Project documentation

Digital Naturalist - AI Enabled Tool For Biodiversity Researchers

1. INTRODUCTION

1.1 Project Overview

In this project, we are creating a web application which uses a deep learning model, trained on different species of birds, flowers and mammals (3 subclasses in each for a quick understanding and get the prediction of the bird when an image is given).

1.2 Purpose

When venturing into the woods, field naturalists usually rely on common approaches like always carrying a guidebook around everywhere or seeking help from experienced ornithologists. There should be a handy tool for them to capture, identify and share the beauty to the outside world.

2.LITERATURE SURVEY

2.1 Existing Problem

Nature is a risky sport that requires a lot of patience and caution.Naturalists who are trained and have sufficient knowledge about the visual differences to identify dangerous and non-dangerous Species as opposed to travellers and common tourists who are prone to danger.Thus to explore nature safely we require a modern-age technology

2.2 References

1)**Detection of Birds in the Wild using Deep Learning Methods**. Author: Prathamesh Datar, Kashish Jain, Bhavin Dhedh. Publication: 2018 4th International Conference for Convergence in Technology (I2CT) SDMIT Ujiri, Mangalore, India. Oct 27-28, 2018. Methodology: Object detection and localization is one of the prominent applications of the computer vision. The paper presents a comparative study of state-of-the-art deep learning methods - YOLOv2, YOLOv3 and Mask R-CNN, for detection of birds in the wild. Detection of birds is an important problem across multiple applications including the aviation safety, avian protection and ecological science of migrant bird species.

2)**Research on Artificial Intelligence: Deep Learning to Identify Plant Species**. Author: JiaDong Guo Publication:2022 International Conference on Machine Learning and Knowledge Engineering (MLKE). Methodology: The machine Learning became a popular subject, especially in object recognition area. Aiming at providing a faster and more accurate plant species recognition program, the author introduced a deep learning and CNN, and decided to build a CNN project with PyCharm, anaconda, Kera to find the best way to improve recognition program accuracy and recognition speed.

3)**Inception-v3 for Flower Classification.** Author: Xiaoling Xia, Cui Xu’, Bing Nan Publication: 2017 2nd International Conference on Image, Vision and Computing Methodology: The study of flower classification system is a very important subject in the field of Botany. A classifier of flowers with high accuracy will also bring a lot of fun to people's lives. However, because of the complex background of flowers, the similarity between the different species of flowers, and the differences among the same species of flowers, there are still some challenges in the recognition of flower images.

4)**Deep barcoding: Deep learning for species classification using DNA barcoding.** Author: Cheng-Hong Yang, Kuo-Chuan Wu, Li-Yeh Chuang, and Hsueh-Wei Chang Publication: IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL. 19, NO. 4, JULY/AUGUST 2022 Methodology: DNA barcodes with short sequence fragments are used for species identification. Because of advances in sequencing technologies, DNA barcodes have gradually been emphasized. DNA sequences from different organisms are easily and rapidly acquired. Therefore, DNA sequence analysis tools play an increasingly crucial role in species identification. This study proposed deep barcoding, a deep learning framework for species classification by using DNA barcodes.

5) **An Improved Image Classification Based In Feature Extraction From Convolutional Neural Network: Application To Flower Classification** Author: Faeze Sadati, Behrooz Rezaei Publication: 2021 12th International Conference on Information and Knowledge Technology (IKT) Methodology: The convolutional neural network (CNN) is applied classify flower. the pretrained CNN models in which classification part is removed and instead of it, we use global average pooling (GAP) in the last layer for extracting their features The features obtained from these models are concatenated, and then we use a support vector machine (SVM) as classifier for the flower classification

6) **An Improved faster RCNN marine fish classification identification algorithm.** Author: Yuhang Li, Daqi Zhu, HaoDong Fan Publication: 2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE) Methodology: The algorithm first selects residual network (Resnet) with strong feature extraction capability for feature extraction; then generates candidate target regions through 12 different Anchors to further improve the accuracy of detection; finally, the resulting features are transmitted to two subnetworks to achieve classification and positioning respectively.

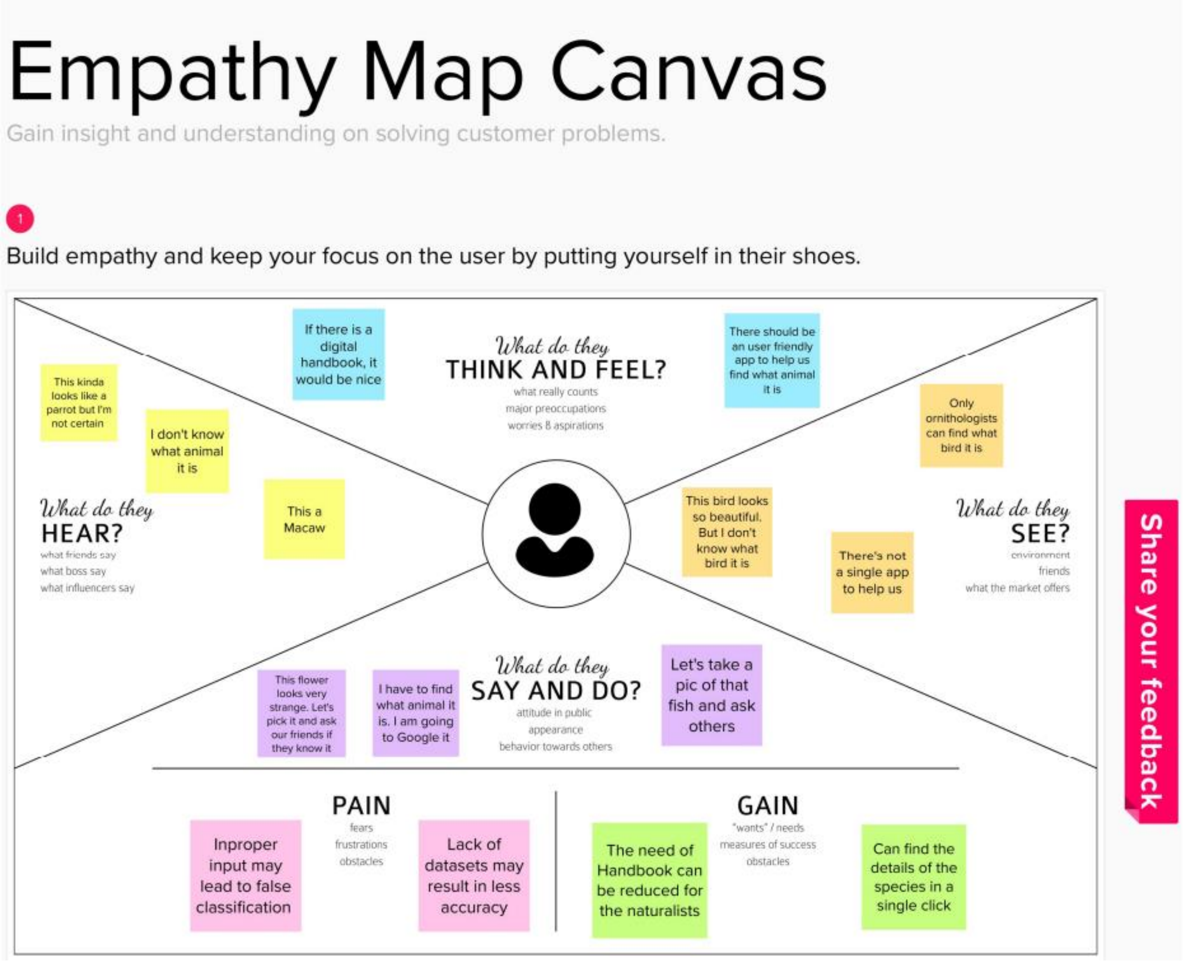
2.3 Problem Statement Definition

When venturing into the woods, field naturalists usually rely on common approaches like always carrying a guidebook around everywhere or seeking help from experienced ornithologists. There should be a handy tool for them to capture, identify and share the beauty to the outside world. In this project, we create a web application which uses a deep learning model, trained on different species of birds, flowers and mammals (3 subclasses in each for a quick understanding)and get the prediction of the bird when an image is been given.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

This allows us to map out various concerns and gain insights on the problems from the customer’s point of view



Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they THINK AND FEEL?

what really counts
major preoccupations
worries & aspirations

If there is a digital handbook, it would be nice

There should be an user friendly app to help us find what animal it is

What do they HEAR?

what friends say
what boss say
what influencers say

This kinda looks like a parrot but I'm not certain

I don't know what animal it is

This a Macaw

What do they SEE?

environment
friends
what the market offers

Only ornithologists can find what bird it is

This bird looks so beautiful. But I don't know what bird it is

There's not a single app to help us

What do they SAY AND DO?

attitude in public
appearance
behavior towards others

This flower looks very strange. Let's pick it and ask our friends if they know it

I have to find what animal it is. I am going to Google it

Let's take a pic of that fish and ask others

PAIN

fears
frustrations
obstacles

Inproper input may lead to false classification

Lack of datasets may result in less accuracy

GAIN

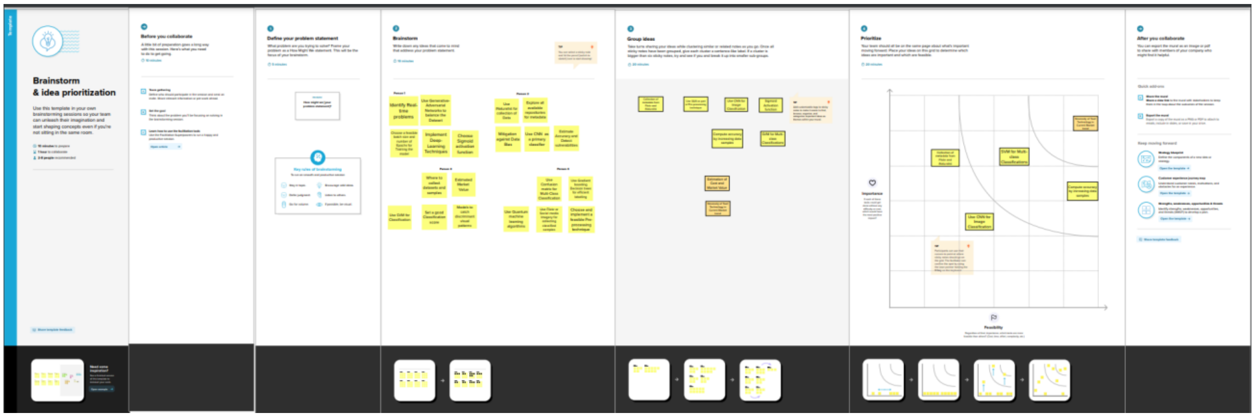
“wants” / needs
measures of success
obstacles

