component that was implemented by the creators of the network to regularize and prevent over fitting. This additional component is known as an **Auxillary Classifier**. One main problem of an extensive network is that they suffer from vanishing gradient descent. Vanishing gradient descent occurs when the update to the weights that arises from back propagation is negligible within the bottom layers as a result of relatively small gradient value. Simply kept, the network stops learning during training .Auxillary Classifiers are added to the intermediate layers of the architecture, namely the third(Inception 4) and sixth (Inception4).Auxillary Classifiers are only utilized during training and removed during inference. The purpose of an auxiliary classifier is to perform a classification based on the inputs within the network's midsection and add the loss calculated during the training back to the total loss of the network .The loss of auxiliary classifiers is weighted, meaning the calculated loss is multiplied by 0.3.

7.1.3 Cascading Style Sheets

While XHTML is used to describe the content in a web page, it is Cascading Style Sheets (CSS) that describe how you want that content to look. In the web design biz, the way the page looks is known as its presentation. CSS is

now the official and standard mechanism for formatting text and page layouts. CSS also provides methods for controlling how documents will be presented in media other than the traditional browser on a screen, such as in print and on handheld devices. It also has rules for specifying the non-visual presentation of documents, such as how they will sound when read by a screen reader. Style sheets are also a great tool for automating production, because you can make changes to all the pages in your site by editing a single style sheet document. Style sheets are supported to some degree by all modern browsers.



by editing a single
are supported to

7.2 MODULE IMPLEMENTATION

The work is decomposed into 4 modules. They are:

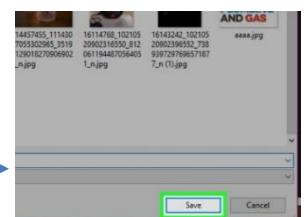
1. Browse/upload image
2. Preprocessing
3. Deep learning
4. classification

7.2.1Module 1-Browse/upload image

Caltech-UCSD .Caltech-UCSD Birds 200(CUB- 200- 2011) is a well-known bird image dataset with 200 category images. The dataset includes birds found mostly in North America. Caltech-UCSD Birds 200 consists of 11,788 pictures and annotations such as 312 binary attributes, 15 component positions, 1 bounding box.



31



7.2 Upload image

7.2.2 Module 2-Preprocessing

Pre-processing developed a gray scale image dataset that is used to pixel-by-pixel image recognition and image size reductions. Then, these functions are aggregated and forwarded to the classifier. This increased processing time while retaining quality of the image.

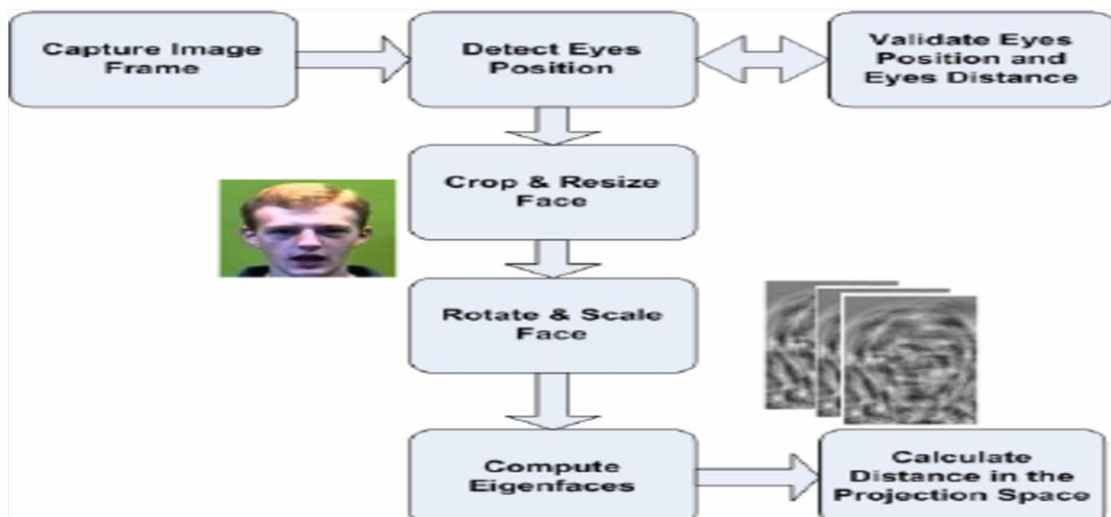


Figure 7.3 Preprocessing.

7.2.3 Module 3-Deep Learning

This input file is fed to the device and forwarded to DCNN where a suitable dataset is coupled with CNN. A DCNN is composed of different layers of convolution. Different alignment or features such as head, body, color, beak, form, and whole bird image are considered to give maximum classification

accuracy This alignment is constructed through a deep network to extract multi functions.

