DIGITAL NATURALIST

Using deep learning

Developed by: Saloni MP, Chandana M, Aparna Rai, Neha UK, Banushree DJ

Smart Bridge-Remote Summer Internship Program

1. INTRODUCTION

A naturalist is someone who studies the patterns of nature, identify different kingdom of flora and fauna in the nature, and identify different kind of flora and fauna in the nature. Being able to identify the flora and fauna around us often leads to an interest in protecting wild species, and collecting and sharing information about the species we see on our travels is very useful for conservation groups like NCC. Natural history is a domain of inquiry involving organisms, including animals, plants, fungi, in their natural environment, leaning more towards observational than experimental methods of study. A person who studies natural history or well-being is called a naturalist or natural historian. Natural history encompasses scientific research but is not limited to it. It involves the study of any category of natural objects or organisms. So, while it dates from studies in the ancient Greco-Roman-world and the medieval Arabic world, through to European Renaissance naturalists working in near isolation, today's naturalist is a cross-discipline umbrella of many specialty sciences; e.g. geobiology has a strong multidisciplinary nature.

The meaning of term "naturalist" or "natural-history" has narrowed progressively with time, while, by contrast, the meaning of the term "nature" has widened. In antiquity, "natural history" covered essentially anything connected with nature, or used materials drawn from nature.

Our task as naturalist has grown widely in the field of natural-historians. As our role has increased from identification to saviours as well. Not only identifying flora and fauna but also to know about their habits, habitats, living and grouping lead to fetching services for protection as well.

1.1 Overview

Deep-learning-based techniques and methods are becoming popular in digital naturalist studies, as their performance is superior in image analysis fields, such as object detection, image classification and semantic segmentation. Deep learning techniques have achieved state of-the-art performance for automatic segmentation of digital naturalist through multi-model image sensing. The Convolutional Neural Network (CNN) is a powerful method for image recognition and prediction. However, CNN is mostly used for flora and fauna segmentation, classification, and prediction of features and habitats.

More deep-learning-based methods that are utilized for natural living segmentation, classification, and prediction and by using the algorithm of a Flask model has been implemented and tested. Among all the deep learning methods and techniques, CNNs perform better for image segmentation, classification, and prediction.

1.2 Purpose

Our aim from the project is to make use of pandas, matplotlib, numpy, tensorflow, keras libraries from python to extract the libraries for deep learning for the flora and fauna prediction for digital naturalist. And in the end, to predict whether the image is of flora or fauna and withdrawing the conclusions.

2 LITERATURE SURVEY

Image segmentation plays a significant role in naturalist's image processing as natural images have different diversities. For this segmentation, they have used scanned images of animals and plants. A technique called image segmentation is presented for segmenting areas of interest in these images. The proposed algorithm specifies the issues that are associated with segmenting images and allows for fast, strong, and flexible subdivision without requiring true manual tracing. The boundary extraction problem is formulated as a Hidden Markov Model (HMM) and the present technique to the second-order Viterbi algorithm with state pruning is used to find the best boundary in a robust and efficient manner based on the pull out external and internal local costs, thus handling much inexact user boundary definitions than existing methods. Experimental results using images shows that the proposed algorithm achieves accurate segmentation in natural images without the need for accurate boundary definition as per existing Intelligent Scissors methods. For usability testing shows that the proposed algorithm requires less user interaction than Intelligent Scissors. The main advantage of using the proposed approach of user interaction over the conventional IS approach is that the user does not need to trace around the boundary carefully. These methods are refined based on image gradients, making it highly sensitive to contrast non-uniformities typically found in images. It is most vastly used for segmentation and classification brought in the accuracy 92% with a split ratio of 70:30 of 309 training set images and 134 testing set images, i.e. 70% of training images and 30% of testing images. In the future, they have planned to work with both plant and animal images, achieve more efficient segmentation. Working with a larger dataset will be more challenging in this aspect, and they have wanted to build a dataset emphasizing the abstract with respect to their country which will accelerate the scope of their work.

2.1 Existing problem

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

2.2 Proposed Solution

Flora and fauna at early stage is very difficult task for naturalists to identify. Images are more prone to noise and other environmental interference. So, it becomes difficult for digital naturalists to identify flora and fauna and their features. So here we come up with the system, where system will detect the concerning images. Here we convert image into grayscale image. User has to select the image. System will process the image by applying image processing steps. We applied a unique algorithm to detect images from the specified image. But edges of the image are not sharp in early stage of specification. So, we apply image segmentation on image to detect edges of the images. In this method we applied image segmentation to detect the species. Here we proposed image segmentation process and many image filtering techniques for accuracy.