The need of Handbook can be reduced for the naturalists

Can find the details of the species in a single click

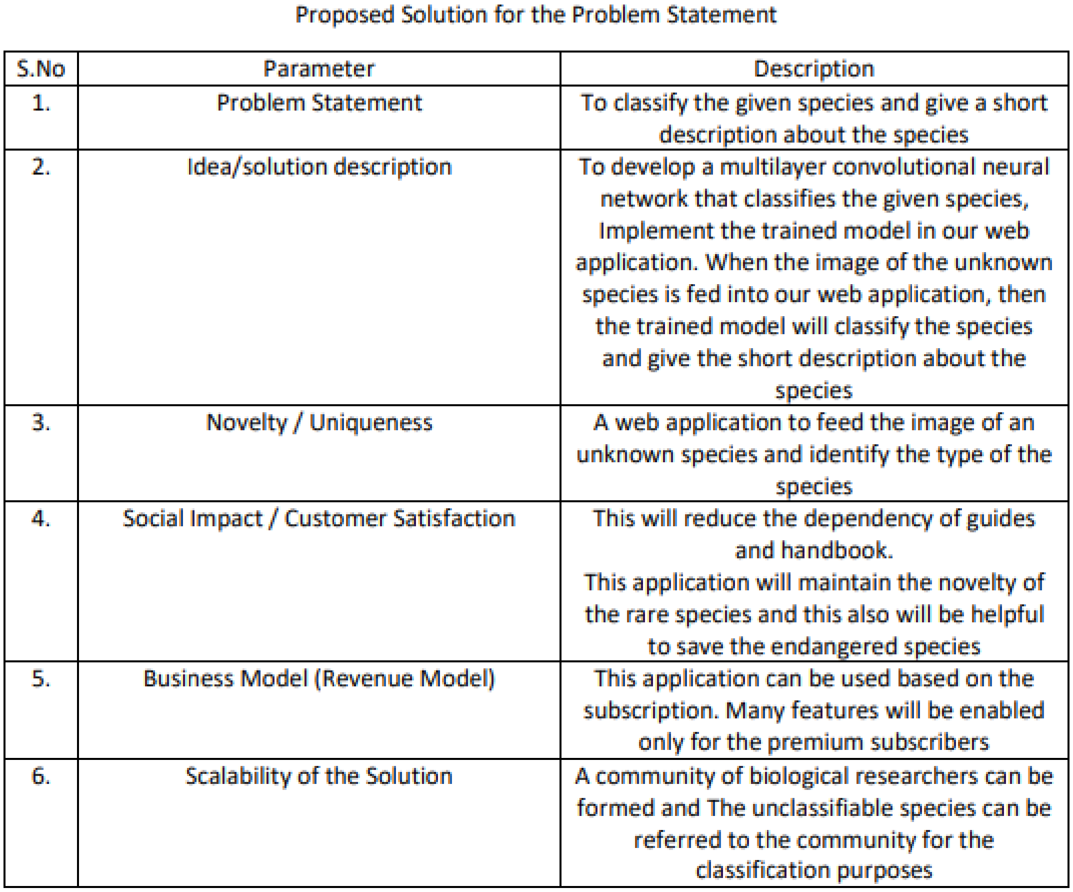
3.2 Ideation & Brainstorming

It is a collaborative thinking process that is open to suggestions by any teammate and the actual implementable ideas are filtered


Brainstorm
& idea prioritization
Before you collaborate
Define your problem statement
Brainstorm
Group ideas
Prioritize
After you collaborate
Identify Real-time problems
Use Generative Adversarial Networks to balance the Dataset
Implement Deep Learning Techniques
Choose Sigmoid activation function
Mitigation against Data Bias
Use CNN as a primary classifier
Where to collect datasets and samples
Estimated Market Value
Use CNN for Classification
Set a good Classification score
Use Quantum machine learning algorithms
Choose and implement a feasible Pre-processing technique
Importance
Feasibility

3.3 Proposed solution

It wraps up the whole overview of the project including the technical and practical impacts in a descriptive manner and flow

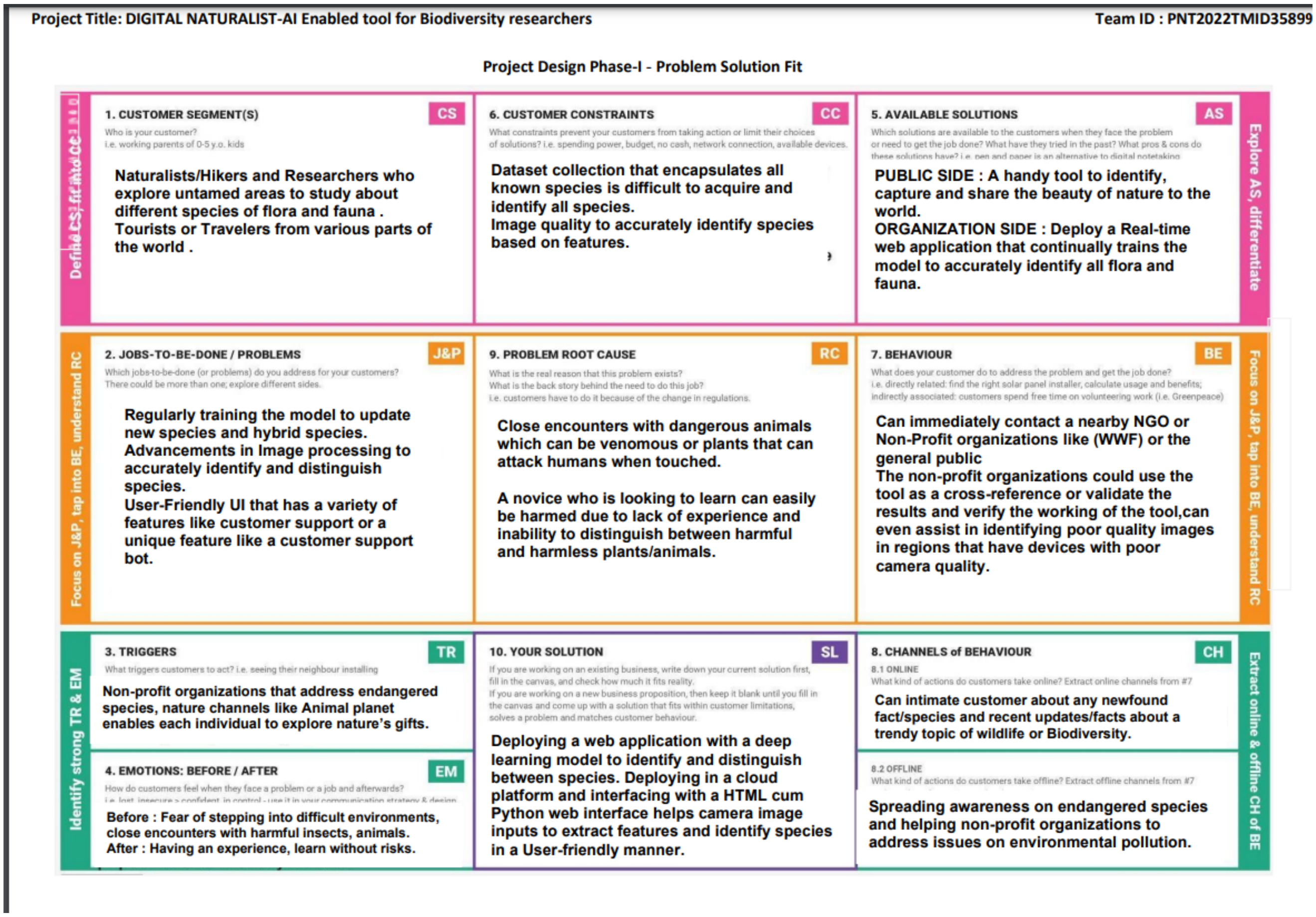


Proposed Solution for the Problem Statement

S.No	Parameter	Description
1.	Problem Statement	To classify the given species and give a short description about the species
2.	Idea/solution description	To develop a multilayer convolutional neural network that classifies the given species, Implement the trained model in our web application. When the image of the unknown species is fed into our web application, then the trained model will classify the species and give the short description about the species
3.	Novelty / Uniqueness	A web application to feed the image of an unknown species and identify the type of the species
4.	Social Impact / Customer Satisfaction	This will reduce the dependency of guides and handbook. This application will maintain the novelty of the rare species and this also will be helpful to save the endangered species
5.	Business Model (Revenue Model)	This application can be used based on the subscription. Many features will be enabled only for the premium subscribers
6.	Scalability of the Solution	A community of biological researchers can be formed and The unclassifiable species can be referred to the community for the classification purposes

3.4 Problem Solution Fit

It embodies all the elements that need to be thoroughly checked including the Who,What,When,Where,Why.



Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE

Project Title: DIGITAL NATURALIST-AI Enabled tool for Biodiversity researchers

Team ID : PNT2022TMID35899

Project Design Phase-I - Problem Solution Fit

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Naturalists/Hikers and Researchers who explore untamed areas to study about different species of flora and fauna .
Tourists or Travelers from various parts of the world .

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Dataset collection that encapsulates all known species is difficult to acquire and identify all species.
Image quality to accurately identify species based on features.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

PUBLIC SIDE : A handy tool to identify, capture and share the beauty of nature to the world.
ORGANIZATION SIDE : Deploy a Real-time web application that continually trains the model to accurately identify all flora and fauna.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Regularly training the model to update new species and hybrid species.
Advancements in Image processing to accurately identify and distinguish species.
User-Friendly UI that has a variety of features like customer support or a unique feature like a customer support bot.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Close encounters with dangerous animals which can be venomous or plants that can attack humans when touched.

A novice who is looking to learn can easily be harmed due to lack of experience and inability to distinguish between harmful and harmless plants/animals.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Can immediately contact a nearby NGO or Non-Profit organizations like (WWF) or the general public
The non-profit organizations could use the tool as a cross-reference or validate the results and verify the working of the tool,can even assist in identifying poor quality images in regions that have devices with poor camera quality.

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing

Non-profit organizations that address endangered species, nature channels like Animal planet enables each individual to explore nature's gifts.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design

Before : Fear of stepping into difficult environments, close encounters with harmful insects, animals.
After : Having an experience, learn without risks.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Deploying a web application with a deep learning model to identify and distinguish between species. Deploying in a cloud platform and interfacing with a HTML cum Python web interface helps camera image inputs to extract features and identify species in a User-friendly manner.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

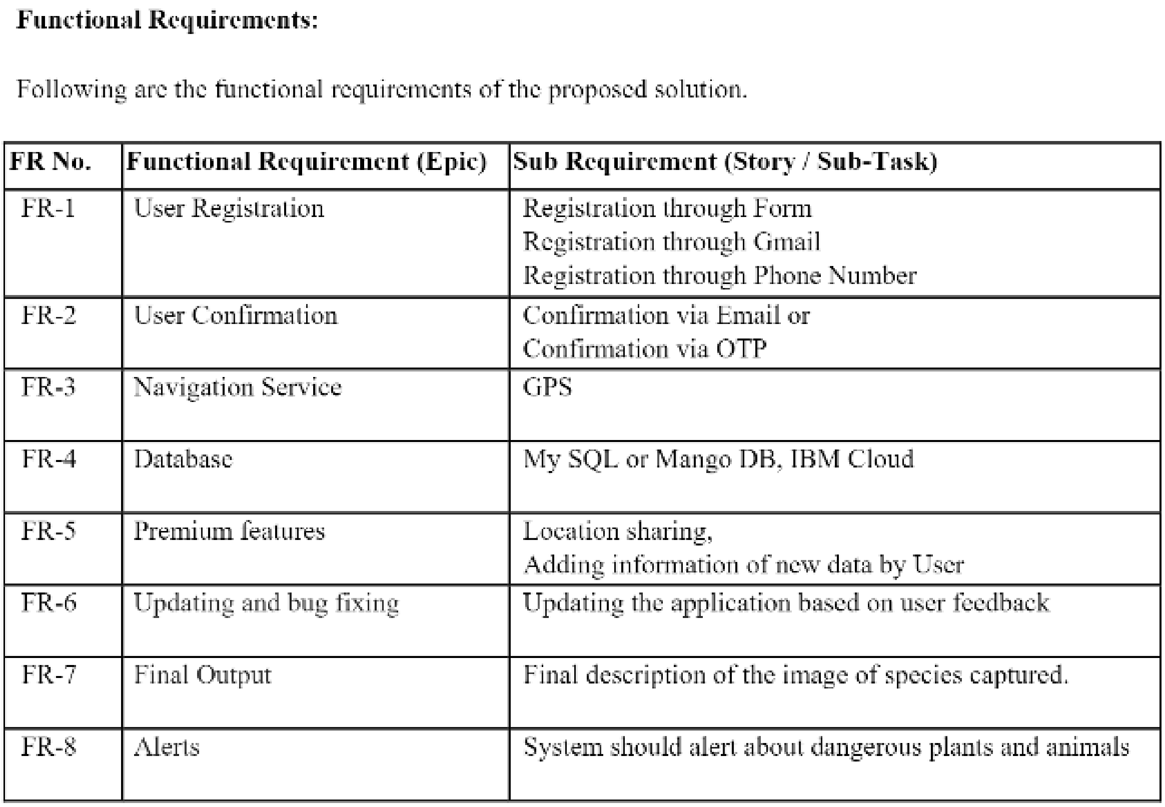
Can intimate customer about any newfound fact/species and recent updates/facts about a trendy topic of wildlife or Biodiversity.