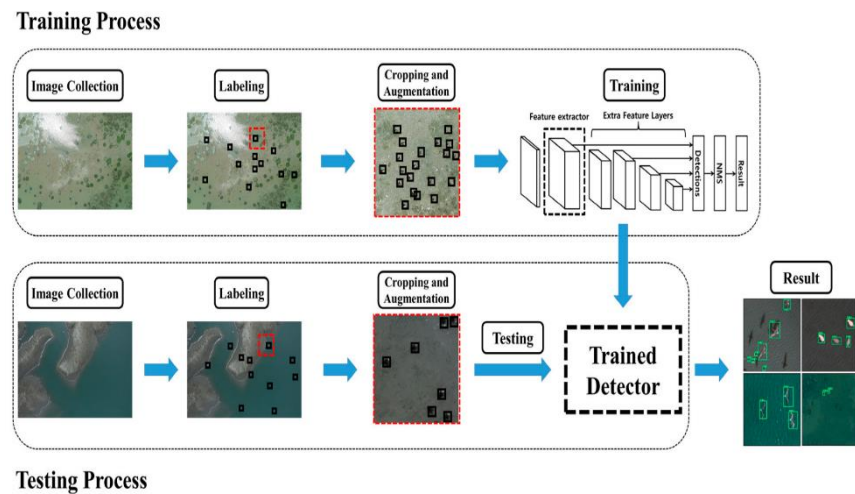


Figure 7.4 Deep learning

7.2.4 Module 4 -Classification

In this analysis, the Google Net framework has been used to identify the images. Tensorflow is a software library which is created by Google and which is an open source in nature It allows developers to monitor each neuron (node) in order to adjust the parameters to achieve the desired output. Tensorflow has a number of built-in image classification libraries. Tensorflow produces an autograph consisting of sequence of processing nodes and retraining the dataset to achieve greater recognition accuracy using retrain.py in Google.

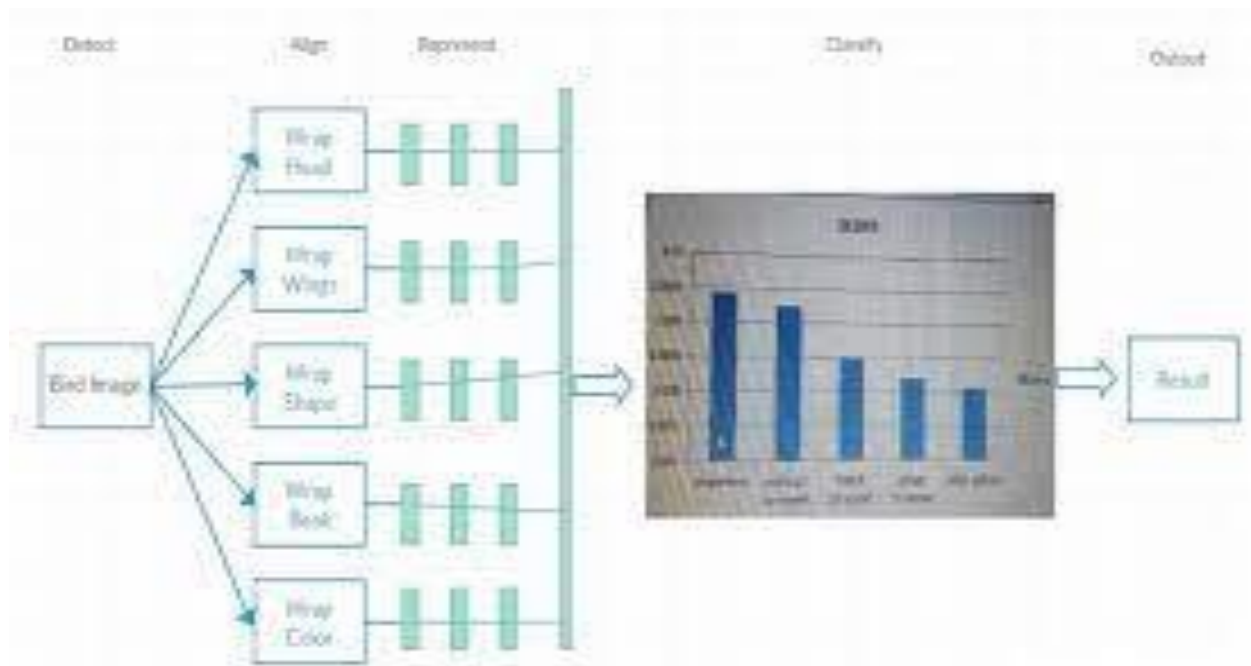


Figure 7.5 Classification

\

CHAPTER 8

ALGORITHMS OF DEEP LEARNING

8.1 INTRODUCTION

Deep learning has gained massive popularity in scientific computing, and its algorithms are widely used by industries that solve complex problems. All deep learning algorithms use different types of neural networks to perform specific tasks.