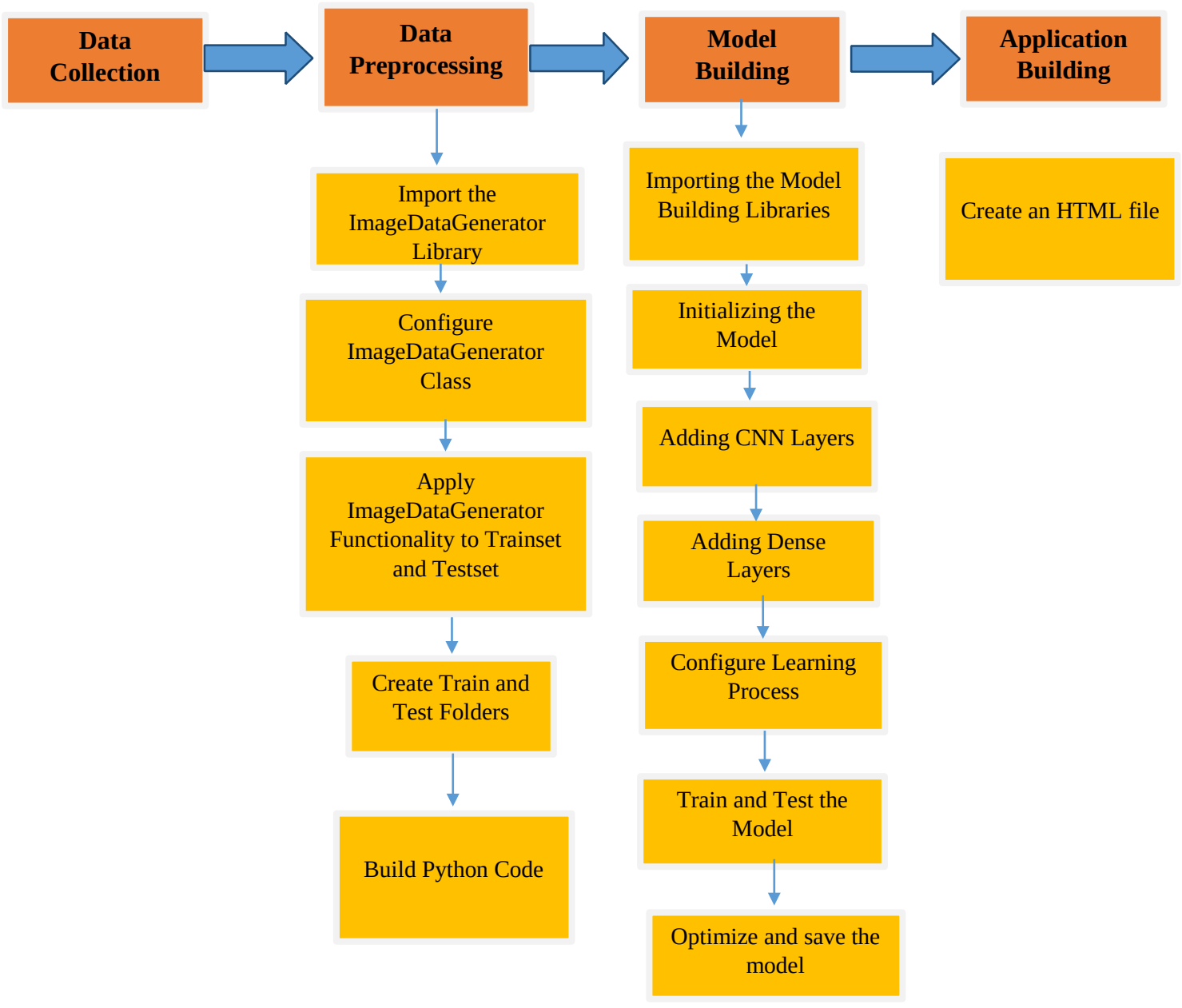
In this firstly we train the machine with some of the specified images which predict that figure is of those species or not then the user can use this by giving image data so it can predict flora and fauna image is present or not.