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7

Spreading awareness on endangered species and helping non-profit organizations to address issues on environmental pollution.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

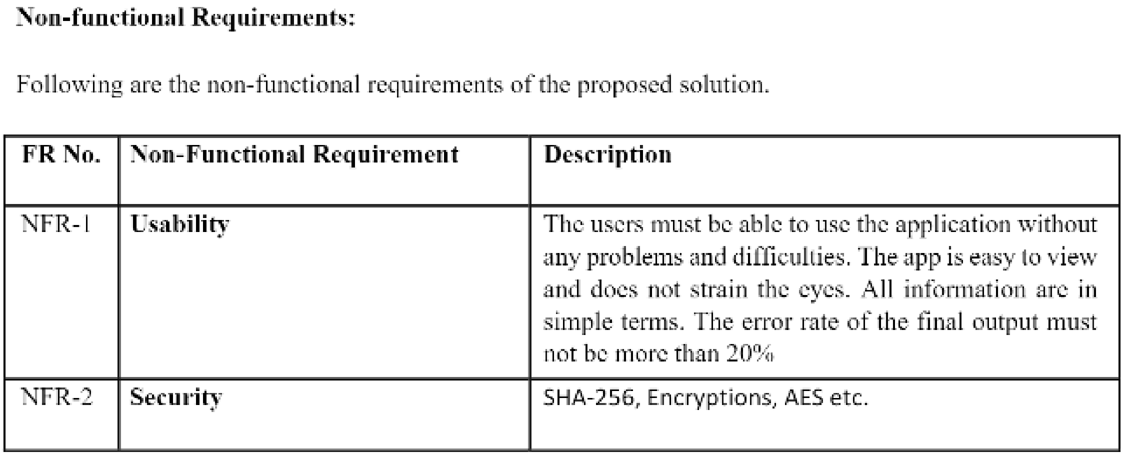


Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Phone Number
FR-2	User Confirmation	Confirmation via Email or Confirmation via OTP
FR-3	Navigation Service	GPS
FR-4	Database	My SQL or Mango DB, IBM Cloud
FR-5	Premium features	Location sharing, Adding information of new data by User
FR-6	Updating and bug fixing	Updating the application based on user feedback
FR-7	Final Output	Final description of the image of species captured.
FR-8	Alerts	System should alert about dangerous plants and animals

4.2 Non-Functional requirements



Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The users must be able to use the application without any problems and difficulties. The app is easy to view and does not strain the eyes. All information are in simple terms. The error rate of the final output must not be more than 20%
NFR-2	**Security**	SHA-256, Encryptions, AES etc.

NFR-3	**Reliability**	The system must perform without failure in 80 percent of the time.
NFR-4	**Performance**	Under normal load, the system must show the results within 15 seconds, and under maximum it can take up to load 30 seconds
NFR-5	**Availability**	The application will be available 99 % of the time in a month.
NFR-6	**Scalability**	The system must be able to support 10,000 users while using it. As the usage and user base of this application grows, more features can be added like languages based on the geographical usage, premium or subscription model, etc.

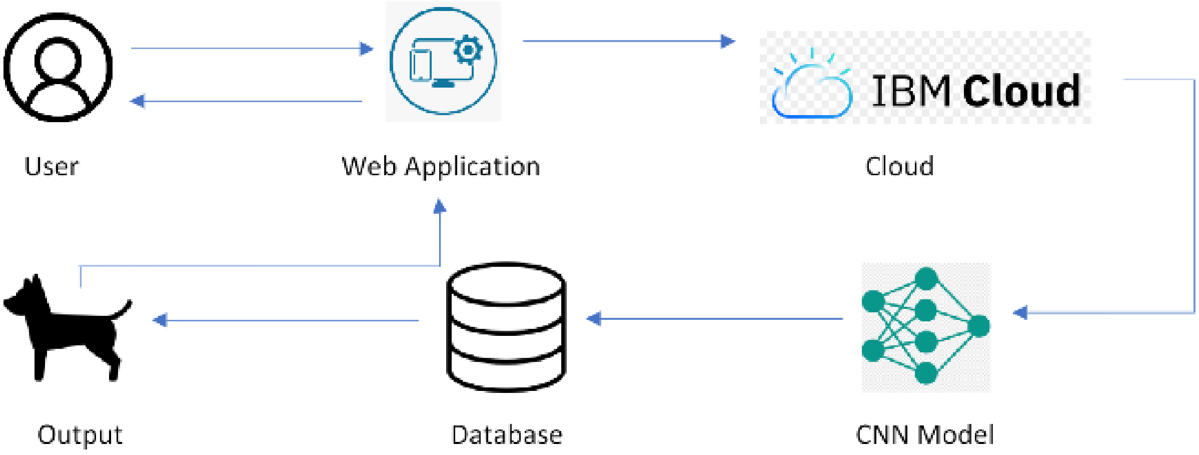
5. PROJECT DESIGN

5.1 Data Flow Diagrams

Data Flow Diagrams:

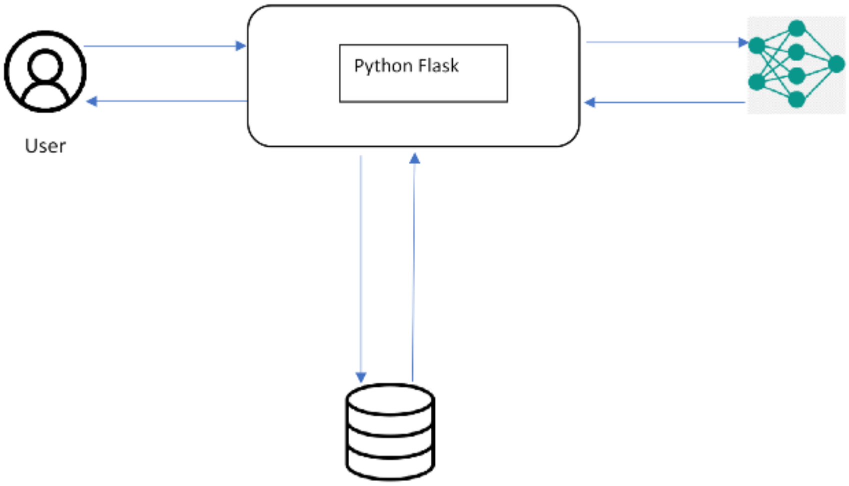
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Simplified DFD:

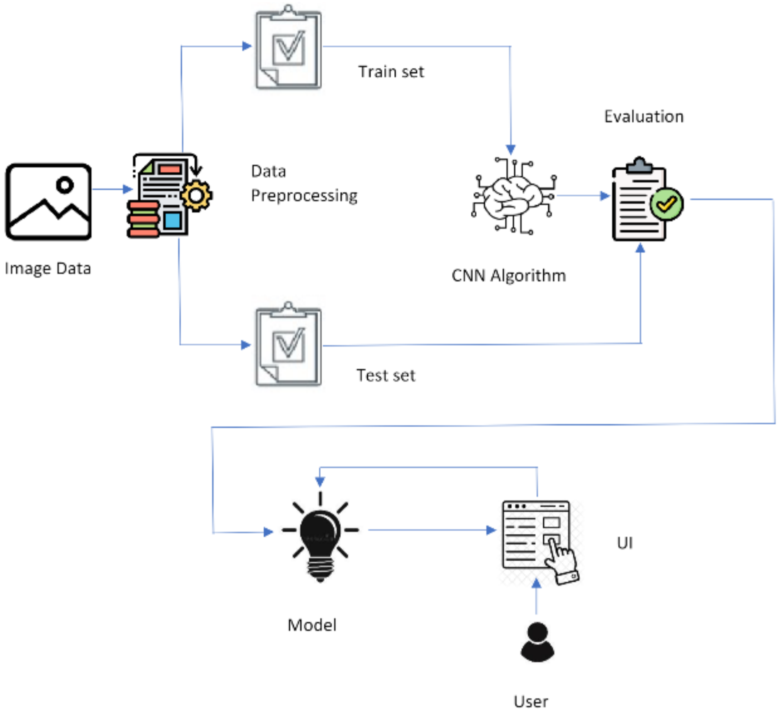


1. The user uploads the image of an animal/bird/plants etc.
2. The Web Application sends it to the IBM Cloud.
3. The CNN models accept the input and classifies the type of the species
4. The information is matched with the database and its respective description is given as the output.

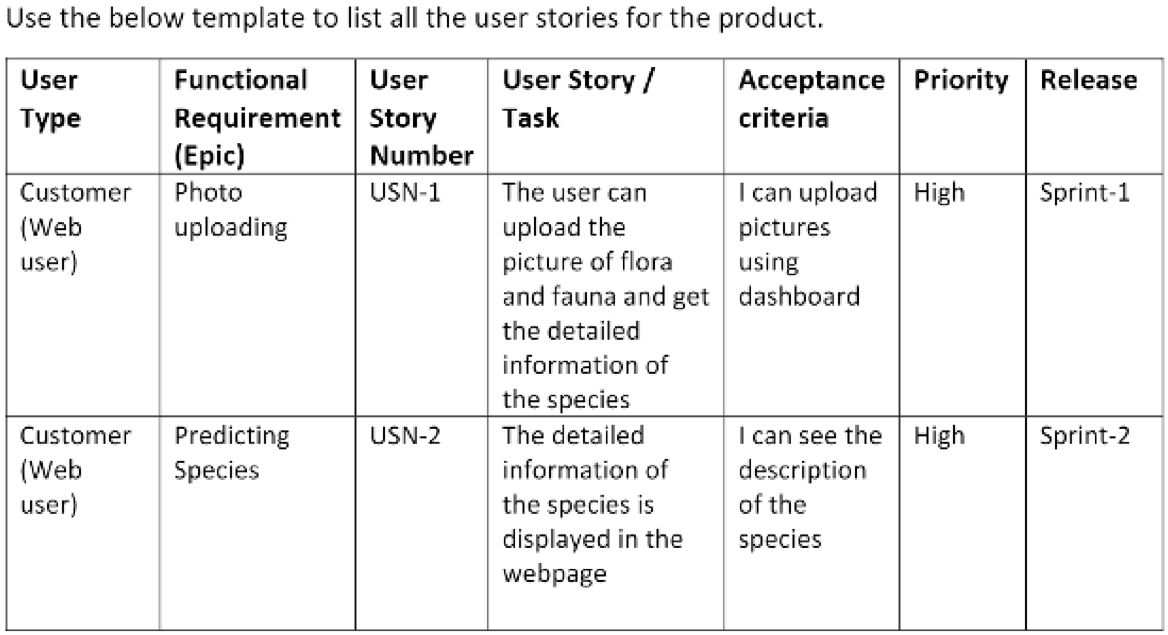
DFD Standard:



5.2 Solution & Technical Architecture



5.3 User Stories



Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Photo uploading	USN-1	The user can upload the picture of flora and fauna and get the detailed information of the species	I can upload pictures using dashboard	High	Sprint-1
Customer (Web user)	Predicting Species	USN-2	The detailed information of the species is displayed in the webpage	I can see the description of the species	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

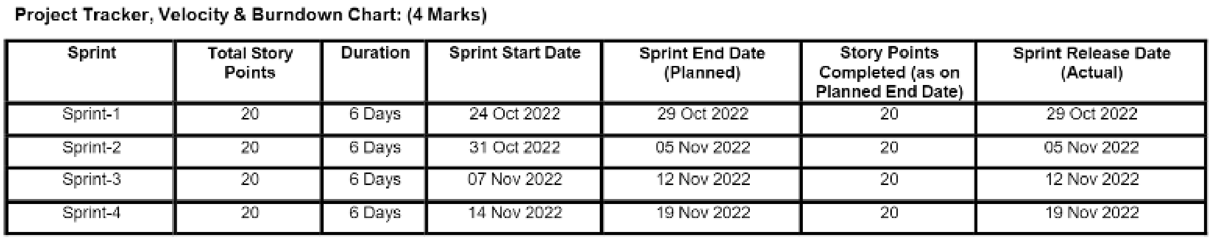
Sprint is the subdivision of our project work. Here we have divided our project into four sprints. Each sprint consists of the activities to be done as follows.