This article examines essential artificial neural networks and how deep learning algorithms work to mimic the human brain. It covers the following topics

- Deep Learning
- Defining Neural Networks

Deep Learning

Deep learning uses artificial neural networks to perform sophisticated computations on large amounts of data. It is a type of machine learning that works based on the structure and function of the human brain.

Deep learning algorithms train machines by learning from examples. Industries such as health care, e-Commerce, entertainment, and advertising commonly use deep learning.

Defining Neural Networks

A Neural Network is structured like the human brain and consists of artificial neurons, also known as nodes. These nodes are stacked next to each other in three layers:

- The input layer
- The hidden layer(s)
- The output layer

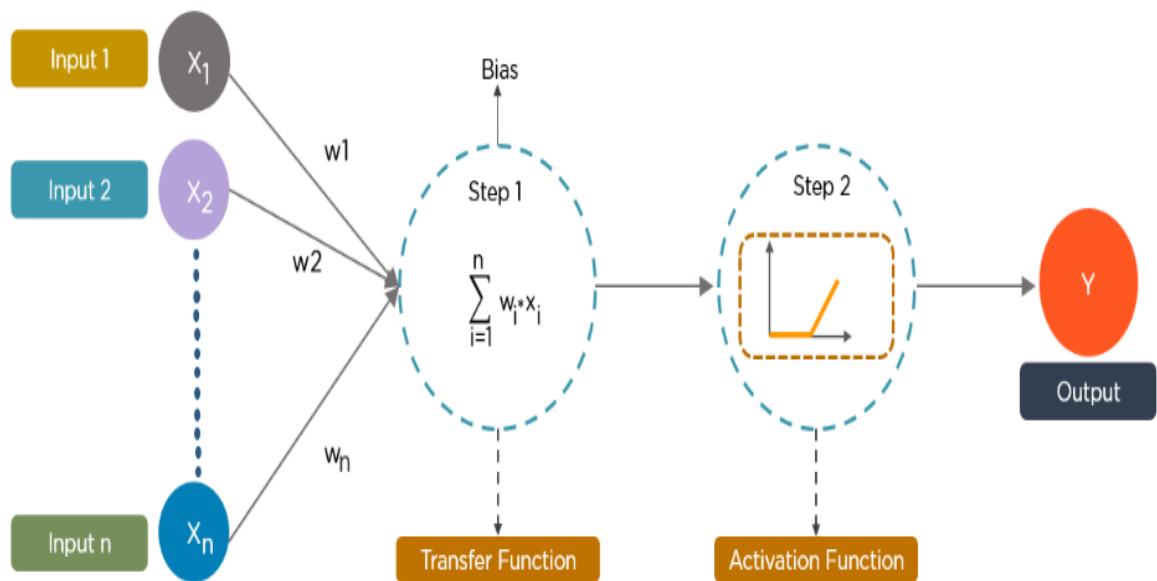


Figure 8.1 Neural Network Representation

Data provides each node with information in the form of inputs. The node multiplies the inputs with random weights, calculates them, and adds a bias. Finally, nonlinear functions, also known as activation functions, are applied to determine which neuron to fire.

Deep Learning Algorithms

While deep learning algorithms feature self-learning representations, they depend upon ANNs that mirror the way the brain computes information. During the training process, algorithms use unknown elements in the input distribution to extract features, group objects, and discover useful data patterns. Much like training machines for self-learning, this occurs at multiple levels, using the algorithms to build the models.

Deep learning models make use of several algorithms. While no one network is considered perfect, some algorithms are better suited to perform specific tasks. To choose the right ones, it's good to gain a solid understanding of all primary algorithms

1. Convolutional Neural Networks (CNNs)

CNN also known as Convolutional Nets, consist of multiple layers and are mainly used for image processing and object detection. Yann Le Cun developed the first CNN in 1988 when it was called Le Net. It was used for recognizing characters like ZIP codes and digits.

CNNs Work

CNN's have multiple layers that process and extract features from data:

Convolution Layer

- CNN has a convolution layer that has several filters to perform the convolution operation.

Rectified Linear Unit (ReLU)

- CNN's have a ReLU layer to perform operations on elements. The output is a rectified feature map.

Pooling Layer

- The rectified feature map next feeds into a pooling layer. Pooling is a down-sampling operation that reduces the dimensions of the feature map.
- The pooling layer then converts the resulting two-dimensional arrays from the pooled feature map into a single, long, continuous, linear vector by flattening it.

Fully Connected Layer

- A fully connected layer forms when the flattened matrix from the pooling layer is fed as an input, which classifies and identifies the images.

Below is an example of an image processed via CNN.