3. THEORETICAL ANALYSIS

It is important to detect species as early as possible. Manual detection of a species cell is a tiresome task and involves human error, and hence computer-aided mechanisms are applied to obtain better results as compared with manual detection systems. In **deep learning**, this is generally done by extracting features through a convolutional neural network (**CNN**) and then classifying using a fully connected network.

We have trained a convolutional neural network and obtained a prediction accuracy of up to 92%. CNN is a modified variety of deep neural net which depends upon the correlation of neighbouring pixels. It uses randomly defined patches for input at the start, and modifies them in the training process. Once training is done, the network uses these modified patches to predict and validate the result in the testing and validation process. Convolutional neural networks have achieved success in the image classification problem, as the defined nature of CNN matches the data point distribution in the image. As a result, many image processing tasks adapt CNN for automatic feature extraction.

3.1 Block Diagram



3.2 Software Designing

- Jupyter Notebook Environment
- Spyder
- Deep Learning Algorithms
- Python (Sequential, Dense, Conv2D, MaxPool2D, Flatten)
- HTML
- Flask

We developed this digital naturalist prediction by using the Python language, which is a high-level programming language along with Deep Learning Algorithm such as CNN. For coding we used the Jupyter Notebook of Anaconda distributions and Spyder, an integrated scientific programming in python language. Flask is used as a user interface for the prediction. Hypertext Markup Language (**HTML**) is the standard markup language for documents designed to be displayed in a web browser.

4. Experimental Investigation

In our project, we have used the Digital Naturalist Dataset. This dataset contains two folders: test set and training set. In test set folder, we have three categories called dove, peacock and robin, where, dove has the images having dove bird, peacock has the images having peacock bird, robin has the images having robin bird. Similarly, in the training set folder. Having 309 images belonging to 3 classes and 134 images belonging to 3 classes.

Importing Libraries

In [1]:
```
from tensorflow.keras.models import Sequential #for initializing
from tensorflow.keras.layers import Dense #adding layers
from tensorflow.keras.layers import Conv2D # adding convolution layer
from tensorflow.keras.layers import MaxPooling2D # max pooling
from tensorflow.keras.layers import Flatten
```

Initalizing the model

In [2]:
```
model=Sequential()
```

Adding CNN Layers:

Adding Convolution layer

In [3]:
```
model.add(Conv2D(32,3,3,input_shape=(64,64,3),activation='relu'))
#1st parameter in conv2D = no. of Feature detectors
#2nd &3rd parameter = size of feat. Detect.
#4th parameter = Expected input image shape
#5th parameter =Activation fnctn.
```

Adding Max Pooling layer

In [4]:
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

Adding Flatten Layer

In [5]:
```
model.add(Flatten())#converts ndimension to 1 Dimension
```

In [6]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 31, 31, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 7200)              0
=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

Adding Dense Layers

In [7]:
```
model.add(Dense(units=128,activation='relu',kernel_initializer='random_uniform'))
```

In [8]:
```
model.add(Dense(units=3,activation='softmax',kernel_initializer='random_uniform'))
```

In [9]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 31, 31, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 7200)              0
_________________________________________________________________
dense (Dense)                (None, 128)               460928
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 387
=================================================================
Total params: 461,611
Trainable params: 461,611
Non-trainable params: 0
```

Configuring the Learning Process

In [10]:
```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

Train and Test Model

In [11]:
```
from keras.preprocessing.image import ImageDataGenerator
```

```
Using TensorFlow backend.
```

In [12]:
```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_fl
ip=True)
test_datagen= ImageDataGenerator(rescale=1./255)
```

In [13]:
```
x_train=train_datagen.flow_from_directory(r'C:\Users\salon\Internship\dataset4\dataset4\trai
ning_set',target_size=(64,64),batch_size=32,class_mode='categorical')
x_test= test_datagen.flow_from_directory(r'C:\Users\salon\Internship\dataset4\dataset4\test_
set',target_size=(64,64),batch_size=32,class_mode='categorical')
#more than two categories class_mode='categorical'
```

```
Found 309 images belonging to 3 classes.
Found 134 images belonging to 3 classes.
```

In [14]:
```
print(x_train.class_indices)
```

```
{'dove': 0, 'peacock': 1, 'robin': 2}
```

In [15]:
```
model.fit_generator(x_train,steps_per_epoch=309,epochs=1,validation_data=x_test,validation_s
teps=134)
```