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Getting Hands on with IBM Watson Assistant	USN-1	Setting Up IBM Cloud and Watson Assistant	5	High	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-1	Dataset Gathering	USN-2	Acquire the Datasets for the Species Classification and do the required preprocessing. Upload those datasets to IBM Cloud	15	High	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-2	Building the CNN Model and Splitting the dataset	USN-3	Build a CNN model using the appropriate layers for the Species Classification. Split the preprocessed dataset to train, test and validation data	4	High	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-2	Train, Test, and Validate	USN-4	Train the model using the Training datasets and Validate it with the validation dataset. Test the Model using the testing dataset and analyze the Performance Metrics	8	High	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-2	Optimization and Species Classification	USN-5	Improve the Accuracy and Time Complexity of the model	8	High	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-3	User Interface Dashboard	USN-6	As an User I should be able to capture and upload the image in this Web App	10	Medium	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-3	Description of the collected datasets	USN-7	Store the Description of the collected dataset classes in the Backend for output purposes	10	High	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-3	Output Page of the App	USN-8	As an User I should be able to interpret the information of the unknown species in a crisp manner using the Web App	10	High	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-4	Integrating our model in cloud	USN-9	Bridging the CNN model for classification with the IBM Cloud to display the identification and description of the species.	15	High	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G
Sprint-4	Web Page Optimization andcustomer support	USN-10	As an User I need a smooth user experience. The site should withstand a heavy load and traffic. There should be no failures and all my queries should be handled	5	Medium	PRIYADHARSHAN S VAROON K NAVEEN KUMAR S NARESH GUPTHA G

6.2 Sprint Delivery Schedule

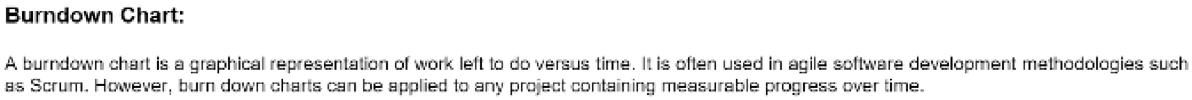
The duration for each sprint is given as follows



Project Tracker, Velocity & Burndown Chart: (4 Marks)

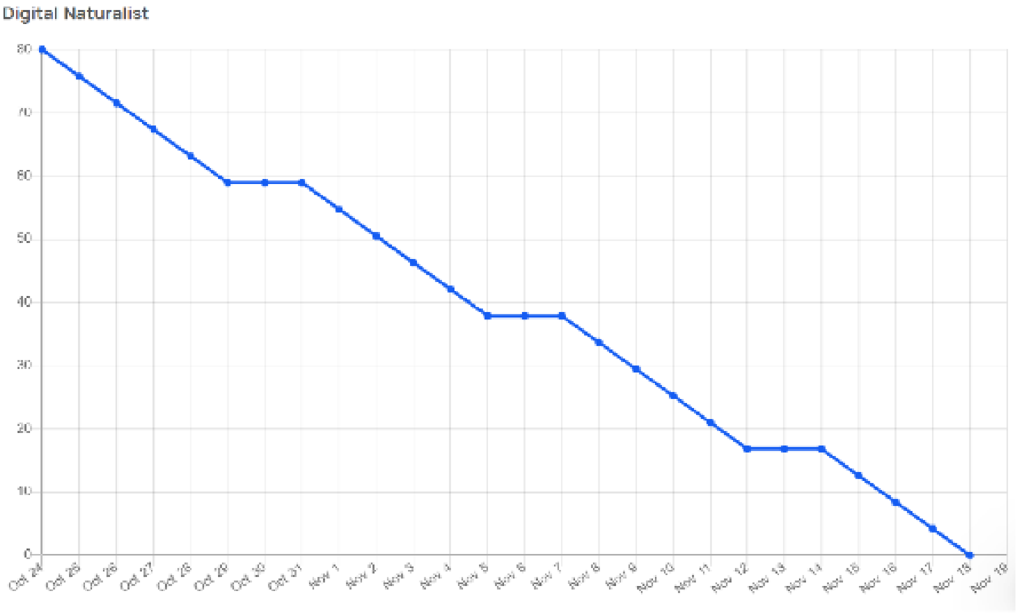
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA



Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



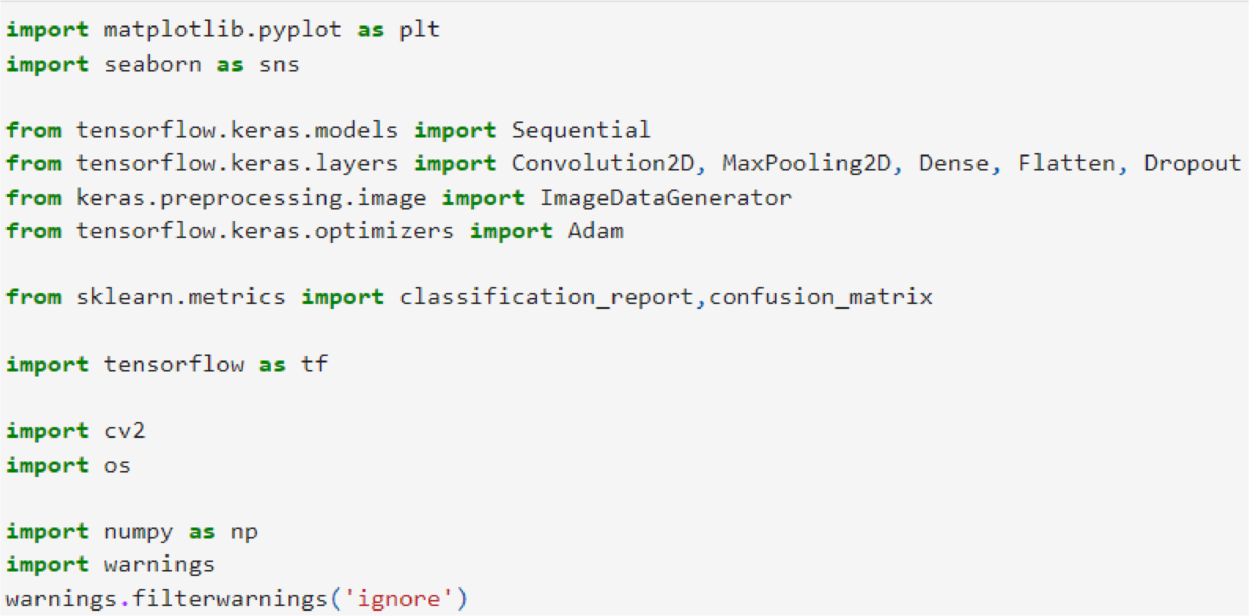
Velocity:
Imagine we have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{6} = 3.33$$

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

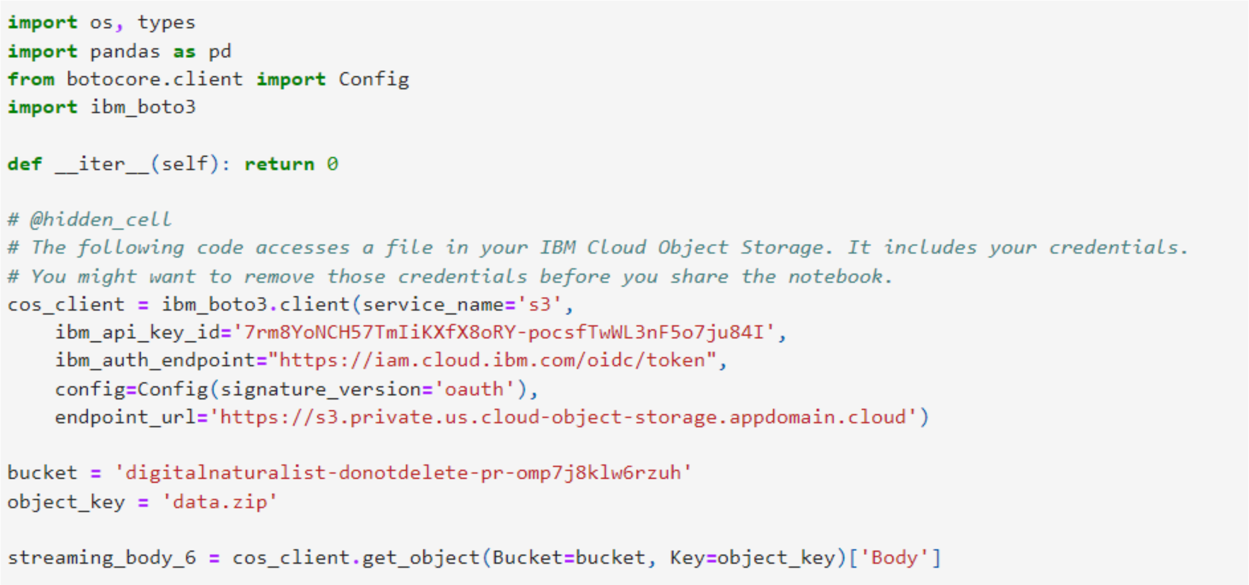
The following code was implemented in IBM Cloud Pak

Importing the dataset



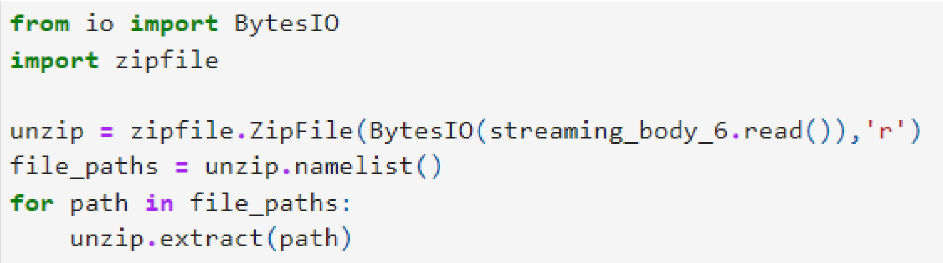
```python
import matplotlib.pyplot as plt
import seaborn as sns

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Dense, Flatten, Dropout
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam

from sklearn.metrics import classification_report,confusion_matrix

import tensorflow as tf

import cv2
import os

import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

Creating the Bucket in the IBM cloud storage



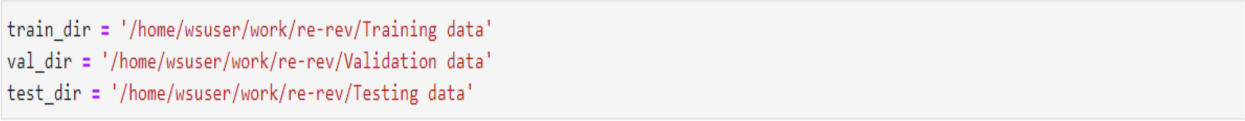
```python
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='7rm8YoNCH57TmIiKXfX8oRY-pocsfTwWL3nF5o7ju84I',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'digitalnaturalist-donotdelete-pr-omp7j8klw6rzuh'
object_key = 'data.zip'

streaming_body_6 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']
```

Unzipping the zipped dataset

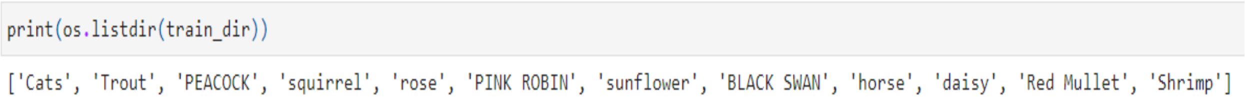