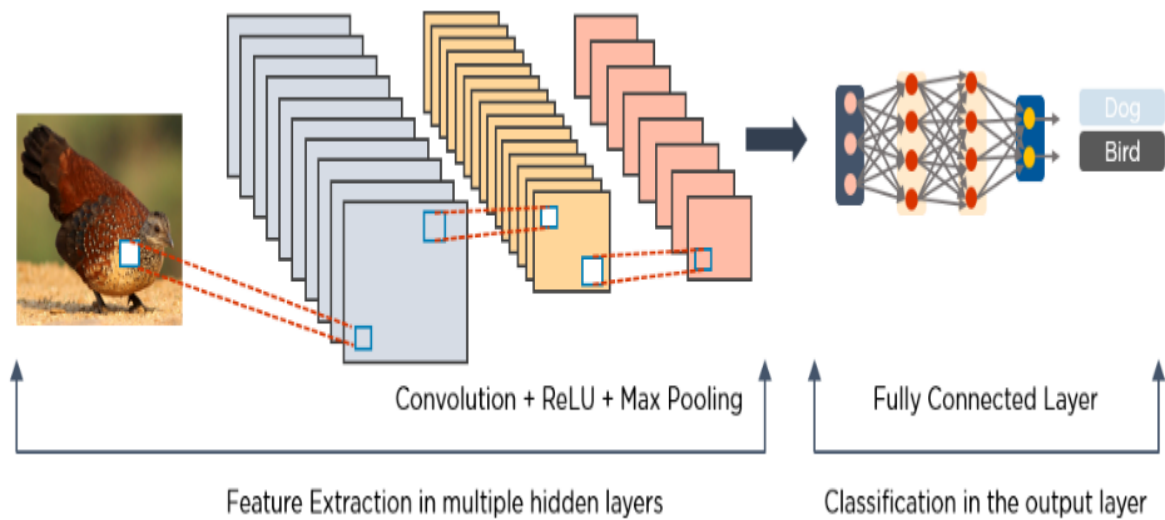


Figure 8.2 Convolutional Neural Network

8.4 TESTCASES

Species	Gaussian Noise	Gaussian Blur	Flip	Eye	Beak	Hue	Sharp	Add	Affine	Total accuracy
Eagle	yes	Yes	Yes	Yes	Yes	No	No	No	No	90
Emu	Yes	Yes	Yes	Yes	Yes	Yes	yes	yes	yes	78
Brown pelica	Yes	Yes	Yes	Yes	Yes	Yes	Yes	yes	Yes	65
Condor	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	91
Mute Swan	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	91
Turul	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	yes	78

Garuda	Yes	Yes	Yes	Yes	Yes	Yes	yes	No	No	80
Lapwing	Yes	Yes	Yes	Yes	Yes	No	No	No	No	99
Hoopoe	Yes	Yes	Yes	Yes	Yes	No	No	No	No	100
Caracara	Yes	Yes	Yes	Yes	Yes	No	No	No	No	81
Kiwi	Yes	Yes	Yes	Yes	Yes	No	No	No	No	81
Chukar	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	78
Jungle fowl	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	78
Blue crane	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	yes	78
Cuban Tara	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	80
Owl	yes	yes	Yes	Yes	yes	yes	yes	No	No	80

CONCLUSION AND FUTURE WORK

CONCLUSION

This research examined the use of deep learning algorithms to address the problem of identification of bird species. The framework used Deep learning algorithm (Unsupervised Learning) for image classification on the dataset (Caltech-UCSD Birds 200). It consists of 200 categories or 11,788 photos. The CUB-200-2011 dataset consisted of mixed parameters such as pose, wings, color, beak, arms, etc. The initial study of the 200 images of Caltech-UCSD Birds for these species could then be carried out using the findings as test images. The experiment was able to achieve 90.93 percent by fine-tuning the images in Tensor-flow and trained on the Google Net dataset and on the Caltech-UCSD Birds 200 dataset, which included 11,788 images, the classification average accuracy. Future research will concentrate on developing Architecture for Real Time Smartphone.

FUTURE WORK

Future research will concentrate on developing Architecture for Real Time Smartphone.

CODING

Front end(Web page designing)

```
.img-preview {  
    width: 256px;  
    height: 256px;  
    position: relative;  
    border: 5px solid #F8F8F8;  
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);  
    margin-top: 1em;  
    margin-bottom: 1em;  
}
```

```
.img-preview>div {  
    width: 100%;  
    height: 100%;  
    background-size: 256px 256px;  
    background-repeat: no-repeat;  
    background-position: center;  
}
```

```
input[type="file"] {  
    display: none;  
}
```

```
.upload-label{  
    display: inline-block;  
    padding: 12px 30px;  
    background: #39D2B4;  
    color: #fff;  
    font-size: 1em;  
    transition: all .4s;  
    cursor: pointer;  
}
```

```
.upload-label:hover{  
  
    background: #34495E;  
  
    color: #39D2B4;  
  
}
```

```
.loader {  
  
    border: 8px solid #f3f3f3; /* Light grey */  
  
    border-top: 8px solid #3498db; /* Blue */  
  
    border-radius: 50%;  
  
    width: 50px;  
  
    height: 50px;  
  
    animation: spin 1s linear infinite;  
  
}
```

```
@keyframes spin {  
  
    0% { transform: rotate(0deg); }  
  
    100% { transform: rotate(360deg);
```

