IBM – NALAIYATHIRAN PROJECT

Digital Naturalist-AI tool Enabled for Bio Diversity Researchers

Team Members

NIVETHAA S

PRASHIDHA S

NIVETHA S

ARUNA B

Project Report Format

1. Introduction

1.1 Overview

A naturalist is someone who studies the patterns of nature, identifies a different kind of flora and fauna in nature. Being able to identify the flora and fauna around us often leads to an interest in protecting wild spaces, and collecting and sharing information about the species we see on our travels is very useful for conservation groups like NCC.

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.

Field naturalists can only use this web app from anywhere to identify the birds, flowers, mammals and other species they see on their hikes, canoe trips and other excursions.

In this project, we are creating a web application which uses a deep learning model, trained on different species of birds, flowers and mammals (2 subclasses in each for a quick understanding)and get the prediction of the bird when an image is been given.

1.2 Purpose

Naturalist can easily use this Web Application at any time at any where

for reasearch purpose.

People can also utilize this Web Application to identify any plants, birds, mammals and getting knowledge.

2. Literature Survey

2.1 Existing Problem

- **Always need to carry heavy books to refer about the species.**
- **Need to depend on the experts to clarify the doubt about the species.**
- **Only expert knows details others don't know about it.**

2.2 References

1. McDaniel, and Karen M Warkentin. 2009. Frequency information in the vibration-cued escape hatching of red-eyed treefrogs. The Journal of experimental P Agre. 1997. Toward a Critical Technical Practice: Lessons Learned in Trying to Reform Al. Social science, technical systems, and cooperative. Retrieved October 25, 2013 from

2. Michael S Caldwell, J Gregory biology 212, Pt 4: 566--75.Google ScholarCross Ref

3. Rémy Chauvin. 1977. Ethology: the biological study of animal behavior. International Universities Press. Retrieved November 7, 2012 from I D Couzin and N R Franks. 2003. Self-organized lane formation and optimized traffic flow in army ants. Proceedings. Biological sciences / The Royal Society 270, 1511: 139--46.Google ScholarCross Ref

4. Robert P. Crease. 1993. The Play of Nature: Experimentation as Performance (Indiana Series in the Philosophy of Technology). Indiana University Press. Retrieved April 10, 2013

5. M Descombe. 2010. Good Research Guide?: For small-scale social research projects (2nd Edition). For Samll. scale Social: 319.

6. P Agre. 1997. Toward a Critical Technical Practice: Lessons Learned in Trying to Reform Al. Social science, technical systems, and cooperative. Retrieved October 25, 2013.

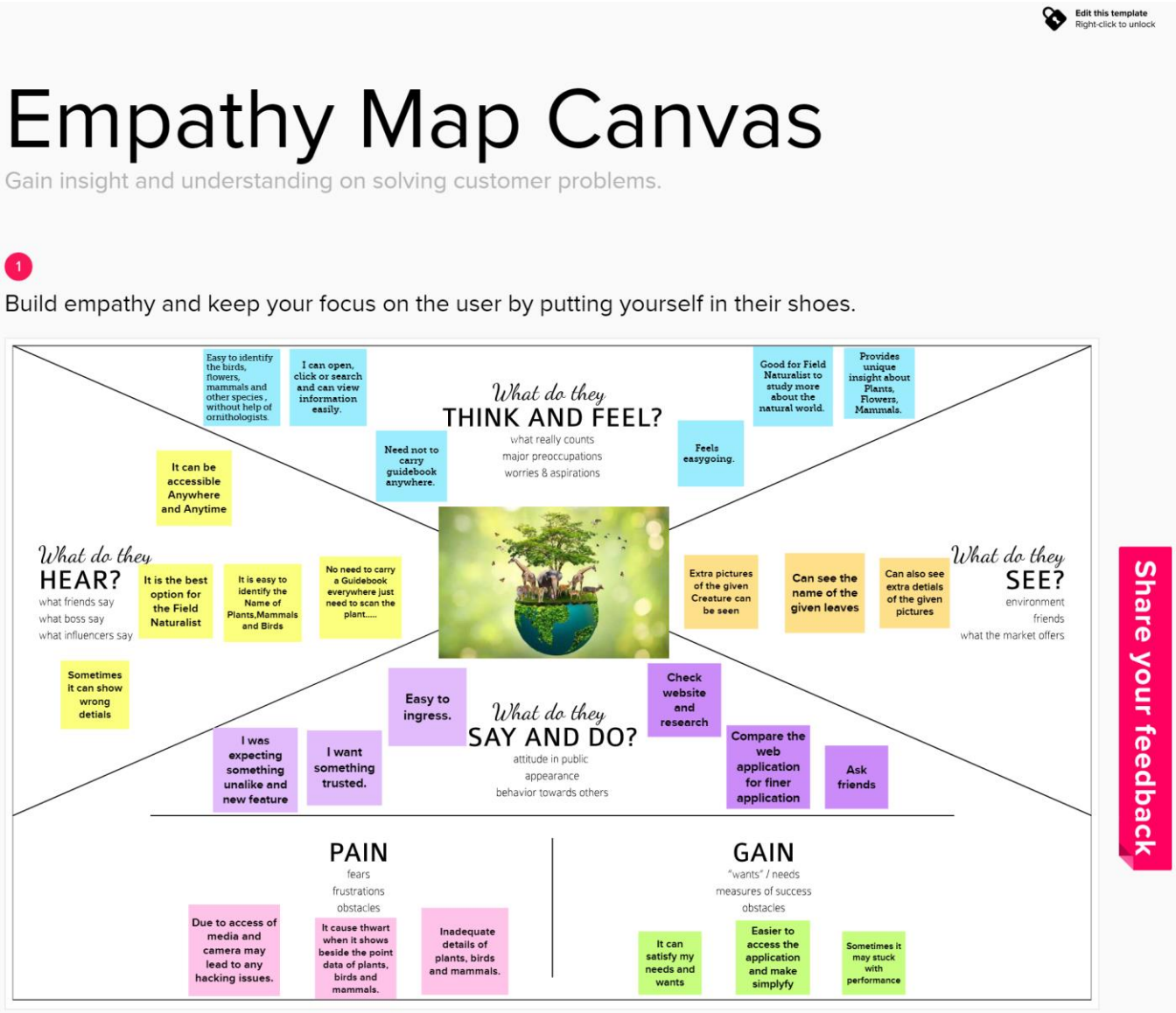
2.3 Problem Statement

Identifying different species of flora and fauna just based on human knowledge is impractical due to the sheer number of species that exist. Being able to identify flora and fauna around us often leads to an interest in protecting wild spaces, and collecting and sharing information about the species. Hence, it is necessary to be able to identify them accurately.

To build a CNN-based model that can classify the different species of birds, mammals and flowers and get the prediction of the bird when an image is given. The classification model will be built as a web app with a simple UI for easy usability. This application can be used by naturalists to explore and identify the various species present in nature.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

3.3 Proposed Solution

1. Problem Statement (Problem to be solved)

Identifying different species of flora and fauna just based on human knowledge is impractical due to the sheer number of species that exist. Being able to identify flora and fauna around us often leads to an interest in protecting wild spaces, and collecting and sharing information about the species. Hence, it is necessary to be able to identify them accurately.