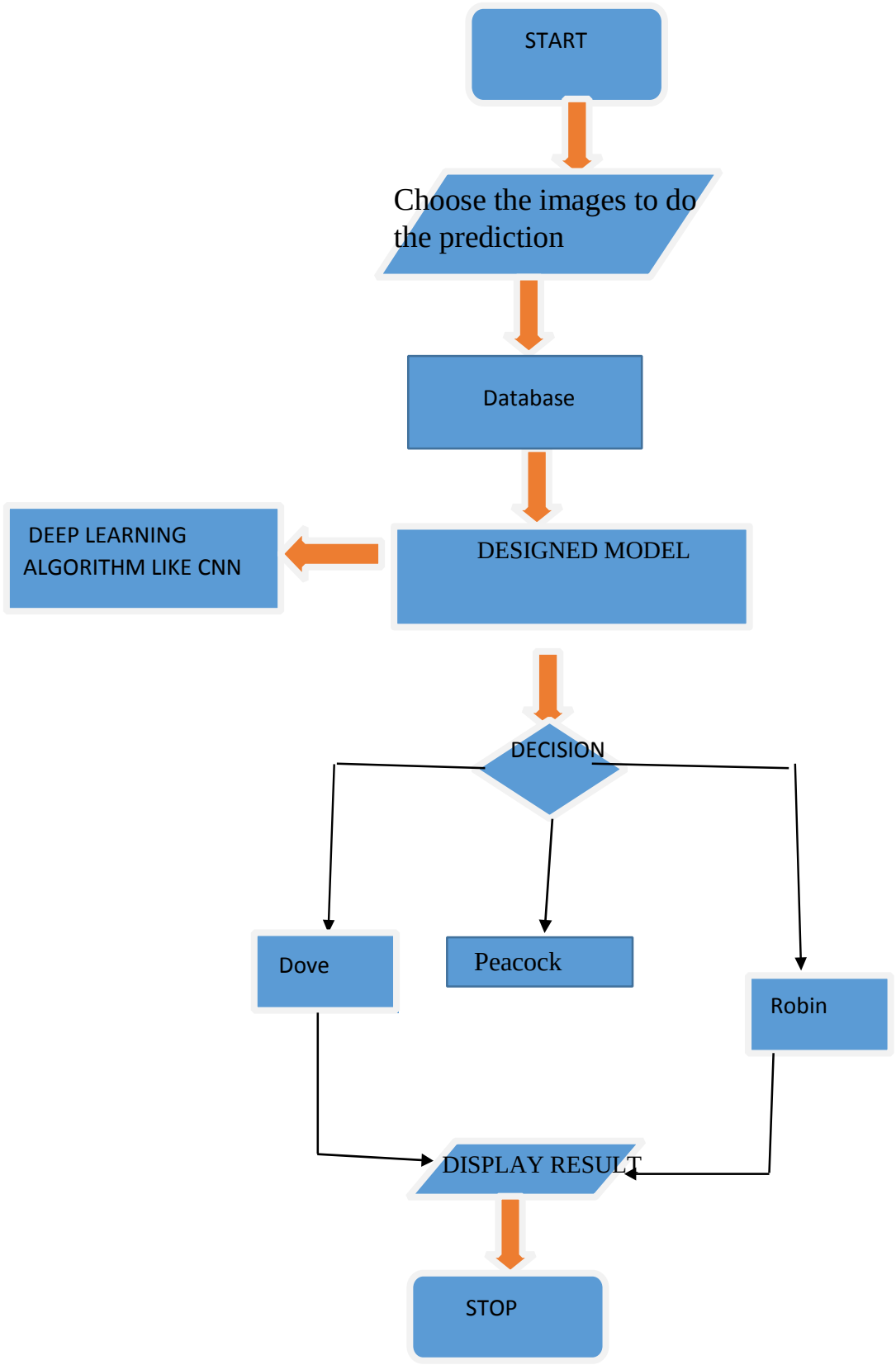
```
309/309 [==============================] - 233s 755ms/step - loss: 0.2583 - accuracy: 0.9118
- val_loss: 0.0674 - val_accuracy: 0.9776
```

Out[15]:
```
<tensorflow.python.keras.callbacks.History at 0x29ab2807648>
```

Save the model

In [16]:
```
model.save('testmodel6.h5')
```

5. FLOWCHART:


START
Choose the images to do the prediction
Database
DESIGNED MODEL
DEEP LEARNING ALGORITHM LIKE CNN
DECISION
Dove
Peacock
Robin
DISPLAY RESULT
STOP

6. RESULT

In this paper, the CNN algorithm is used to predict its performance. The results show that 92% accuracy.