Image classification coding in python

```
from __future__ import division, print_function  
  
# coding=utf-8  
  
import sys  
  
import os
```

```

import glob

import re

import numpy as np


# Keras

import torch

import torchvision

import torch.nn as nn

from PIL import Image

import torchvision.transforms as transforms

import torchvision.models as models


def load_checkpoint(filepath):

    checkpoint = torch.load(filepath, map_location = 'cpu')

    model = checkpoint['model']

    model.load_state_dict(checkpoint['state_dict'])

    for parameter in model.parameters():

        parameter.requires_grad = False


    model.eval()

    return model

model = load_checkpoint("bird_species.pth")

```

Flask utils

```
from flask import Flask, redirect, url_for, request, render_template
```

```
from werkzeug.utils import secure_filename
```

```
from event.pywsgi import WSGIServer
```

classe = ['AFRICAN FIREFINCH The African firefinch, also called the dark firefinch or blue-billed firefinch due to the color of its bill, is a common species of estrildid finch found in almost all parts of Africa. Scientific name: Lagonosticta rubricata Conservation status: Least Concern (Population stable) Kingdom: Animalia', 'ALBATROSS Albatrosses are very large seabirds in the family Diomedidae. They range widely in the Southern Ocean and the North Pacific. They are absent from the North Atlantic, although fossil remains show they once occurred there and occasional vagrants are found. Scientific name: Diomedidae Higher classification: Petrels', 'ALEXANDRINE PARAKEET The Alexandrine parakeet, also known as the Alexandrine parrot, Alexandrine ring-necked parakeet or greater rose-ringed parakeet is a medium-sized parrot in the genus Psittacula of the family Psittaculidae. Scientific name: Psittacula eupatria Higher classification: Rose-ringed Parakeets', 'AMERICAN AVOCET The American avocet is a large wader in the avocet and stilt family, Recurvirostridae. It spends much of its time foraging in shallow water or on mud flats, often sweeping its bill from side to side in water as it seeks its crustacean and insect prey. Scientific name: Recurvirostra americana Higher classification: Avocets ', 'AMERICAN BITTERN', 'AMERICAN COOT', 'AMERICAN GOLDFINCH', 'AMERICAN KESTREL', 'AMERICAN PIPIT', 'AMERICAN REDSTART', 'ANHINGA', 'ANNAS HUMMINGBIRD', 'ANTBIRD', 'ARARIPE MANAKIN', 'ASIAN CRESTED IBIS', 'BALD EAGLE The bald eagle is a bird of prey found in North America. A sea eagle, it has two known subspecies and forms a species pair with the white-tailed eagle. Its range includes most of Canada and Alaska, all of the contiguous United States, and northern Mexico. Scientific name: Haliaeetus leucocephalus Conservation status: Least Concern (Population increasing) Speed: 120 – 160 km/h (Diving) Wingspan: 1.8 – 2.3 m (Adult)', 'BALI STARLING', 'BALTIMORE ORIOLE', 'BANANAQUIT', 'BAR-TAILED GODWIT', 'BARN OWL', 'BARN SWALLOW', 'BARRED PUFFBIRD', 'BAY-BREASTED WARBLER', 'BEARDED BARBET', 'BELTED KINGFISHER', 'BIRD OF