```
from io import BytesIO
import zipfile

unzip = zipfile.ZipFile(BytesIO(streaming_body_6.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

Dividing the dataset into training, testing and validation datasets



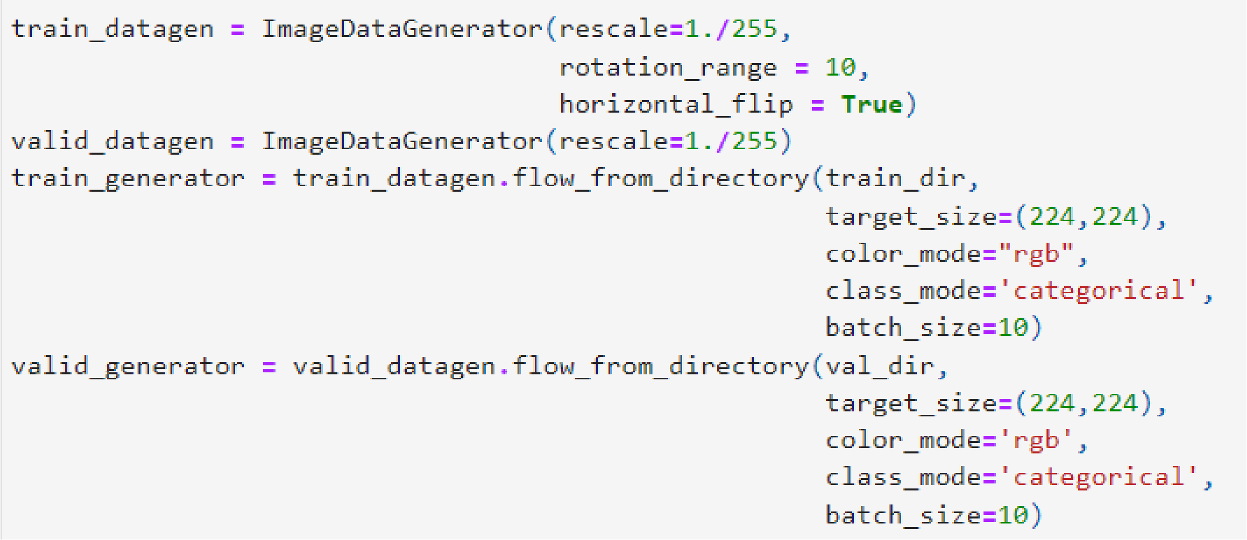
```
train_dir = '/home/wsuser/work/re-rev/Training data'
val_dir = '/home/wsuser/work/re-rev/Validation data'
test_dir = '/home/wsuser/work/re-rev/Testing data'
```



```
print(os.listdir(train_dir))
```

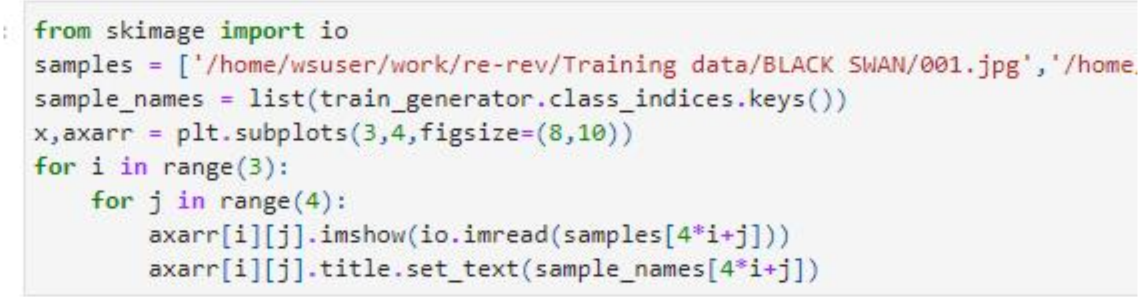
```
['Cats', 'Trout', 'PEACOCK', 'squirrel', 'rose', 'PINK ROBIN', 'sunflower', 'BLACK SWAN', 'horse', 'daisy', 'Red Mullet', 'Shrimp']
```

Creating the Training and Validation data generators

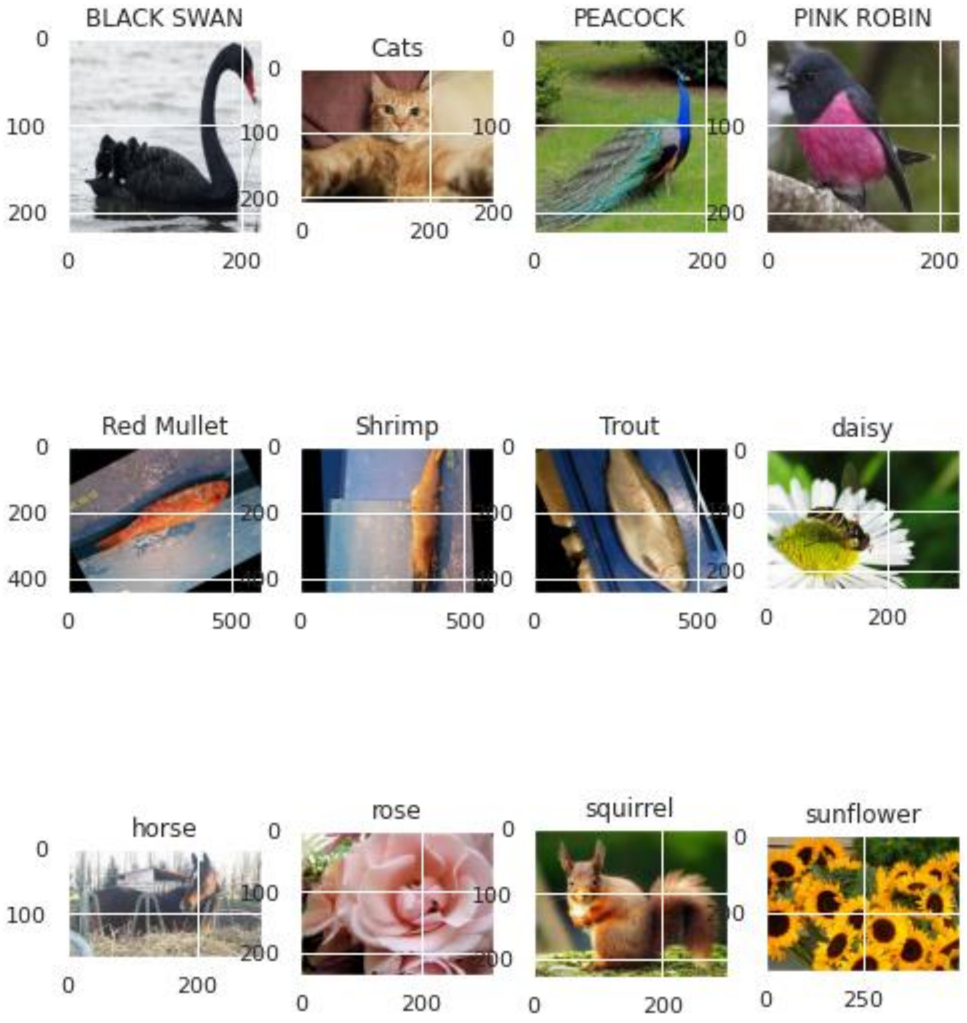


```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   rotation_range = 10,
                                   horizontal_flip = True)
valid_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(train_dir,
                                                    target_size=(224,224),
                                                    color_mode="rgb",
                                                    class_mode='categorical',
                                                    batch_size=10)
valid_generator = valid_datagen.flow_from_directory(val_dir,
                                                    target_size=(224,224),
                                                    color_mode='rgb',
                                                    class_mode='categorical',
                                                    batch_size=10)
```

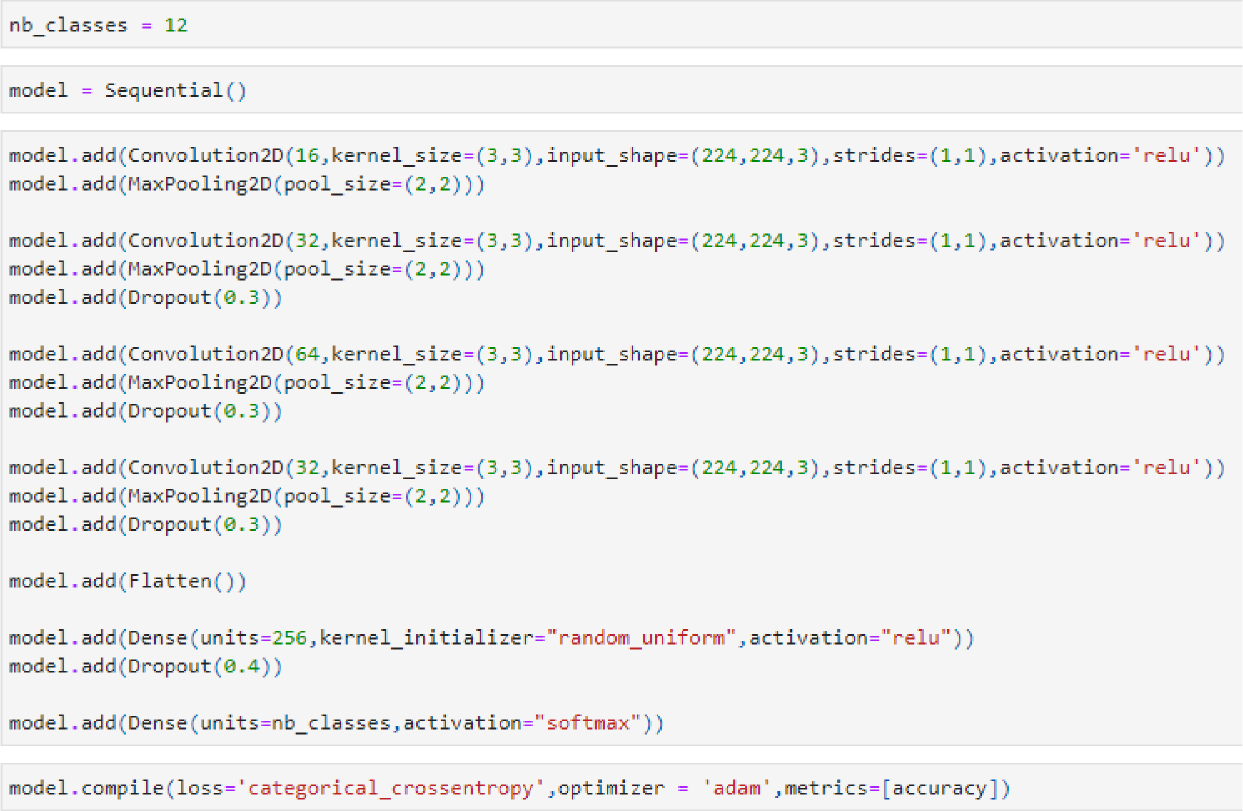
Sample images of the classes:



```
from skimage import io
samples = ['/home/wsuser/work/re-rev/Training data/BLACK SWAN/001.jpg','/home
sample_names = list(train_generator.class_indices.keys())
x,axarr = plt.subplots(3,4,figsize=(8,10))
for i in range(3):
    for j in range(4):
        axarr[i][j].imshow(io.imread(samples[4*i+j]))
        axarr[i][j].title.set_text(sample_names[4*i+j])
```



Creating a CNN model using the following layers and compiling the model using the Adam optimizer