Snapshots:

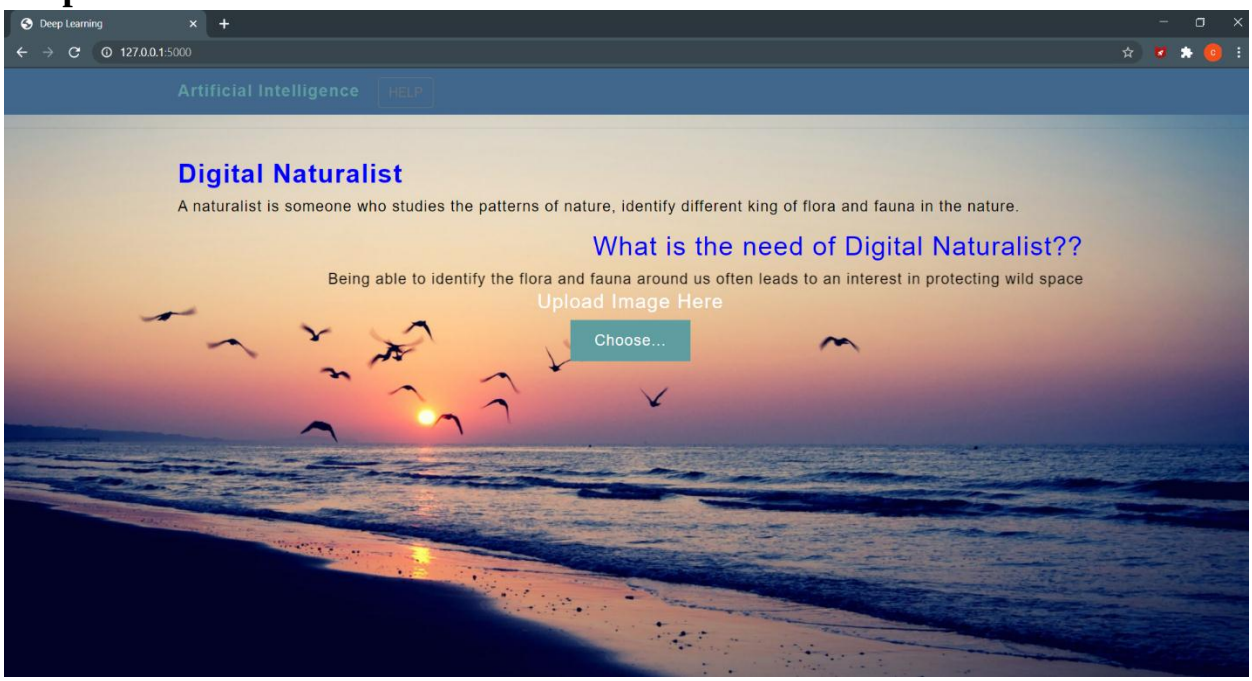


Fig:1-Home Page

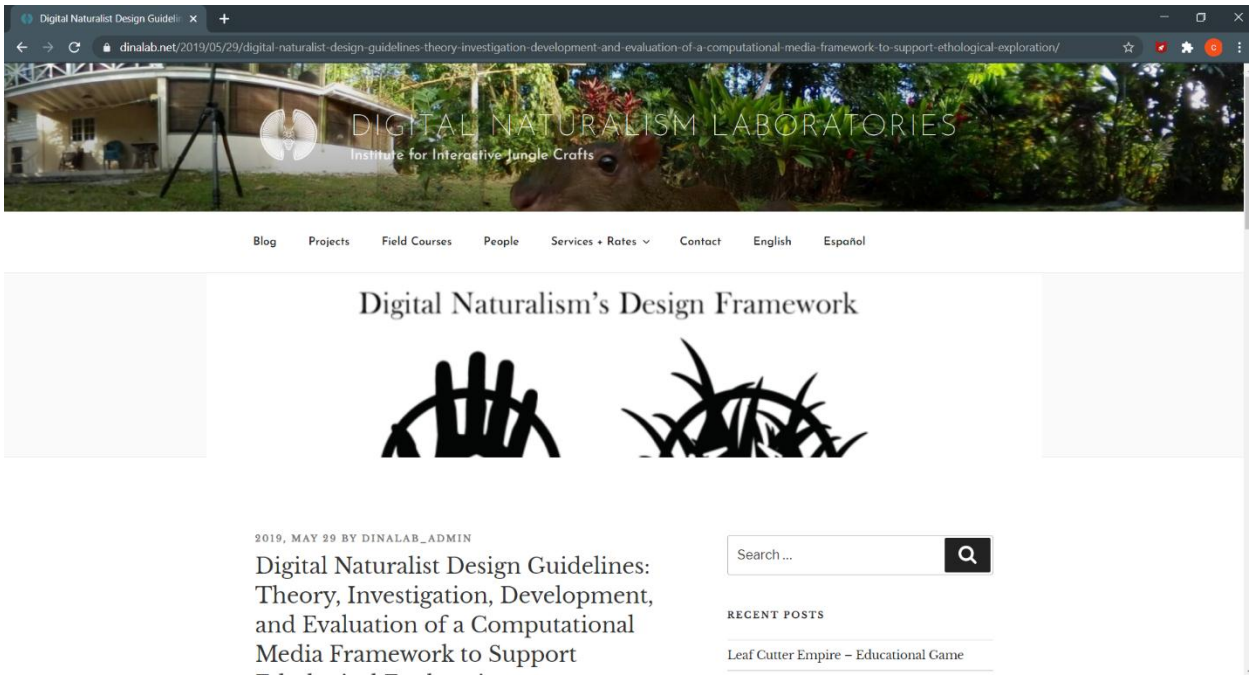


Fig:2-when you clicked on help it will redirect to https://www.dinalab.net/2019/05/29/digital-naturalist-design-guidelines-theory-investigation-development-and-evaluation-of-a-computational-media-framework-to-support-ethological-exploration/ this website.

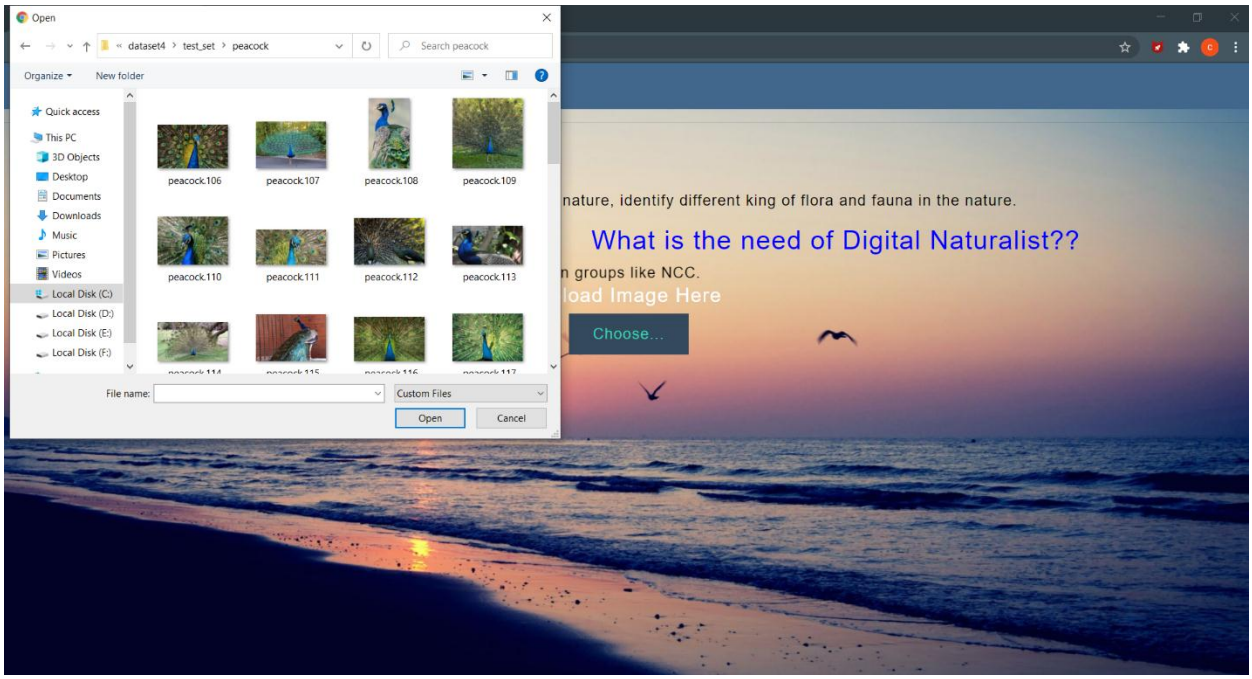


Fig:3-when you pressed choose, it will ask you to choose a file from the localhost.