1 To build a CNN-based model that can classify the different species of birds, mammals and flowers and get the prediction of the bird when an image is given. The classification

model will be built as a web app with a simple UI for easy usability. This application can be used by naturalists to explore and identify thevarious species present in nature.

2. Idea / Solution description • Frequently update the database. • Get reviews from users. • Feedback through voice and message. • Language translation. • Show similar images with description. • Shows their habits, habitats, living and grouping.

3. Novelty / Uniqueness Finding the plant just by scanning the leaves. No further details is essential. It shows the information like their habits, habitats, etc,… Field naturalists can use this web app from anywhere to identify the birds, flower, mammals and other species they see on their hikes canoe trips and other excursions.

4. Social Impact / Customer Satisfaction Being able to identify the flora and fauna around us often leads to an interest in protecting the world species. Collecting and sharing information about the species we see on our travels is very useful for conserving groups like NCC.

5. Business Model (Revenue Model) Can make money through subscription based. Partnership with Laboratories and Researchers around the world.

6. Scalability of the Solution Can use the web app anytime and anywhere around the world. Database is updated Frequently. More then 2 Language Description is available. According to the customers review we can add extra information if they point out. If there is any issue in using the app we can sort out

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

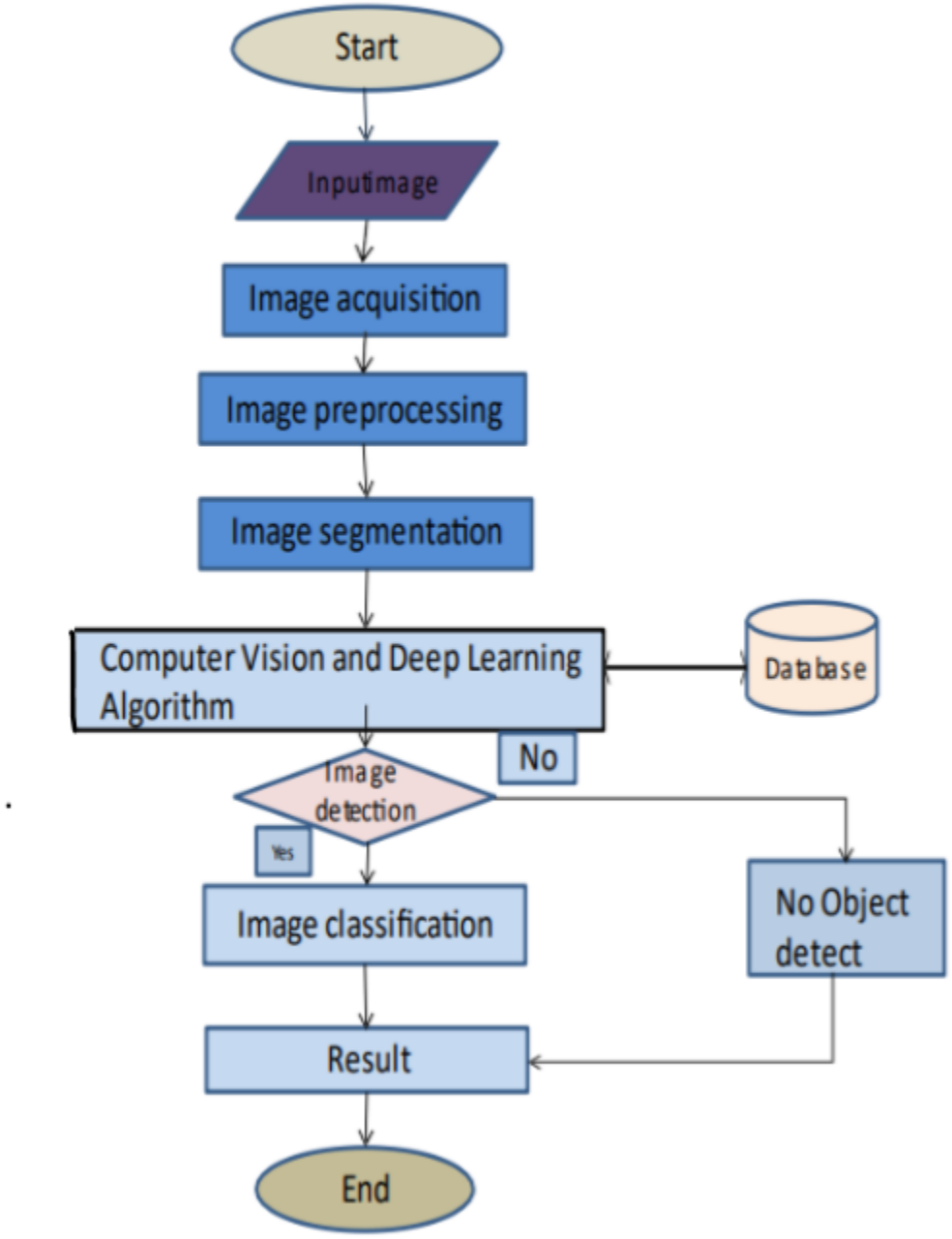
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Mobile Number Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via Mobile Number
FR-3	Login	Create userid and password for logging in during fistr login. Use this for future use of web app.
FR-4	Scanning	Scan a image using your device camera.
FR-5	uploading	Can upload an image from your device.
FR-6	Displaying Detials	It Shows a details about the species. Lot of details will be available in our web app.

4.2 Non-Functional requirements

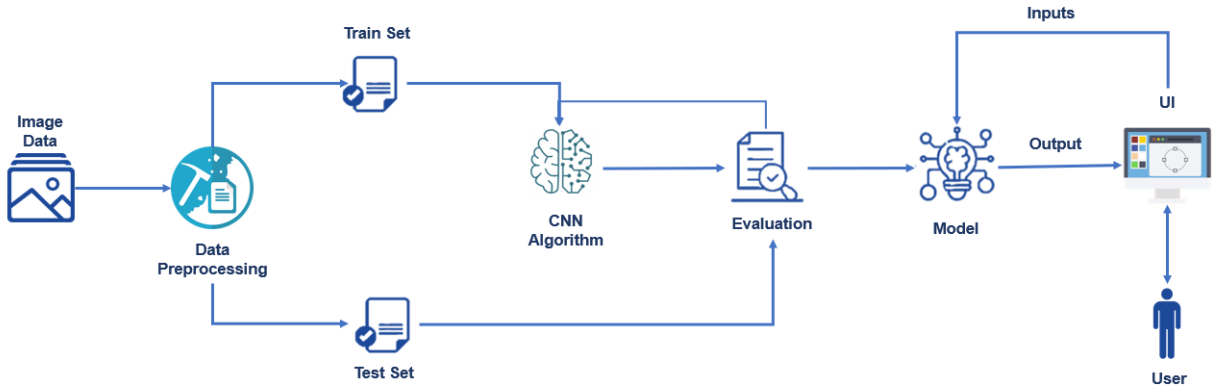
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Naturalist can use this application to get the details about the species by just scanning the species or by uploading the image of the species.
NFR-2	**Security**	Particular password is there for you. No one can able to access the web app without the knowledge of user since there will be separate password for every user. One Time Password will be generated.
NFR-3	**Reliability**	All species information will be available in our web app .Lot of information will be available in our web app. Customers reviews are always welcome
NFR-4	**Performance**	Response will be fast. Correct information will be provided. Can be even accessible in lower bandwidth.
NFR-5	**Availability**	Will be available all the time. Easy to access. Will make available for everyone. It can be make available wherever and whenever.
NFR-6	**Scalability**	Will update the details according to the user needs. It works more efficienty. Demands of the customer will be updated.