PARADISE', 'BLACK FRANCOLIN', 'BLACK SKIMMER', 'BLACK SWAN', 'BLACK THROATED WARBLER', 'BLACK VULTURE', 'BLACK-CAPPED CHICKADEE', 'BLACK-NECKED GREBE', 'BLACK-THROATED SPARROW', 'BLACKBURNIAM WARBLER', 'BLUE GROUSE', 'BLUE HERON', 'BOBOLINK', 'BROWN NOODY', 'BROWN THRASHER', 'CACTUS WREN', 'CALIFORNIA CONDOR', 'CALIFORNIA GULL The California gull, or seagull, is a medium-sized gull, smaller on average than the herring gull but larger on average than the ring-billed gull, though it may overlap in size greatly with both. Scientific name: *Larus californicus* Mass: 680 g Length: 51 – 58 cm', 'CALIFORNIA QUAIL', 'CANARY', 'CAPE MAY WARBLER', 'CAPUCHINBIRD', 'CARMINE BEE-EATER', 'CASPIAN TERN', 'CASSOWARY', 'CHARA DE COLLAR', 'CHIPPING SPARROW', 'CHUKAR PARTRIDGE', 'CINNAMON TEAL', 'COCK OF THE ROCK', 'COCKATOO', 'COMMON GRACKLE', 'COMMON HOUSE MARTIN', 'COMMON LOON', 'COMMON POORWILL', 'COMMON STARLING', 'COUCHS KINGBIRD', 'CRESTED AUKLET', 'CRESTED CARACARA', 'CROW', 'CROWNED PIGEON', 'CUBAN TODY', 'CURL CRESTED ARACURI', 'D-ARNAUDS BARBET', 'DARK EYED JUNCO The dark-eyed junco is a species of junco, a group of small, grayish New World sparrows. This bird is common across much of temperate North America and in summer ranges far into the Arctic. It is a very variable species, much like the related fox sparrow, and its systematics are still not completely untangled. Scientific name: *Junco hyemalis* Higher classification: Juncos Mass: 19 g (Adult) Family: Passerellidae Length: 12 – 16 cm', 'DOWNY WOODPECKER', 'EASTERN BLUEBIRD', 'EASTERN MEADOWLARK', 'EASTERN ROSELLA', 'EASTERN TOWEE The eastern towhee is a large New World sparrow. The taxonomy of the towhees has been under debate in recent decades, and formerly this bird and the spotted towhee were considered a single species, the rufous-sided towhee. Their breeding habitat is brushy areas across eastern North America. Scientific name: *Pipilo erythrophthalmus* Higher classification: Pipilo', 'ELEGANT TROGON', 'ELLIOTS PHEASANT', 'EMPEROR PENGUIN', 'EMU', 'EURASIAN MAGPIE', 'EVENING GROSBEAK', 'FLAME TANAGER The flame-colored tanager, formerly known as the stripe-backed tanager, is a medium-sized American songbird. Formerly placed in the tanager family Thraupidae, other members of its genus and it are now classified in the cardinal family Cardinalidae. Higher classification: Northern tanagers Scientific name: *Piranga bidentata*

Conservation status: Least Concern', 'FLAMINGO', 'FRIGATE', 'GAMBELS QUAIL', 'GILA WOODPECKER', 'GILDED FLICKER', 'GLOSSY IBIS The glossy ibis is a water bird in the order Pelecaniformes and the ibis and spoonbill family Threskiornithidae. The scientific name derives from Ancient Greek plegados and Latin, falcis, both meaning "sickle" and referring to the distinctive shape of the bill. Scientific name: Plegadis falcinellus Higher classification: Plegadis', 'GOLD WING WARBLER', 'GOLDEN CHEEKED WARBLER', 'GOLDEN CHLOROPHONIA', 'GOLDEN EAGLE', 'GOLDEN PHEASANT', 'GOLDEN PIPIT', 'GOULDIAN FINCH', 'GRAY CATBIRD', 'GRAY PARTRIDGE', 'GREEN JAY', 'GREY PLOVER', 'GUINEAFOWL', 'GYRFALCON', 'HARPY EAGLE', 'HAWAIIAN GOOSE', 'HOODED MERGANSER', 'HOOPOES', 'HORNBILL', 'HORNED GUAN', 'HORNED SUNGEM', 'HOUSE FINCH', 'HOUSE SPARROW', 'IMPERIAL SHAQ', 'INCA TERN', 'INDIAN BUSTARD The great Indian bustard or Indian bustard is a bustard found on the Indian subcontinent. A large bird with a horizontal body and long bare legs, giving it an ostrich like appearance, this bird is among the heaviest of the flying birds. Scientific name: Ardeotis nigriceps Higher classification: Ardeotis', 'INDIGO BUNTING The indigo bunting is a small seed-eating bird in the cardinal family, Cardinalidae. It is migratory, ranging from southern Canada to northern Florida during the breeding season, and from southern Florida to northern South America during the winter. Scientific name: Passerina cyanea, Higher classification: North American buntings, Family: Cardinalidae', 'JABIRU The jabiru is a large stork found in the Americas from Mexico to Argentina, except west of the Andes. It sometimes wanders into the United States, usually in Texas, but has been reported as far north as Mississippi. It is most common in the Pantanal region of Brazil and the Eastern Chaco region of Paraguay. Scientific name: Jabiru mycteria Family: Ciconiidae', 'JAVAN MAGPIE', 'KAKAPO', 'KILLDEAR', 'KING VULTURE The king vulture is a large bird found in Central and South America. It is a member of the New World vulture family Cathartidae. This vulture lives predominantly in tropical lowland forests stretching from southern Mexico to northern Argentina. Scientific name: Sarcoramphus papa Higher classification: Sarcoramphus', 'KIWI', 'KOOKABURRA', 'LARK BUNTING The lark bunting is a medium-sized American sparrow native to central and western North America. It is also the state bird of Colorado. Scientific name: Calamospiza melanocorys Kingdom: Animalia', 'LEARS MACAW', 'LILAC ROLLER', 'LONG-EARED OWL',