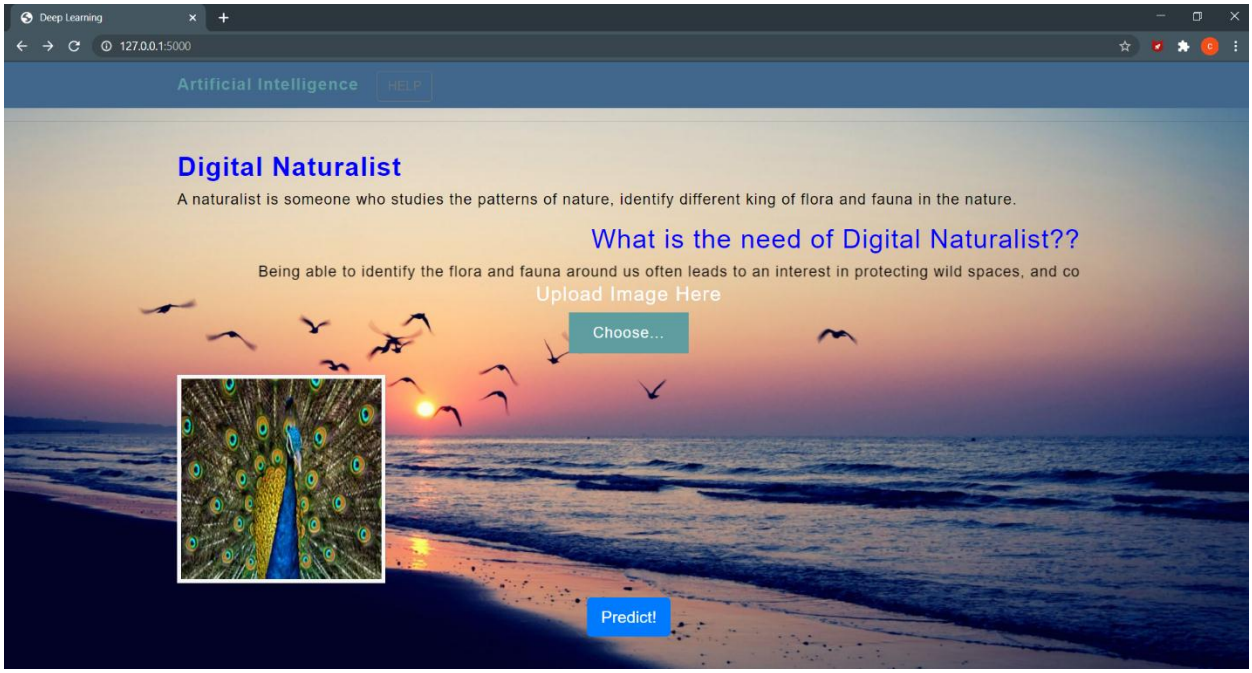


Fig:4-Selected image will be displayed

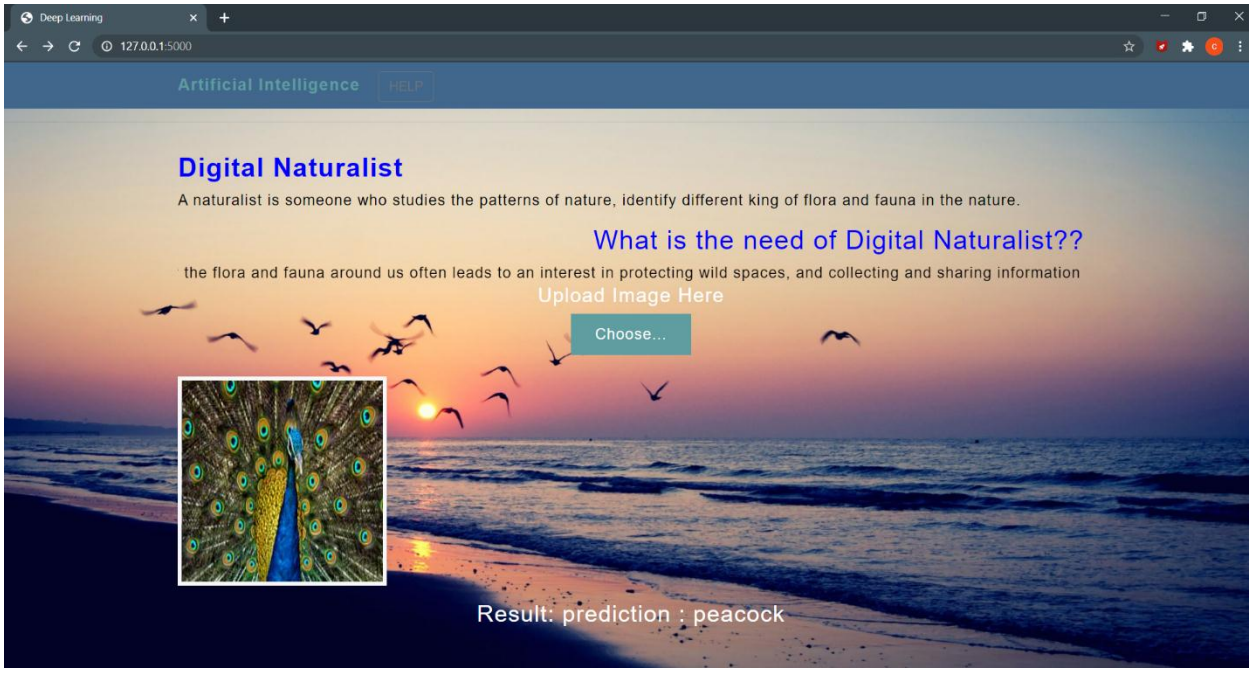


Fig:5-click on **predict** to see the output. If the selected picture