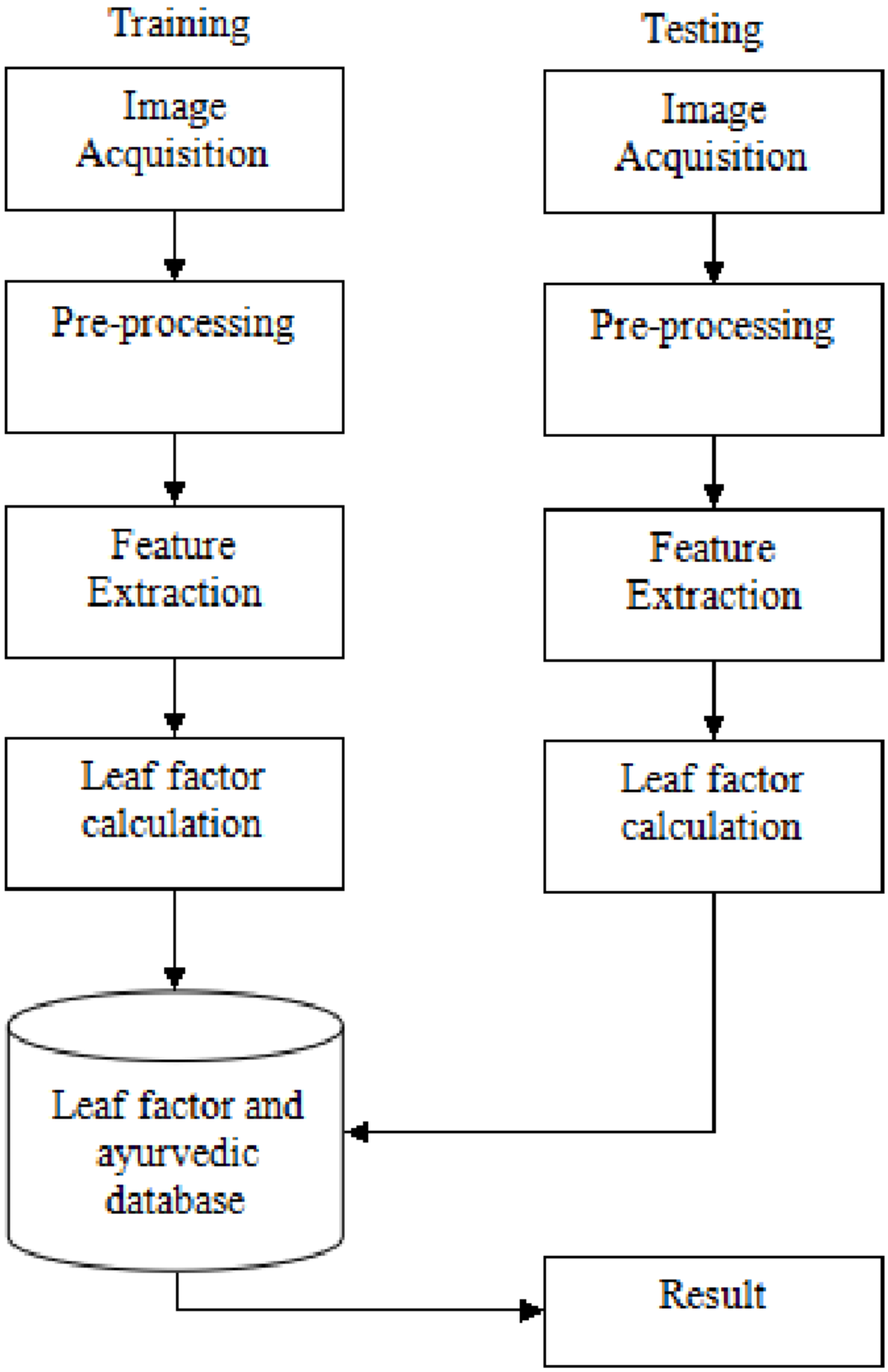
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture




Training
Testing
Image Acquisition
Image Acquisition
Pre-processing
Pre-processing
Feature Extraction
Feature Extraction
Leaf factor calculation
Leaf factor calculation
Leaf factor and ayurvedic database
Result

5.3 User Stories

1. Collection of all dataset for pre processing
2. Augment the collection of dataset
3. Loading the dataset and pre processing the images.
4. Perform CNN with pre processed dataset.
5. Training and Testing the model
6. Save the trained model
7. Build the Application for identification
8. As a user, I can register for the application by entering my email, password, and confirming my password
9. As a user, I can log into the application by entering email & password
10. Indentify the given images
11. Logout form the application

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

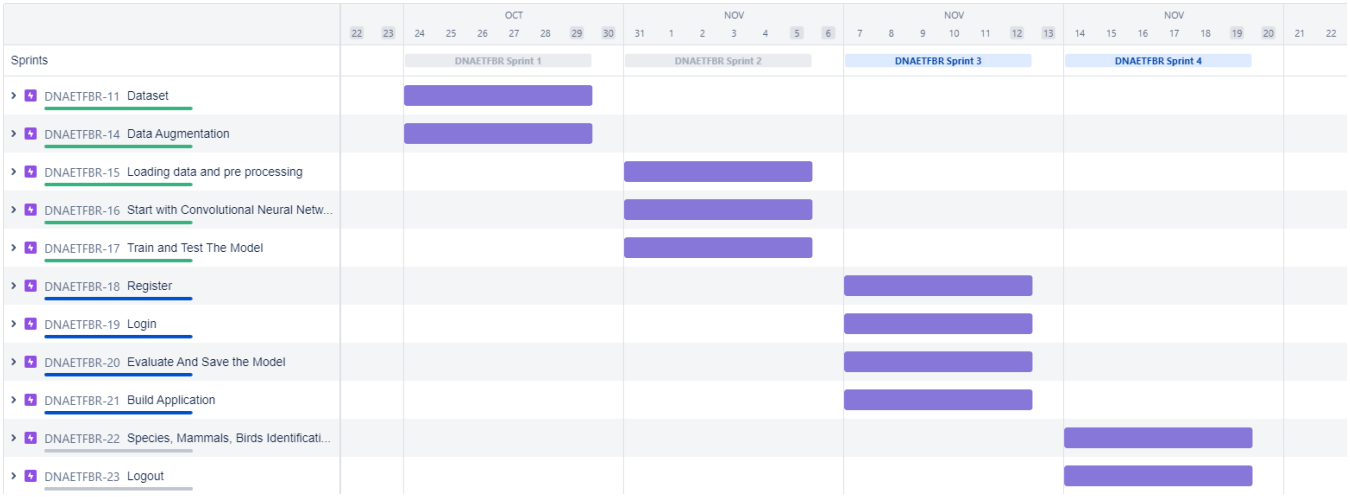
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Nivethaa S Prashidha S
Sprint-1	Data Augmentation	USN-4	Augment the collection of dataset	5	Medium	Prashidha S Nivetha S
Sprint-3	Login	USN-5	As a user, I can log into the application by entering email & password	2	High	Nivethaa S Prashidha S
Sprint-1	Dataset	USN-6	Collection of all dataset for pre processing	5	High	Aruna B Nivetha S
Sprint-2	Loading data and pre processing	USN-7	Loading the dataset and pre processing the images.	2	Medium	Aruna B Nivethaa S
Sprint-2	Start with Convolutional Neural Network	USN-8	Perform CNN with pre processed dataset.	3	Medium	Prashidha S Nivetha S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Train and Test the model	USN-9	Training and Testing the model	2	High	Nivethaa S Prashidha S
Sprint-3	Evaluate and save the model	USN-10	Save the trained model	4	Medium	Aruna B Nivetha S
Sprint-3	Build application	USN-11	Build the Application for identification	4	High	Aruna B Nivethaa S
Sprint-4	Species, Mammals, Birds identification	USN-12	Indentify the given images	5	High	Prashidha S Nivetha S
Sprint-4	Logout	USN-13	Logout form the application	3	Low	Nivethaa.S Prashidha S