'MALABAR HORNBILL', 'MALACHITE KINGFISHER' The malachite kingfisher is a river kingfisher which is widely distributed in Africa south of the Sahara. It is largely resident except for seasonal climate-related movements. Scientific name: *Corythornis cristatus* Conservation status: Least Concern', 'MALEO', 'MALLARD DUCK', 'MANDRIN DUCK', 'MARABOU STORK', 'MASKED BOOBY', 'MIKADO PHEASANT', 'MOURNING DOVE', 'MYNA', 'NICOBAR PIGEON', 'NORTHERN CARDINAL', 'NORTHERN FLICKER', 'NORTHERN GANNET', 'NORTHERN GOSHAWK', 'NORTHERN JACANA', 'NORTHERN MOCKINGBIRD', 'NORTHERN PARULA', 'NORTHERN RED BISHOP', 'OCELLATED TURKEY', 'OKINAWA RAIL', 'OSPREY', 'OSTRICH', 'PAINTED BUNTIG', 'PALILA', 'PARADISE TANAGER', 'PARUS MAJOR', 'PEACOCK', 'PELICAN', 'PEREGRINE FALCON', 'PHILIPPINE EAGLE', 'PINK ROBIN', 'PUFFIN', 'PURPLE FINCH', 'PURPLE GALLINULE', 'PURPLE MARTIN', 'PURPLE SWAMPHEN' The western swamphen is a swamphen in the rail family Rallidae, one of the six species of purple swamphen. From the French name talève sultane, it is also known as the sultana bird. This chicken-sized bird, with its large feet, bright plumage and red bill and frontal shield is easily recognisable in its native range. Scientific name: *Porphyrio porphyrio* Mass: 770 g Length: 51 cm', 'QUETZAL', 'RAINBOW LORIKEET', 'RAZORBILL', 'RED FACED CORMORANT', 'RED FACED WARBLER', 'RED HEADED DUCK', 'RED HEADED WOODPECKER', 'RED HONEY CREEPER', 'RED THROATED BEE EATER', 'RED WINGED BLACKBIRD', 'RED WISKERED BULBUL', 'RING-NECKED PHEASANT', 'ROADRUNNER', 'ROBIN', 'ROCK DOVE', 'ROSY FACED LOVEBIRD', 'ROUGH LEG BUZZARD', 'RUBY THROATED HUMMINGBIRD', 'RUFIOUS KINGFISHER', 'RUFUOS MOTMOT', 'SAND MARTIN', 'SCARLET IBIS', 'SCARLET MACAW', 'SHOEBILL', 'SMITHS LONGSPUR', 'SNOWY EGRET', 'SNOWY OWL', 'SORA', 'SPANGLED COTINGA', 'SPLENDID WREN', 'SPOON BILED SANDPIPER', 'SPOONBILL', 'STEAMER DUCK', 'STORK BILLED KINGFISHER', 'STRAWBERRY FINCH', 'STRIPPED SWALLOW', 'SUPERB STARLING', 'TAIWAN MAGPIE', 'TAKAHE', 'TASMANIAN HEN', 'TEAL DUCK', 'TIT MOUSE', 'TOUCHAN', 'TOWNSENDS WARBLER', 'TREE SWALLOW', 'TRUMPTER SWAN', 'TURKEY VULTURE', 'TURQUOISE MOTMOT', 'VARIED THRUSH', 'VENEZUELIAN TROUPIAL', 'VERMILION FLYCATHER', 'VIOLET GREEN SWALLOW', 'WATTLED CURASSOW', 'WHIMBREL', 'WHITE CHEEKED

```
TURACO', 'WHITE NECKED RAVEN', 'WHITE TAILED TROPIC', 'WILD  
TURKEY', 'WILSONS BIRD OF PARADISE', 'WOOD DUCK', 'YELLOW  
CACIQUE', 'YELLOW HEADED BLACKBIRD']
```

Define a flask app

```
app = Flask(__name__)
```

Process our image

```
def process_image(image_path):
```

```
    img = Image.open(image_path)
```

```
    width, height = img.size
```

```
    img = img.resize((255, int(255*(height/width))) if width < height else  
(int(255*(width/height)), 255))
```

```
    width, height = img.size
```

```
    left = (width - 224)/2
```

```
    top = (height - 224)/2
```

```
    right = (width + 224)/2
```

```
    bottom = (height + 224)/2
```

```
    img = img.crop((left, top, right, bottom))
```

```
    img = np.array(img)
```

```
    img = img.transpose((2, 0, 1))
```

```
    img = img/255
```

```
    img[0] = (img[0] - 0.5)/0.5
```

```
    img[1] = (img[1] - 0.5)/0.5
```

```

img[2] = (img[2] - 0.5)/0.5

img = img[np.newaxis,:]

image = torch.from_numpy(img)

image = image.float()

return image