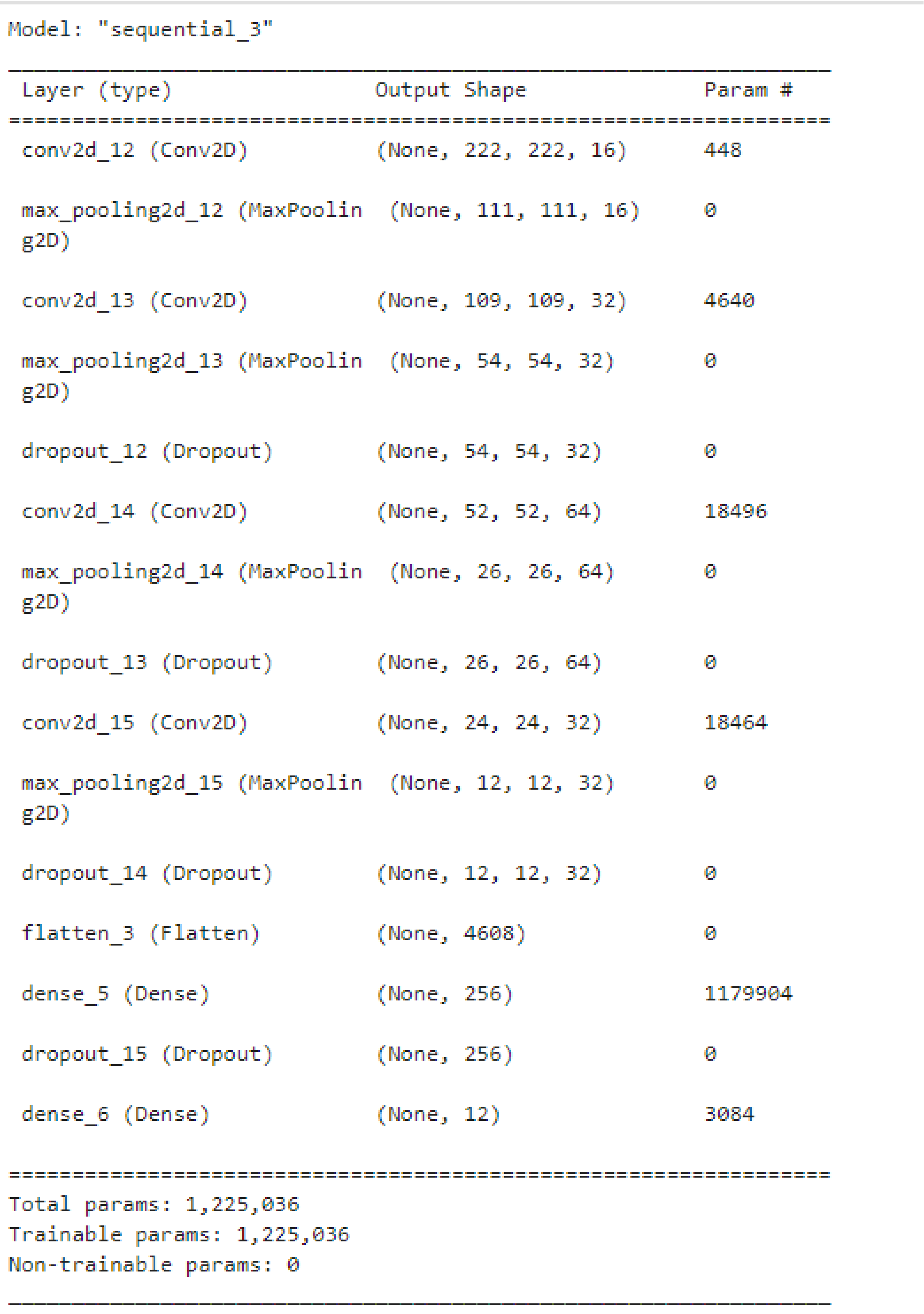
```
nb_classes = 12
```

```
model = Sequential()
```

```
model.add(Convolution2D(16,kernel_size=(3,3),input_shape=(224,224,3),strides=(1,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Convolution2D(32,kernel_size=(3,3),input_shape=(224,224,3),strides=(1,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Convolution2D(64,kernel_size=(3,3),input_shape=(224,224,3),strides=(1,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Convolution2D(32,kernel_size=(3,3),input_shape=(224,224,3),strides=(1,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Flatten())

model.add(Dense(units=256,kernel_initializer="random_uniform",activation="relu"))
model.add(Dropout(0.4))

model.add(Dense(units=nb_classes,activation="softmax"))
```

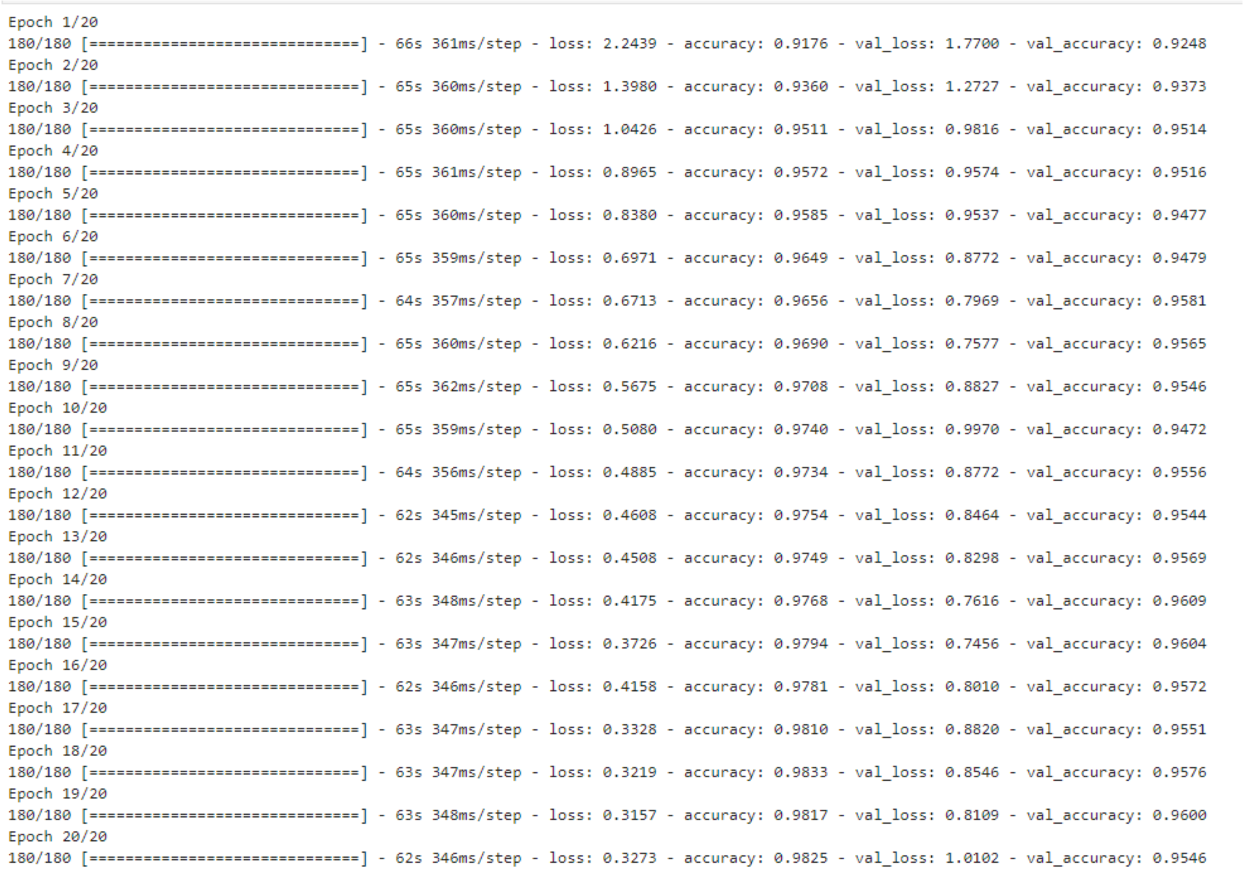
```
model.compile(loss='categorical_crossentropy',optimizer = 'adam',metrics=[accuracy])
```

Summary of the model:



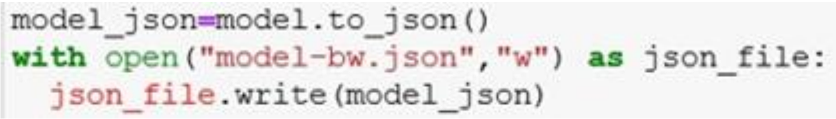
```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_12 (Conv2D)          (None, 222, 222, 16)      448

 max_pooling2d_12 (MaxPoolin  (None, 111, 111, 16)     0
 g2D)

 conv2d_13 (Conv2D)          (None, 109, 109, 32)      4640

 max_pooling2d_13 (MaxPoolin  (None, 54, 54, 32)       0
 g2D)

 dropout_12 (Dropout)        (None, 54, 54, 32)        0

 conv2d_14 (Conv2D)          (None, 52, 52, 64)        18496

 max_pooling2d_14 (MaxPoolin  (None, 26, 26, 64)       0
 g2D)

 dropout_13 (Dropout)        (None, 26, 26, 64)        0

 conv2d_15 (Conv2D)          (None, 24, 24, 32)        18464

 max_pooling2d_15 (MaxPoolin  (None, 12, 12, 32)       0
 g2D)

 dropout_14 (Dropout)        (None, 12, 12, 32)        0

 flatten_3 (Flatten)         (None, 4608)              0

 dense_5 (Dense)             (None, 256)               1179904

 dropout_15 (Dropout)        (None, 256)               0

 dense_6 (Dense)             (None, 12)                3084

=================================================================
Total params: 1,225,036
Trainable params: 1,225,036
Non-trainable params: 0
_________________________________________________________________
```

Training and Validation of the model



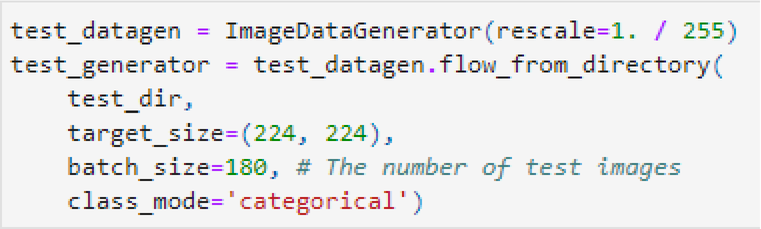
```
Epoch 1/20
180/180 [==============================] - 66s 361ms/step - loss: 2.2439 - accuracy: 0.9176 - val_loss: 1.7700 - val_accuracy: 0.9248
Epoch 2/20
180/180 [==============================] - 65s 360ms/step - loss: 1.3980 - accuracy: 0.9360 - val_loss: 1.2727 - val_accuracy: 0.9373
Epoch 3/20
180/180 [==============================] - 65s 360ms/step - loss: 1.0426 - accuracy: 0.9511 - val_loss: 0.9816 - val_accuracy: 0.9514
Epoch 4/20
180/180 [==============================] - 65s 361ms/step - loss: 0.8965 - accuracy: 0.9572 - val_loss: 0.9574 - val_accuracy: 0.9516
Epoch 5/20
180/180 [==============================] - 65s 360ms/step - loss: 0.8380 - accuracy: 0.9585 - val_loss: 0.9537 - val_accuracy: 0.9477
Epoch 6/20
180/180 [==============================] - 65s 359ms/step - loss: 0.6971 - accuracy: 0.9649 - val_loss: 0.8772 - val_accuracy: 0.9479
Epoch 7/20
180/180 [==============================] - 64s 357ms/step - loss: 0.6713 - accuracy: 0.9656 - val_loss: 0.7969 - val_accuracy: 0.9581
Epoch 8/20
180/180 [==============================] - 65s 360ms/step - loss: 0.6216 - accuracy: 0.9690 - val_loss: 0.7577 - val_accuracy: 0.9565
Epoch 9/20
180/180 [==============================] - 65s 362ms/step - loss: 0.5675 - accuracy: 0.9708 - val_loss: 0.8827 - val_accuracy: 0.9546
Epoch 10/20
180/180 [==============================] - 65s 359ms/step - loss: 0.5080 - accuracy: 0.9740 - val_loss: 0.9970 - val_accuracy: 0.9472
Epoch 11/20
180/180 [==============================] - 64s 356ms/step - loss: 0.4885 - accuracy: 0.9734 - val_loss: 0.8772 - val_accuracy: 0.9556
Epoch 12/20
180/180 [==============================] - 62s 345ms/step - loss: 0.4608 - accuracy: 0.9754 - val_loss: 0.8464 - val_accuracy: 0.9544
Epoch 13/20
180/180 [==============================] - 62s 346ms/step - loss: 0.4508 - accuracy: 0.9749 - val_loss: 0.8298 - val_accuracy: 0.9569
Epoch 14/20
180/180 [==============================] - 63s 348ms/step - loss: 0.4175 - accuracy: 0.9768 - val_loss: 0.7616 - val_accuracy: 0.9609
Epoch 15/20
180/180 [==============================] - 63s 347ms/step - loss: 0.3726 - accuracy: 0.9794 - val_loss: 0.7456 - val_accuracy: 0.9604
Epoch 16/20
180/180 [==============================] - 62s 346ms/step - loss: 0.4158 - accuracy: 0.9781 - val_loss: 0.8010 - val_accuracy: 0.9572
Epoch 17/20
180/180 [==============================] - 63s 347ms/step - loss: 0.3328 - accuracy: 0.9810 - val_loss: 0.8820 - val_accuracy: 0.9551
Epoch 18/20
180/180 [==============================] - 63s 347ms/step - loss: 0.3219 - accuracy: 0.9833 - val_loss: 0.8546 - val_accuracy: 0.9576
Epoch 19/20
180/180 [==============================] - 63s 348ms/step - loss: 0.3157 - accuracy: 0.9817 - val_loss: 0.8109 - val_accuracy: 0.9600
Epoch 20/20
180/180 [==============================] - 62s 346ms/step - loss: 0.3273 - accuracy: 0.9825 - val_loss: 1.0102 - val_accuracy: 0.9546
```