6.2 Sprint Delivery Schedule

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	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	5 Nov 2022
Sprint-3	12	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	4	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

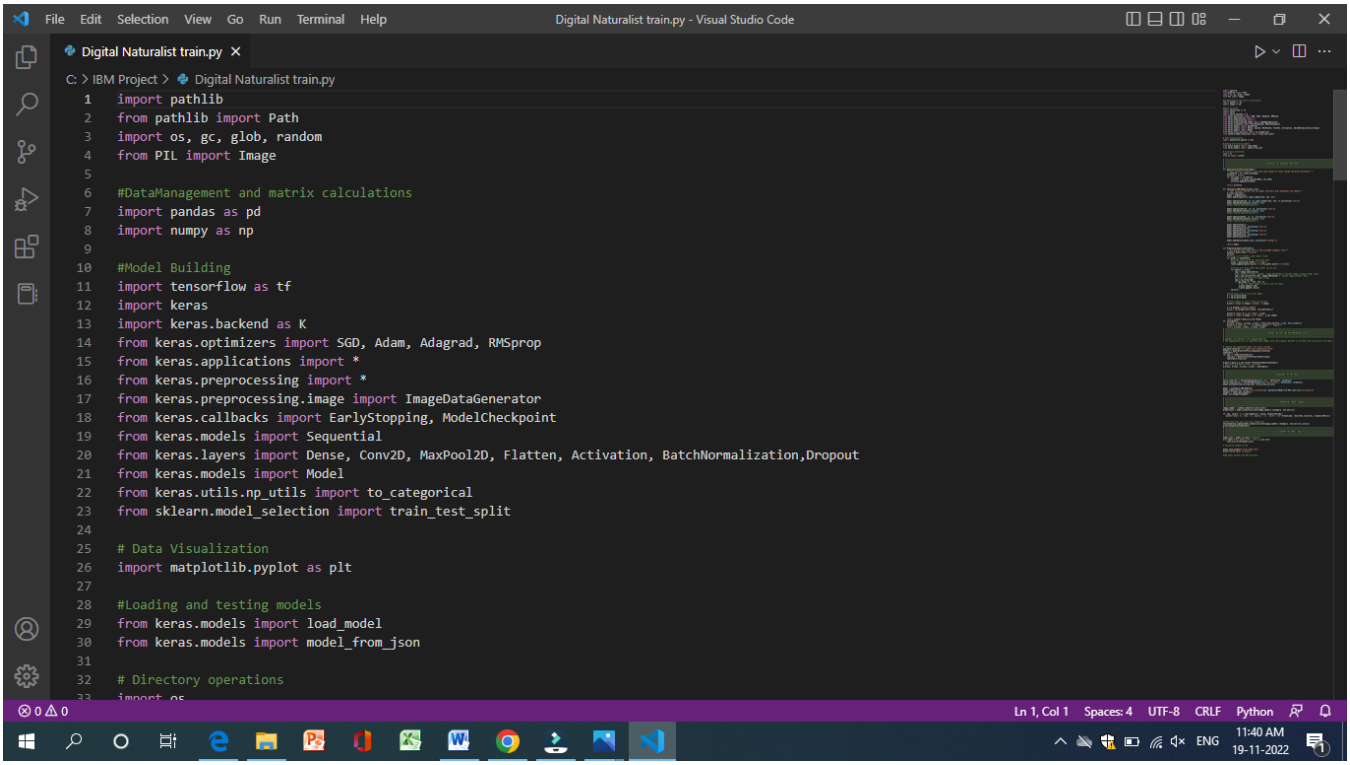
6.3 Reports from JIRA



7. CODING AND SOLUTIONING

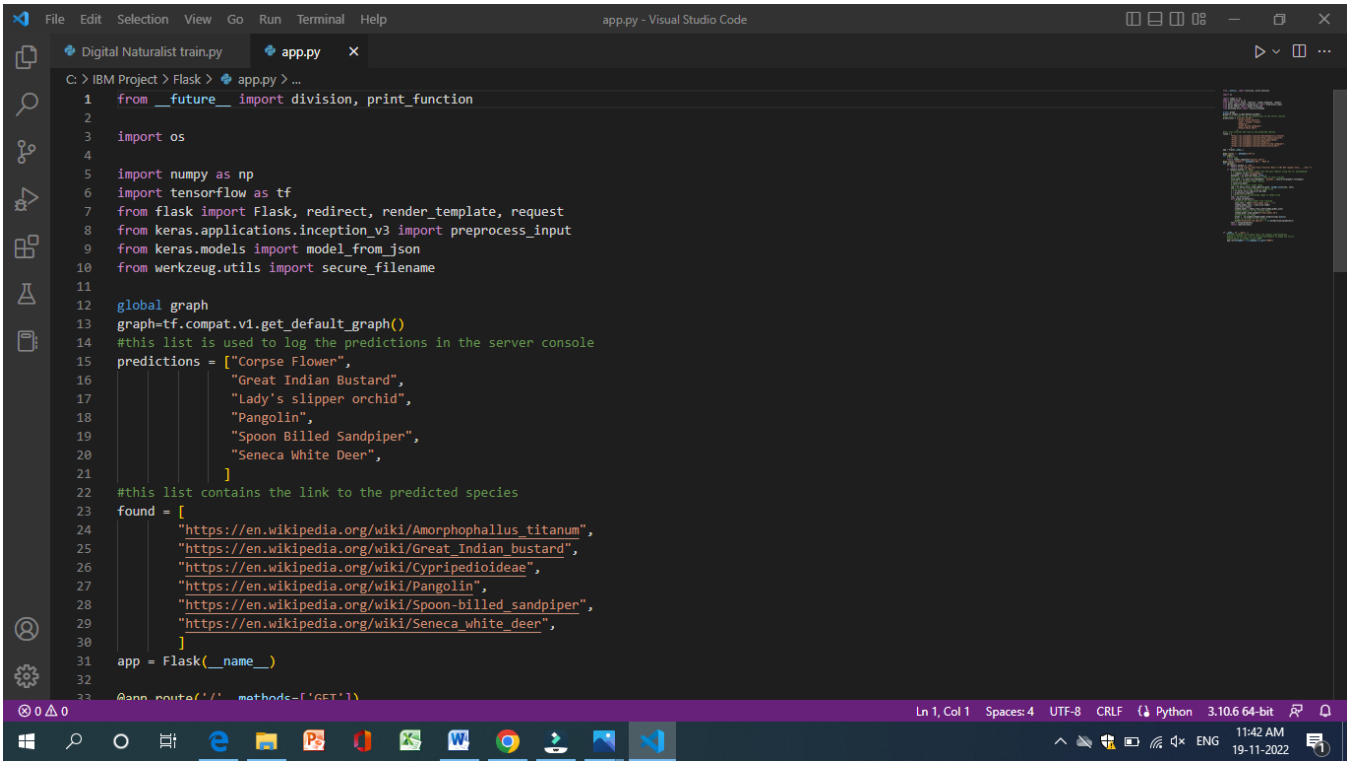
7.1 Feature 1

User just need to upload the image of the birds, animals or plants and they will get the name of the species.



7.2 Feature 2

By predicting the species, the website is redirected to the website where they can see all the details of the species.


File Edit Selection View Go Run Terminal Help
app.py - Visual Studio Code
Digital Naturalist train.py
app.py
C: > IBM Project > Flask > app.py > ...
from __future__ import division, print_function
import os
import numpy as np
import tensorflow as tf
from flask import Flask, redirect, render_template, request
from keras.applications.inception_v3 import preprocess_input
from keras.models import model_from_json
from werkzeug.utils import secure_filename
global graph
graph=tf.compat.v1.get_default_graph()
#this list is used to log the predictions in the server console
predictions = ["Corpse Flower",
"Great Indian Bustard",
"Lady's slipper orchid",
"Pangolin",
"Spoon Billed Sandpiper",
"Seneca White Deer",
]
#this list contains the link to the predicted species
found = [
"https://en.wikipedia.org/wiki/Amorphophallus_titanum",
"https://en.wikipedia.org/wiki/Great_Indian_bustard",
"https://en.wikipedia.org/wiki/Cypripedioideae",
"https://en.wikipedia.org/wiki/Pangolin",
"https://en.wikipedia.org/wiki/Spoon-billed_sandpiper",
"https://en.wikipedia.org/wiki/Seneca_white_deer",
]
app = Flask(__name__)
@app.route('/', methods=['GET'])
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.6 64-bit
ENG
11:42 AM
19-11-2022

8. TESTING

8.2 User Acceptance Testing