```

Using our model to predict the label

```

def predict(image, model):

    output = model(image)

    output = torch.exp(output)

    probs, classes = output.topk(1, dim=1)

    k = classes.item()

    return classe[k]

```

```

def model_predict(img_path, model):

    img = process_image(img_path)

    pred = predict(img, model)

    return pred

```

```

@app.route('/', methods=['GET'])

```

```

def index():

    # Main page

    return render_template('index.html')

```

```

@app.route('/predict', methods=['GET', 'POST'])

def upload():

    if request.method == 'POST':

        # Get the file from post request

        f = request.files['file']

        # Save the file to ./uploads

        basepath = os.path.dirname(__file__)

        file_path = os.path.join(

            basepath, 'uploads', secure_filename(f.filename))

        f.save(file_path)

        pred = model_predict(file_path, model)

        return pred

    return None

if __name__ == '__main__':

    #app.run(host = '0.0.0.0', port = 8080)

    app.run(debug = True)

```

APPENDIX

Hostname: andres-XPS-15-9570

edgeIQ: v0.11.0

Status: Running

Output

Model:

alwaysai/shufflenet_1x_g3

Inference time: 0.005 s

1. label: **great grey owl**

confidence: 16.942

2. label: leopard, confidence:

11.815

3. label: ruffed grouse,

confidence: 10.27

4. label: ptarmigan, confidence:

10.129

5. label: junco, confidence:

8.875



Figure A1. Prediction results

	Accuracy %																	
	Sylvia borin	Sylvia nisoria	Sylvia atricapilla	Parus ater	Parus caeruleus	Parus major	Corvus corax	Corvus corone	Nucifraga caryocatactes	Streptopelia decaocto	Columba oenas	Columba palumbus	Buteo buteo	Pernis apivorus	Accipiter gentilis	Anser anser	Anser albifrons	Anser erythropus
Sylvia borin	72	45	0	0	5	8	0	0	0	0	0	0	0	0	3	0	0	6
Sylvia nisoria	24	50	0	0	2	4	0	0	0	0	0	0	0	0	0	0	0	1
Sylvia atricapilla	0	0	49	13	0	0	0	0	0	0	0	0	0	15	0	0	0	0
Parus ater	0	0	0	159	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Parus caeruleus	0	1	0	0	74	12	0	0	0	0	0	0	0	0	0	0	0	0
Parus major	1	19	0	0	17	40	0	0	0	0	0	0	0	0	0	0	0	0
Corvus corax	0	0	0	0	0	0	109	5	0	0	0	0	0	0	0	0	12	0
Corvus corone	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	2	0
Nucifraga caryocatactes	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0
Streptopelia decaocto	0	0	0	0	0	0	0	0	0	58	0	0	0	0	0	0	0	0
Columba oenas	0	0	0	0	0	0	0	0	0	0	39	1	0	1	0	2	0	0
Columba palumbus	0	0	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0
Buteo buteo	0	0	16	1	0	0	0	0	0	0	1	1	37	6	0	0	0	0
Pernis apivorus	0	0	0	0	0	0	0	0	0	0	1	1	10	0	0	0	0	0
Accipiter gentilis	1	1	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0
Anser anser	0	0	0	0	0	0	0	0	0	0	10	8	0	4	0	63	0	0
Anser albifrons	0	0	0	0	0	0	5	4	0	0	0	0	0	0	0	0	64	8
Anser erythropus	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	4	45
Average Accuracy																		81,22694

Figure A2. Classification score sheet

REFERENCES

- [1] Tóth, B.P. and Czeba, B., 2016, September. Convolutional Neural Networks for Large-Scale Bird Song Classification in Noisy Environment. In CLEF (Working Notes) (pp. 560-568).
- [2] Gavali, Pralhad, and J. Saira Banu. "Deep Convolutional Neural Network for Image Classification on CUDA Platform." In Deep Learning and Parallel Computing Environment for Bioengineering Systems, pp. 99-122. Academic Press, 2019
- [3] Pradelle, B., Meister, B., Baskaran, M., Springer, J. and Lethin., 2017, November. Polyhedral Optimization of Tensorflow Computation Graphs. In 6th Workshop on Extreme scale Programming Tools (ESPT-2017) at The International Conference for High Performance Computing, Networking, Storage and Analysis (SC17).
- [4] Prof. Pralhad Gavali, Ms. Prachi Abhijeet Mhetre Bird Species Identification using Deep Learning International Journal of Engineering Research & Technology (IJERT) ISSN: 2278- 0181 Vol. 8 Issue 04, April-2019 pp 68-72
- [5] Goering, C., Rodner, E., Freytag, A., Denzler, J., "Nonparametric Part Transfer for Fine-grained Recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2014