Saving the model in json format



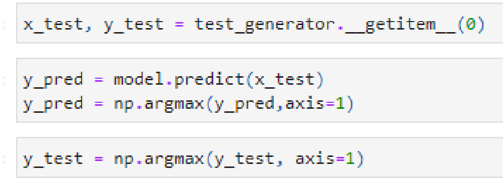
```
model_json=model.to_json()
with open("model-bw.json","w") as json_file:
  json_file.write(model_json)
```

Creating the test datagen:



```
test_datagen = ImageDataGenerator(rescale=1. / 255)
test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=180, # The number of test images
    class_mode='categorical')
```

Predicting the test set:



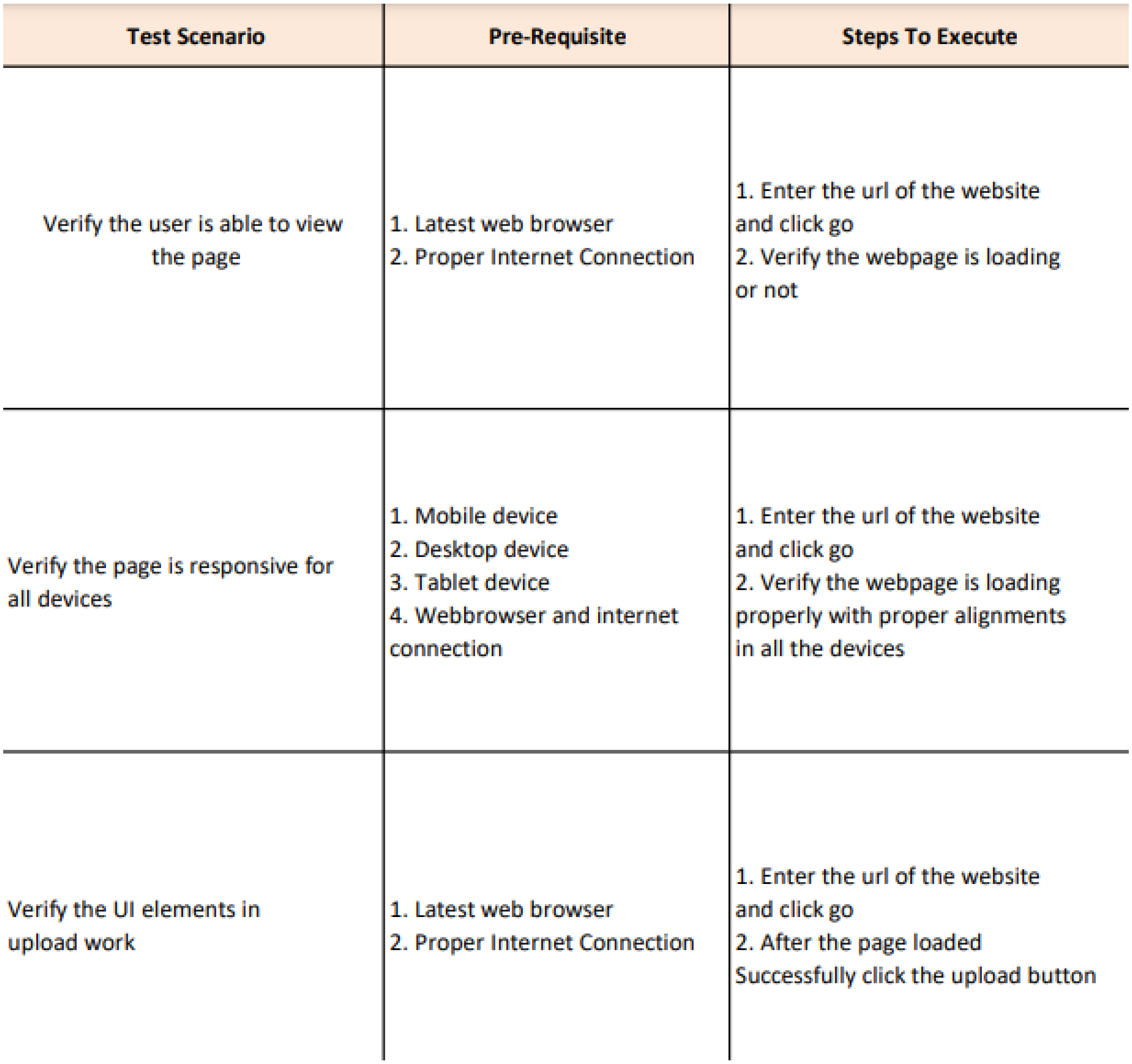
```
x_test, y_test = test_generator.__getitem__(0)
```

```
y_pred = model.predict(x_test)
y_pred = np.argmax(y_pred,axis=1)
```

```
y_test = np.argmax(y_test, axis=1)
```

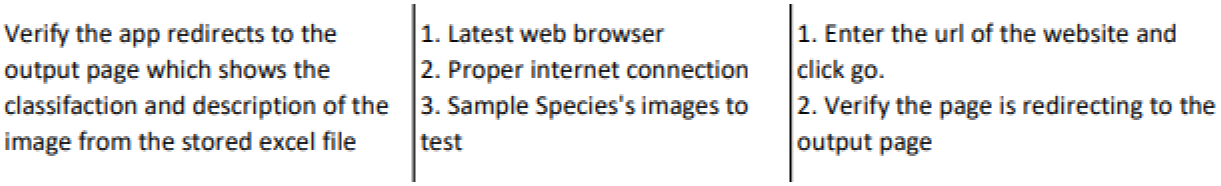
8. TESTING

8.1 Test Cases



Test Scenario	Pre-Requisite	Steps To Execute
Verify the user is able to view the page	1. Latest web browser 2. Proper Internet Connection	1. Enter the url of the website and click go 2. Verify the webpage is loading or not
Verify the page is responsive for all devices	1. Mobile device 2. Desktop device 3. Tablet device 4. Webbrowser and internet connection	1. Enter the url of the website and click go 2. Verify the webpage is loading properly with proper alignments in all the devices
Verify the UI elements in upload work	1. Latest web browser 2. Proper Internet Connection	1. Enter the url of the website and click go 2. After the page loaded Successfully click the upload button

Verify the page is responding for every user action	1. Latest web browser 2. Proper Internet Connection	1. Enter the url of the website and click go 2. Verify the webpage is loading and working properly during the upload and reset
Verify the app accepts only image formats	1. Latest web browser 2. Proper Internet Connection	1. Enter the url of the website and click go 2. After page loading try to upload non image formats such as pdf, xml, or any audio or video file
Verify the flask app use the saved model	1. Latest web browser 2. Proper Internet Connection	1. Enter the url of the website and click go 2. Verify the webpage is accepting inputs and predecting according to the category of the animal
Verify the uploaded image saved on the server	1. Latest web browser 2. Proper Internet Connection 3. Storage in the server for storing the uploaded image	1. Enter the url of the website and click go 2. After page loading try to upload the image and wait
Verify the uploaded image can be retrived from the storage	1. Latest web browser 2. Proper Internet Connection 3. Storage in the server where the uploaded image can be retrived	1. Enter the url of the website and click go 2. Verify the webpage is accepting inputs and predecting according to the category of the animal

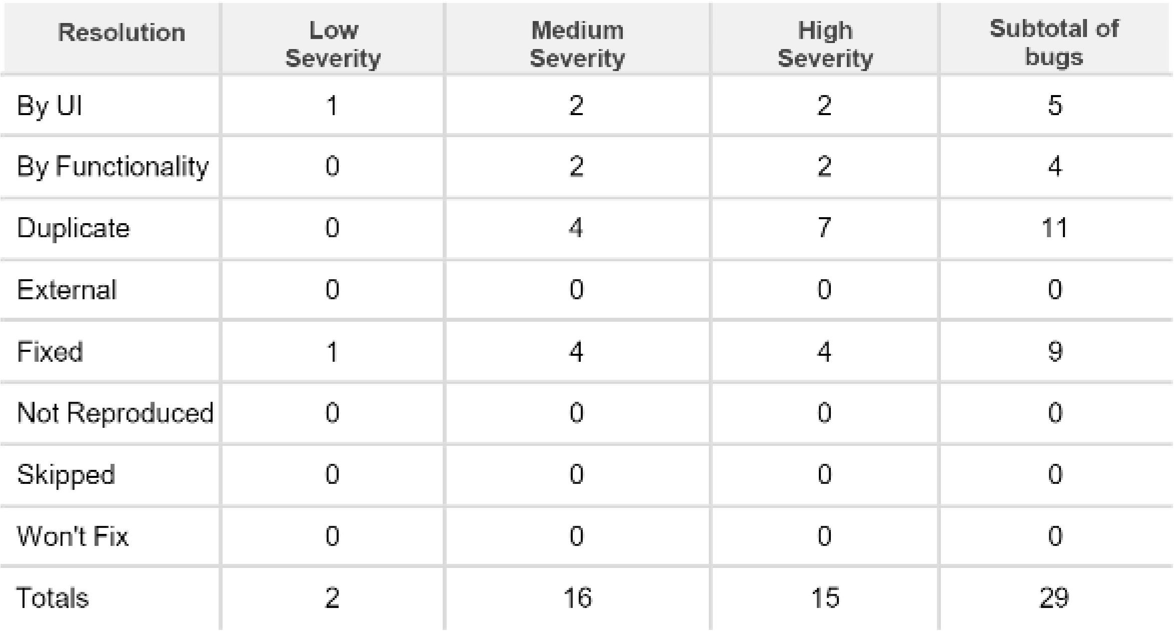


Verify the app redirects to the output page which shows the classifaction and description of the image from the stored excel file	1. Latest web browser 2. Proper internet connection 3. Sample Species's images to test	1. Enter the url of the website and click go. 2. Verify the page is redirecting to the output page