Introduction

Effectively documenting incidents during the testing process is improving software or processes before a system is released. Sometimes, the testers themselves document issues they encounter; but more often, a UAT coordinator verifies, consolidates, and classifies reported issues before assigning them to the appropriate group to address. Then, that IT coordinator again validates and prioritizes the technical issues before handing them off to an IT developer to investigate further and resolve.

During the course of UAT, it is inevitable that issues will be discovered. It is shocking how often documented issues contain insufficient data to facilitate a quick and thorough investigation.

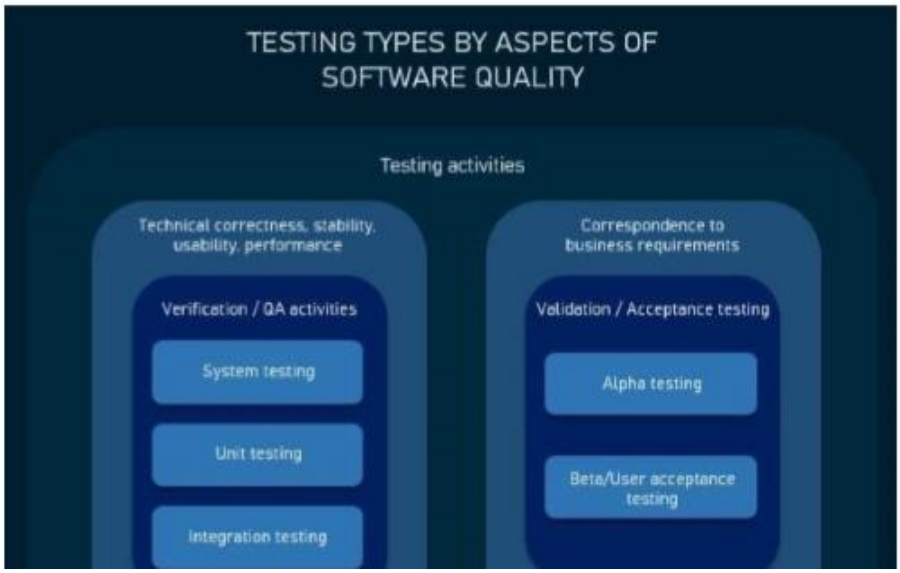
Deliverables of UAT

Every interviewer very quickly stated that UAT is to assure quality. Project managers also stated that it can double as a training exercise for business users as well as ensuring that the requirements set match the functionality that is desired from the system. People managers expressed that one of the most important deliverables is the decision to go forward with the update or new system; the ”green” or ”red” light. Individual contributors

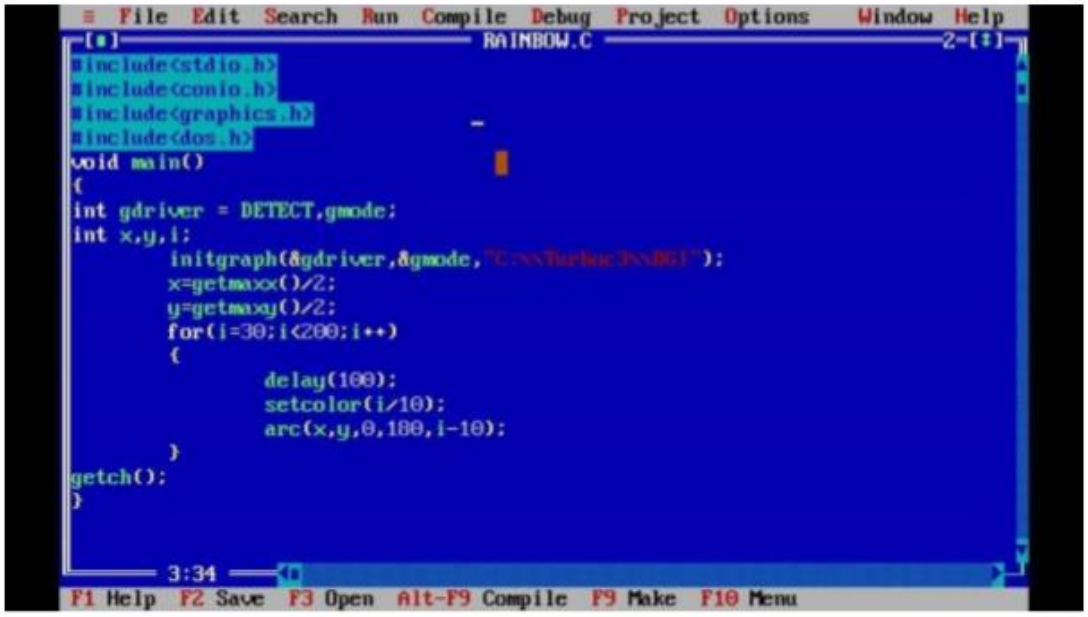
included in the development and actually have a say in what works and what doesn't. Individual contributors stated that they felt that UAT has been done enough when the tests they are running are all success full but that it is a gut-feeling or intuition that says when they are content with the testing. They also stated many perks of UAT such as: learning the new system, cooperation between departments, learning something new, feeling valued by the company and inclusion in decision making. Project Managers stated that the organization at large sometimes acted as though it had forgotten the purpose of UAT - to assure quality and usability of a release.

Test Quality

From the production logs of the SUT a Markov-chain with 68 states (one for each application feature that was left after filtering out non-relevant states) was created. Due to the fact that the SUT was a regular release of an existing system, and not a newly adopted software, a transition matrix could be made on a per-tester level for both the production system logs, as well as the test system logs. Variability due to changes in logging were taken into account by qualitatively examining the log files. As transition matrices for both TEST and PROD had been computed, a similarity score could be computed to directly and in bulk estimate the quality of the testing.



- The assembly code is sent to assembler which assembles the code and converts it into object code.



Usually, when possible, this testing happens in a conference or a war room sort of a set up where the users, PM, QA team representatives all sit together for a day or two and work through all the acceptance test cases.

Once all the tests are run and the results are in hand, the **Acceptance Decision** is made. This is also called the **Go/No-Go decision**. If the users are satisfied it's a Go, or else it's a No-go.

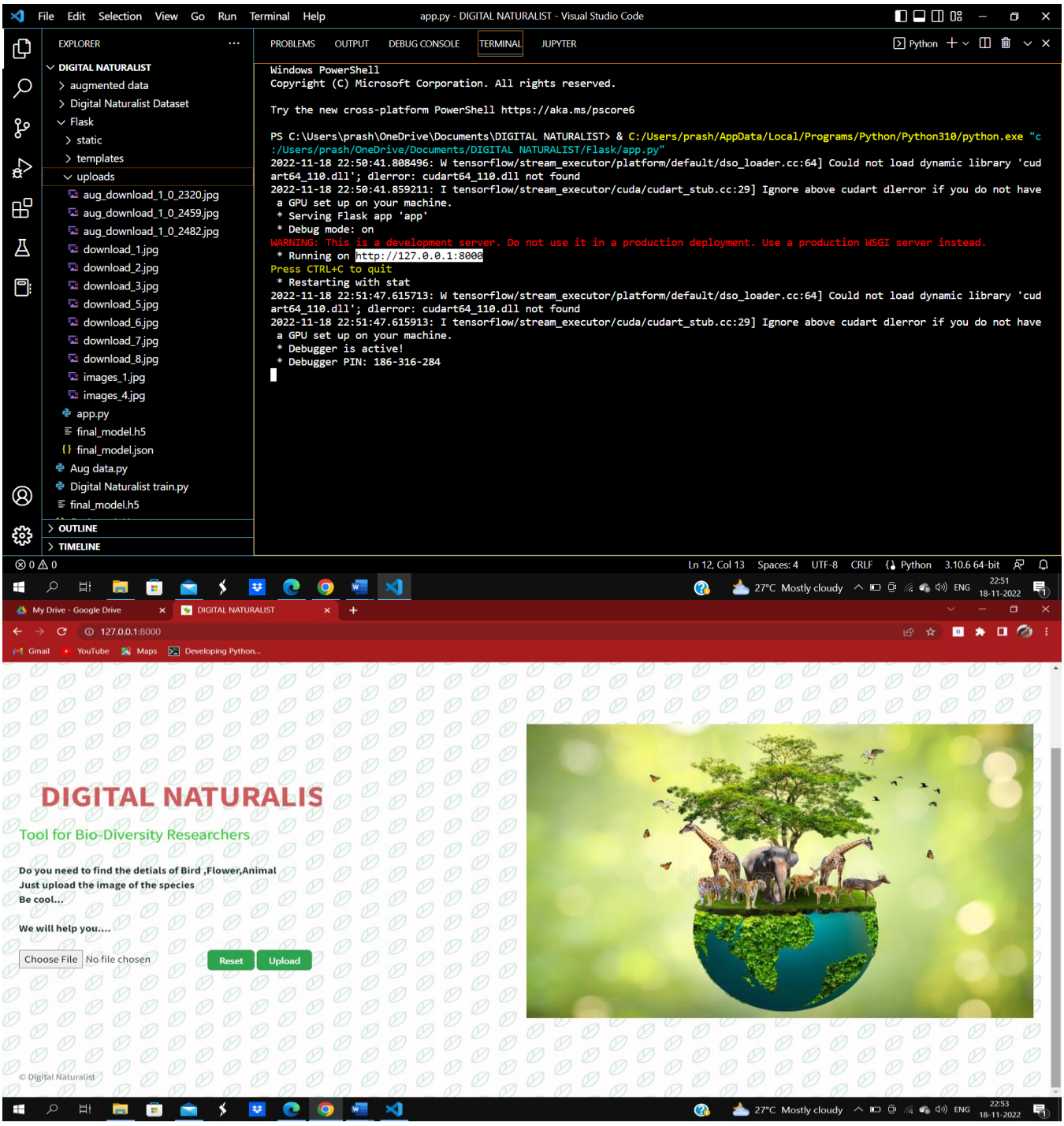
underlying **assumption** even before this test begins is that all that basic stuff is tested and is working fine. God forbid, the users find a bug as basic as that – it is a piece of very bad news for the QA team.

This testing is about the entity that is the primary element in the business.

```
class Program
{
    0 references
    static async Task Main(string[] args)
    {
        WriteLine("Please type the username for the desired user:");
        var username = ReadLine();

        var github = new GitHubClient(new ProductHeaderValue("MyAmazingApp"));

        try
        {
            var user = await github.User.Get(username);
            WriteLine($"The user {user.Name} was succesfully retrieved!");
            WriteLine($"{user.Name} has {user.PublicRepos} public repositories. Do you want to see the list? (y/n)");
            var response = ReadLine();

            if (string.Equals(
                "Y",
                response,
                StringComparison.InvariantCultureIgnoreCase))
            {
```

9. RESULTS

9.1 Performance Metrics



My Drive - Google Drive | Great Indian bustard | WWF India

wwfindia.org/about_wwf/priority_species/threatened_species/great_indian_bustard/

Gmail YouTube Maps Developing Python...