8.2 User Acceptance Testing

Defect Analysis

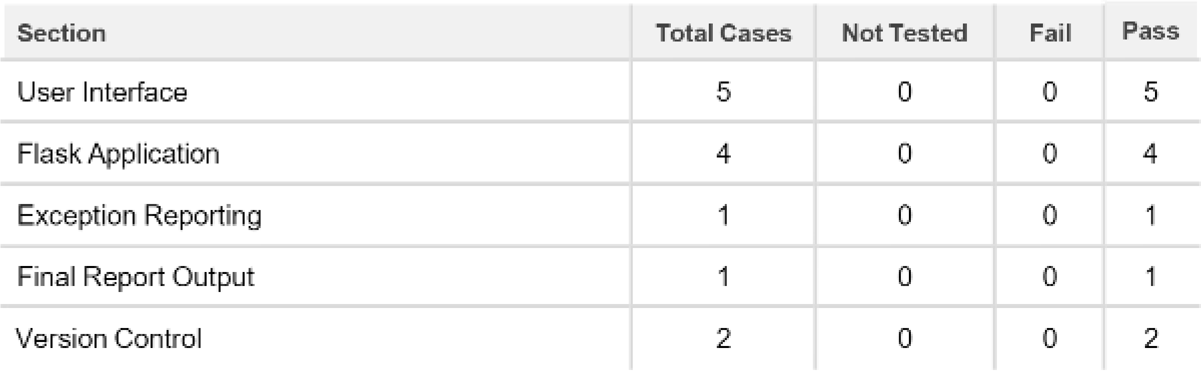
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved



Resolution	Low Severity	Medium Severity	High Severity	Subtotal of bugs
By UI	1	2	2	5
By Functionality	0	2	2	4
Duplicate	0	4	7	11
External	0	0	0	0
Fixed	1	4	4	9
Not Reproduced	0	0	0	0
Skipped	0	0	0	0
Won't Fix	0	0	0	0
Totals	2	16	15	29

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

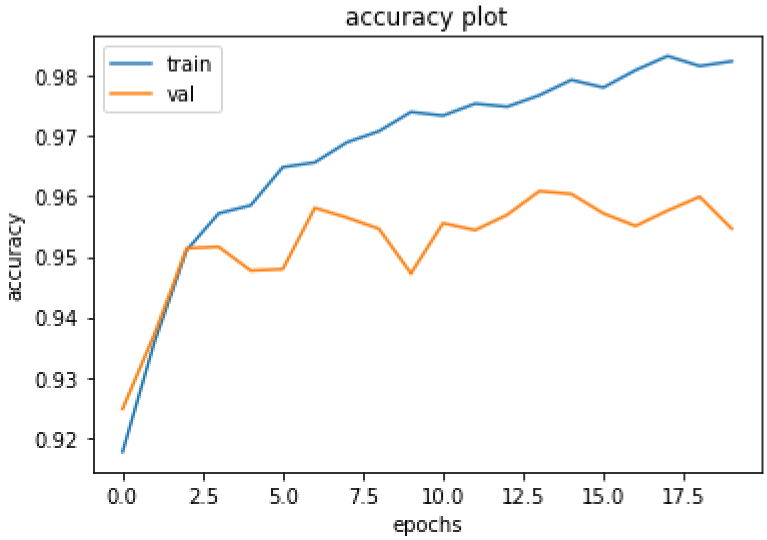


Section	Total Cases	Not Tested	Fail	Pass
User Interface	5	0	0	5
Flask Application	4	0	0	4
Exception Reporting	1	0	0	1
Final Report Output	1	0	0	1
Version Control	2	0	0	2

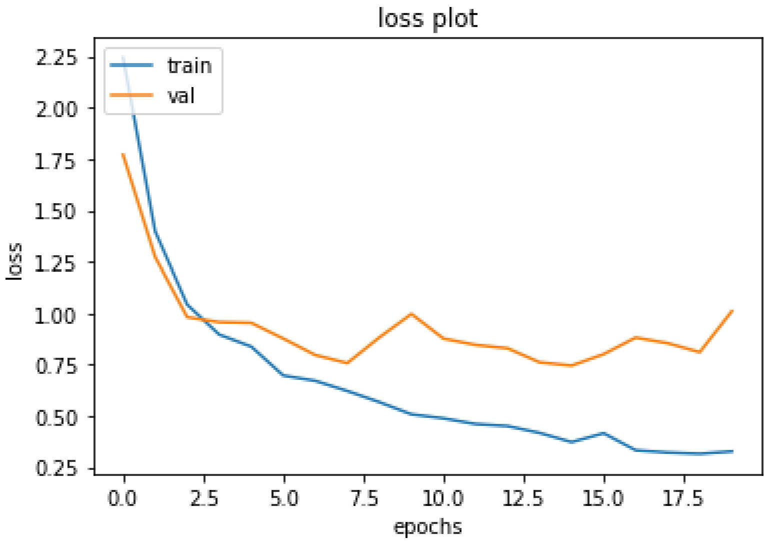
9. RESULTS

9.1 Performance Metrics

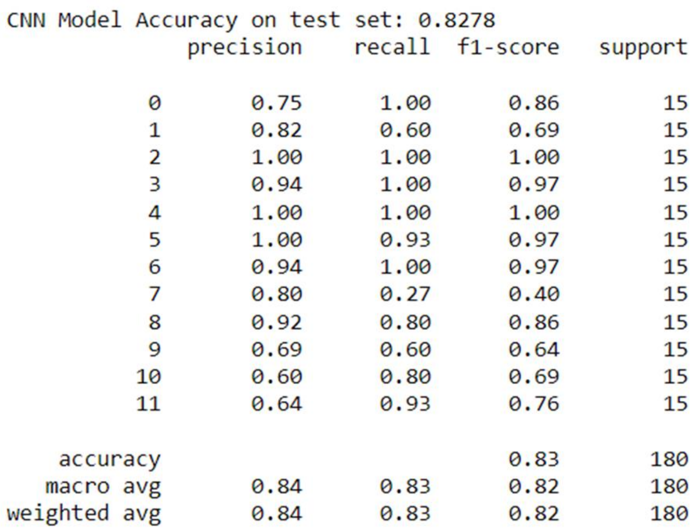
Accuracy plot of training and validation



Loss plot of training and validation:

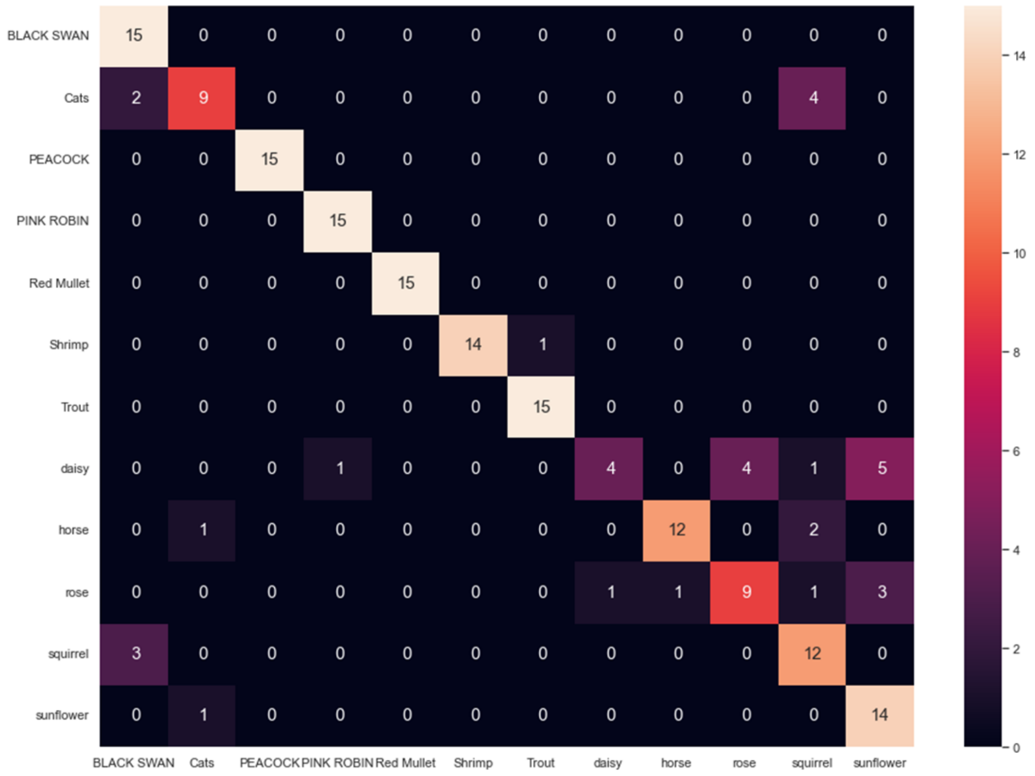


Classification Report With F1 Score



```
CNN Model Accuracy on test set: 0.8278
              precision    recall  f1-score   support

           0       0.75      1.00      0.86        15
           1       0.82      0.60      0.69        15
           2       1.00      1.00      1.00        15
           3       0.94      1.00      0.97        15
           4       1.00      1.00      1.00        15
           5       1.00      0.93      0.97        15
           6       0.94      1.00      0.97        15
           7       0.80      0.27      0.40        15
           8       0.92      0.80      0.86        15
           9       0.69      0.60      0.64        15
          10       0.60      0.80      0.69        15
          11       0.64      0.93      0.76        15

    accuracy                           0.83       180
   macro avg       0.84      0.83      0.82       180
weighted avg       0.84      0.83      0.82       180
```

Confusion Matrix



	BLACK SWAN	Cats	PEACOCK	PINK ROBIN	Red Mullet	Shrimp	Trout	daisy	horse	rose	squirrel	sunflower
BLACK SWAN	15	0	0	0	0	0	0	0	0	0	0	0
Cats	2	9	0	0	0	0	0	0	0	0	4	0
PEACOCK	0	0	15	0	0	0	0	0	0	0	0	0
PINK ROBIN	0	0	0	15	0	0	0	0	0	0	0	0
Red Mullet	0	0	0	0	15	0	0	0	0	0	0	0
Shrimp	0	0	0	0	0	14	1	0	0	0	0	0
Trout	0	0	0	0	0	0	15	0	0	0	0	0
daisy	0	0	0	1	0	0	0	4	0	4	1	5
horse	0	1	0	0	0	0	0	0	12	0	2	0
rose	0	0	0	0	0	0	0	1	1	9	1	3
squirrel	3	0	0	0	0	0	0	0	0	0	12	0
sunflower	0	1	0	0	0	0	0	0	0	0	0	14

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. It reduces the dependency of the handbook. So the travelers or the biodiversity explorers may feel the application be very useful.
2. Can be used to learn more facts about the species
3. Used to portray the salient features of the species
4. Reduces misinterpretation
5. Useful for many study purposes

Disadvantages:

1. Lack of 100% accuracy will lead to mis identification of the species
2. The information of all the species is not guaranteed

11. CONCLUSION

The web application for identifying the unknown species is developed successfully using Deep Learning and the Flask Web Framework

12. FUTURE SCOPE

With better enhancement and image processing techniques we will be able to classify any species from even at a greater distance.

Interfacing with Real-world cameras can help in identifying and alerting the public about the various kinds of animals that live around and learn about them in a matter of seconds

It is an aid to Naturalists who step foot in untrodden land and forests as they explore nature with a tool

13. APPENDIX

Source code for Website: https://github.com/IBM-EPBL/IBM-Project-477-1658303202/tree/main/Project%20Development%20Phase/Sprint%203

Source code for classification algorithm(Notebook file - IBM Cloud): https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/d8cf79ed-e0b3-44dd-b150-cf2ececc025d/view?projectid=9f99f93d-6e5b-44eb-ad3d-f2133b037f06&context=cpdaas

Git-hub repo: https://github.com/IBM-EPBL/IBM-Project-477-1658303202

Demo video: https://drive.google.com/file/d/1bFQl3geOph8DdaZtlolo5xLROPRkG3cY/view?usp=sharing