WWF

About us | Our work | Get involved | Media | Earth Hour | Nature Store | DONATE

Home » Our work » Priority species » Threatened Species » Great Indian bustard

- Our work
- Priority species
- Threatened Species
  - Common leopard
  - Great Indian bustard
  - Himalayan quail
  - House sparrow
  - Sarus Crane
  - Smooth-coated otter
  - Nilgiri tahr
  - Gharial
  - Asiatic lion
  - Ganges river dolphin
  - Black-necked crane
  - Golden mahseer
  - Indian pangolin
  - Brow-antlered deer

Great Indian bustard

Key Facts

Common Name
Great Indian bustard

Scientific Name
Ardeotis nigriceps

Population
200 individuals worldwide

Height
100 cms or 1 metre

Length
Wingspan of 210-250 cm

Weight:
15-18 kg

Status
Listed in Schedule I of the Indian Wildlife (Protection)Act, 1972, in the CMS Convention and in Appendix I of CITES, as Critically Endangered on the IUCN Red List and the National Wildlife Action Plan (2002-2016). It has also been identified as one of the species for the recovery programme under the Integrated Development of Wildlife Habitats of the Ministry of Environment d Forests, Government of India.

© Asad R. Rahmani

DONATE

https://www.wwfindia.org/about_wwf/priority_species/threatened_species/ganges_river_dolp...

27°C Mostly cloudy ENG 22:53 18-11-2022

10. ADVANTAGES & DISADVANTAGES

Advantages

1. Naturalist can always use this web application at where, at any time.

2. Easy to access.

3. People also use this web application for identification.

4. It shows description about flowers, birds, mammals.

Disadvantages

1. Sometimes it may stuck with efficiency.

2. Always need internet to access the web application**.**

11. CONCLUSION

Field naturalist can use this web application from anywhere to identify the birds, flowers, mammals and other species the see on their hikes, canoe trips and other excursions.

In this project, we are creating a web application which uses a deep learning and CNN mode, trained on different species of birds, flowers, and mammals. There is great diversity among naturalists, but some common ground too. All naturalism begins with an admiring attitude is combined with a contempt or distrust for the way that philosophy has been or is conducted.

This project uses for both Digital Naturalist and common people for indentification the birds, flowers and mammals.

12. FUTURE SCOPE

This project can enhance with many features for indentifying and describing the flowers, birds, and mammals. This web app is used for many purpose .And through this web app people can also get some knowledge about flowers, birds and mammals.

This web application will use for business model and make business with this application to other companies**.**

13. APPENDIX

Source code

Augmented Data.py

```
from keras.preprocessing.image import ImageDataGenerator
import cv2
from os import listdir
import time


def hms_string(sec_elapsed):
    h = int(sec_elapsed / (60 * 60))
    m = int((sec_elapsed % (60 * 60)) / 60)
    s = sec_elapsed % 60
    return f"{h}:{m}:{round(s,1)}"


def augment_data(file_dir, n_generated_samples, save_to_dir):
    data_gen = ImageDataGenerator(rotation_range=30,
                                  width_shift_range=0.1,
                                  height_shift_range=0.15,
```

```
                    shear_range=0.25,
                    zoom_range = 0.2,
                    horizontal_flip=True,
                    vertical_flip=False,
                    fill_mode='neares
                   brightness_range=(0.5,1.2)
                    )


    for filename in listdir(file_dir):
        image = cv2.imread(file_dir + '/' + filename)
        image = image.reshape((1,)+image.shape)
        save_prefix = 'aug_' + filename[:-4]
        i=0
        for batch in data_gen.flow(x=image, batch_size=1, save_to_dir=save_to_dir,
                                   save_prefix=save_prefix, save_format='jpg'):
            i += 1
            if i > n_generated_samples:
                break
start_time = time.time()


augmented_data_path = r'E:\IBM Project\augmented data'
augment_data(file_dir=r'E:\IBM Project\augmented data\Bird\GIB_AUG',
n_generated_samples=8, save_to_dir=augmented_data_path+'/Bird/GIB_AUG')
```

```
augment_data(file_dir=r'E:\IBM Project\augmented data\Bird\SPS_AUG',
n_generated_samples=8, save_to_dir=augmented_data_path+'/Bird/SPS_AUG')

augment_data(file_dir=r'E:\IBM Project\augmented data\Flower\Corpse_AUG',
n_generated_samples=8, save_to_dir=augmented_data_path+'/Flower/Corpse_AUG')

augment_data(file_dir=r'E:\IBM Project\augmented data\Flower\LS_Orchid_AUG',
n_generated_samples=8, save_to_dir=augmented_data_path+'/Flower/LS_Orchid_AUG')

augment_data(file_dir=r'E:\IBM Project\augmented data\Mammal\LS_Pangolin_AUG',
n_generated_samples=8, save_to_dir=augmented_data_path+'/Mammal/Pangolin_AUG')

augment_data(file_dir=r'E:\IBM Project\augmented data\Mammal\SW_Deer_AUG',
n_generated_samples=8,
save_to_dir=augmented_data_path+'/Mammal/SW_Deer_AUG')



end_time = time.time()

execution_time = (end_time - start_time)

print("Elapsed Time : "+str(execution_time))
```

Digital Natiralist train.py

```
import pathlib

from pathlib import Path

import os, gc, glob, random

from PIL import Image


#DataManagement and matrix calculations

import pandas as pd

import numpy as np
```

```
#Model Building
import tensorflow as tf
import keras
import keras.backend as K
from keras.optimizers import SGD, Adam, Adagrad, RMSprop
from keras.applications import *
from keras.preprocessing import *
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPool2D, Flatten, Activation,
BatchNormalization,Dropout
from keras.models import Model
from keras.utils.np_utils import to_categorical
from sklearn.model_selection import train_test_split

# Data Visualization
import matplotlib.pyplot as plt

#Loading and testing models
from keras.models import load_model
from keras.models import model_from_json

# Directory operations
import os
```

```
from os import listdir


# ========================================DEFINING THE
REQUIRED
FUNCTIONS================================================
======= #


def generateListofFiles(dirName):
    """This function returns a list with exact paths of files inside the given directory """
    listOfFile = os.listdir(dirName)
    allFiles = list()
    for fol_name in listOfFile:
        fullPath = os.path.join(dirName, fol_name)
        allFiles.append(fullPath)


    return allFiles


def Configure_CNN_Model(output_size):
    """This function defines the cnn model structure and configures the layers"""
    K.clear_session()
    model = Sequential()
    model.add(Dropout(0.4,input_shape=(224, 224, 3)))


    model.add(Conv2D(256, (5, 5),input_shape=(224, 224, 3),activation='relu'))
    model.add(MaxPool2D(pool_size=(2, 2)))
    #model.add(BatchNormalization())
```

```
    model.add(Conv2D(128, (3, 3), activation='relu'))
    model.add(MaxPool2D(pool_size=(2, 2)))
    #model.add(BatchNormalization())

    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPool2D(pool_size=(2, 2)))
    #model.add(BatchNormalization())

    model.add(Flatten())
    model.add(Dense(512, activation='relu'))
    model.add(Dropout(0.3))
    model.add(Dense(256, activation='relu'))
    model.add(Dropout(0.3))
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.3))

    model.add(Dense(output_size, activation='softmax'))

    return model

def PrepreocessData(subfolders):
    """Pre precess the image data in the provided category list"""
    X_data,Y_data,found = [],[],[]
    id_no=0
```

```
    #itering in all folders under Boats folder
    for paths in subfolders:
        #setting folder path for each boat type
        files = glob.glob (paths + "/*.jpg")
        found.append((paths.split('\\')[-2],paths.split('\\')[-1]))

        #itering all files under the folder one by one
        for myFile in files:
            img = Image.open(myFile)
            #img.thumbnail((width, height), Image.ANTIALIAS) # resizes image in-place keeps ratio
            img = img.resize((224,224), Image.ANTIALIAS) # resizes image without ratio
            #convert the images to numpy arrays
            img = np.array(img)
            if img.shape == ( 224, 224, 3):
                # Add the numpy image to matrix with all data
                X_data.append (img)
                Y_data.append (id_no)
        id_no+=1

    #converting lists to np arrays again
    X = np.array(X_data)
    Y = np.array(Y_data)

    # Print shapes to see if they are correct
```

```
    print("x-shape",X.shape,"y shape", Y.shape)

    X = X.astype('float32')/255.0
    y_cat = to_categorical(Y_data, len(subfolders))

    print("X shape",X,"y_cat shape", y_cat)
    print("X shape",X.shape,"y_cat shape", y_cat.shape)

    return X_data,Y_data,X,y_cat,found;
def splitData():
    X_train, X_test, y_train, y_test = train_test_split(X, y_cat, test_size=0.2)
    print("The model has " + str(len(X_train)) + " inputs")
    return X_train, X_test, y_train, y_test
# ========================================================================================== #

# ========================================================================================== #

# ========================================LOADING THE DATA AND PRE-PROCESSING DATA========================================== #


# Augument the datasets with AugumentData.py.
# The AugumentData.py will generate many images with the original dataset to increase the accuracy of the model.
```

```
# Loading the augumented data form local storage
aug_data_location =r"C:\IBM Project\augmented data"
Folders = generateListofFiles(aug_data_location)
subfolders = []
for num in range(len(Folders)):
    sub_fols = generateListofFiles(Folders[num])
    subfolders+=sub_fols

X_data,Y_data,X,y_cat,found= PrepreocessData(subfolders)
# Splitting the data to Test and Train
X_train, X_test, y_train, y_test = splitData()



# =====BUILDING THE CNN
MODEL================================================== #

early_stop_loss = EarlyStopping(monitor='loss', patience=3, verbose=1)
early_stop_val_acc = EarlyStopping(monitor='val_accuracy', patience=3, verbose=1)
model_callbacks=[early_stop_loss, early_stop_val_acc]

model = Configure_CNN_Model(6)
model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.001),metrics=['acc
uracy'])
```

```
weights = model.get_weights()
model.set_weights(weights)



# ====================PREDECTING IMAGE CLASSES================================================== #


image_number = random.randint(0,len(X_test))
predictions = model.predict([X_test[image_number].reshape(1, 224,224,3)])


for idx, result, x in zip(range(0,6), found, predictions[0]):
   print("Label: {}, Type : {}, Species : {} , Score : {}%".format(idx, result[0],result[1], round(x*100,3)))



#predicting the class with max probability
ClassIndex=np.argmax(model.predict([X_test[image_number].reshape(1, 224,224,3)]),axis=1)
print(found[ClassIndex[0]])


# ==========================================================SAVING THE MODEL LOCALLY================================================== #


model_json = model.to_json() #indent=2
with open("final_model.json", "w") as json_file:
```

```
    json_file.write(model_json)

# serialize weights to H5

model.save_weights("final_model.h5")
print("Saved model to disk")

#CNN model tested with 86% accuracy
```

digital.html

```
<!DOCTYPE html>
<html lang="en">

<head>

                        <meta charset="UTF-8">
                        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
                        <meta http-equiv="X-UA-Compatible" content="ie=edge">
                        <title>DIGITAL NATURALIST</title>
                        <meta name="description" content="">
                        <meta name="keywords" content="">

                        <link rel="icon" type="image/x-icon"
href="https://greatbay.org/wp-content/uploads/2020/06/i-naturalist-150x150.png">
```

```
                    <link
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700"
rel="stylesheet">
                    <link rel="stylesheet" href="/static/style.css">


</head>




<body class="leading-normal tracking-normal text-gray-900" style="font-family: 'Source
Sans Pro', sans-serif;">


                    <h1>



                    <div class="h-screen pb-14 bg-right bg-cover">
                        <!--Nav-->
                        <div class="w-full container mx-auto p-6">



                        </div>


                        <!--Main-->
                        <div class="container pt-24 md:pt-48 px-6 mx-auto flex flex-
wrap flex-col md:flex-row items-center">
```

<!--Left Col-->

<h1

class="my-4 text-3xl md:text-5xl text-green-800 font-bold leading-tight text-center md:text-left slide-in-bottom-h1">

 <marquee>

DIGITAL NATURALIST</marquee></h1>

Tool for Bio-Diversity Researchers

Do you need to find the detials of Bird ,Flower,Animal

Just upload the image of the species
Be cool...

We will help you....

<!DOCTYPE html>

<form action="/predict" id="upload-file" method="post" enctype="multipart/form-data">

<input type="file" name="uploadedimg" id="uploadedimg" required accept=".jpg, .png, .jpeg, .gif, .bmp, .tif, .tiff|image/*" >

<input type="reset" value="Reset" class="upload">

<input type="submit" value="Upload" class="upload" onsubmit="check_file">

</form>

<!--Right Col-->

<img class="w-5/6 mx-auto lg:mr-0 slide-in-bottom" src="https://thumbs.dreamstime.com/b/concept-nature-reserve-conserve-wildlife-tiger-deer-global-warming-food-loaf-ecology-human-hands-protecting-wild-animals-228502195.jpg">

<!--Footer-->

<a class="text-gray-500 no-underline hover:no-underline"

href="https://github.com/IBM-EPBL/IBM-Project-46326-1660745221">© Digital Naturalist

```
                    </div>


                </div>

<script>
document.getElementById("uploadedimg").addEventListener("change", validateFile)

function validateFile(){
  const allowedExtensions =  ['jpg','png'],
        sizeLimit = 1_000_000;
  const { name:fileName, size:fileSize } = this.files[0];
  const fileExtension = fileName.split(".").pop();
  if(!allowedExtensions.includes(fileExtension)){
    alert("Only image files - .jpg, .jpeg, .png, .tiff ");
    this.value = null;
  }else if(fileSize > sizeLimit){
    alert("file size too large")
    this.value = null;
  }
}</script>

</body>

</html>
```

Git Hub & Project Demo Link

- ✓ https://drive.google.com/file/d/1AYufC0tZnLoTwy39TPIBCwHgJKUQuOjx/view?usp=share_link
- ✓ https://github.com/IBM-EPBL/IBM-Project-45191-1660728751.git
- ✓ https://www.youtube.com/watch?v=ij4sWvz-Gos