is that of peacock it displays “peacock”.

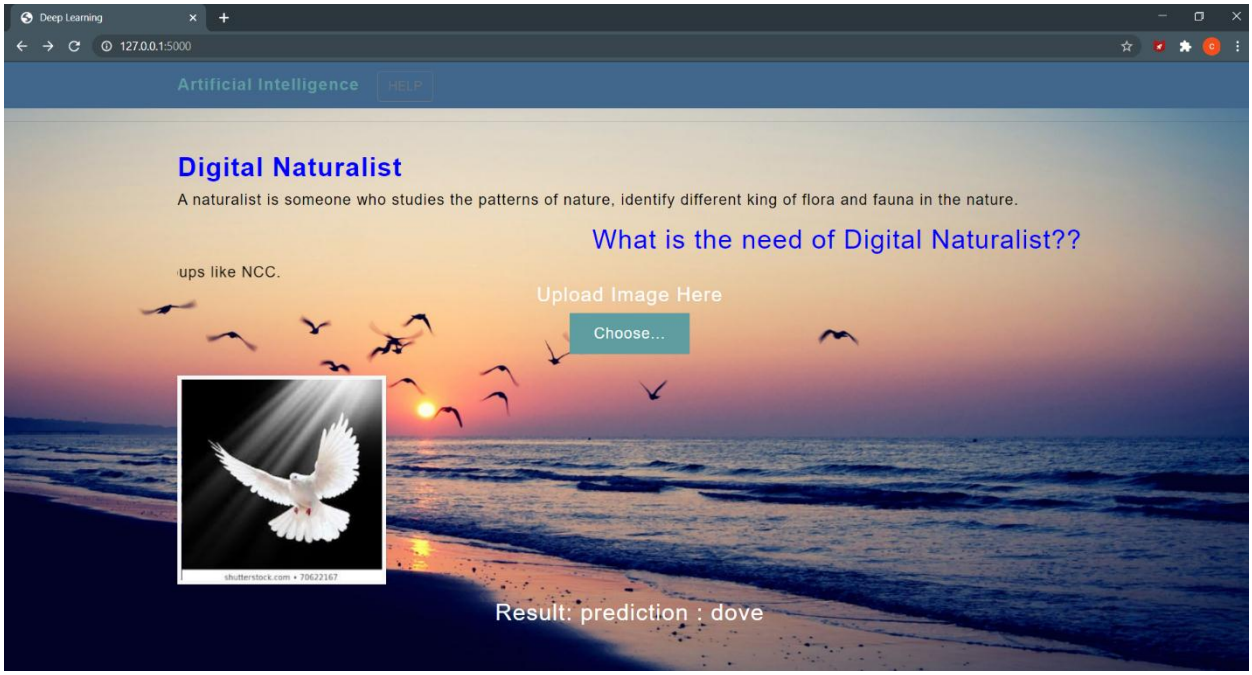


Fig:6-click on **predict** to see the output. If the selected picture is that of dove it displays “dove”.

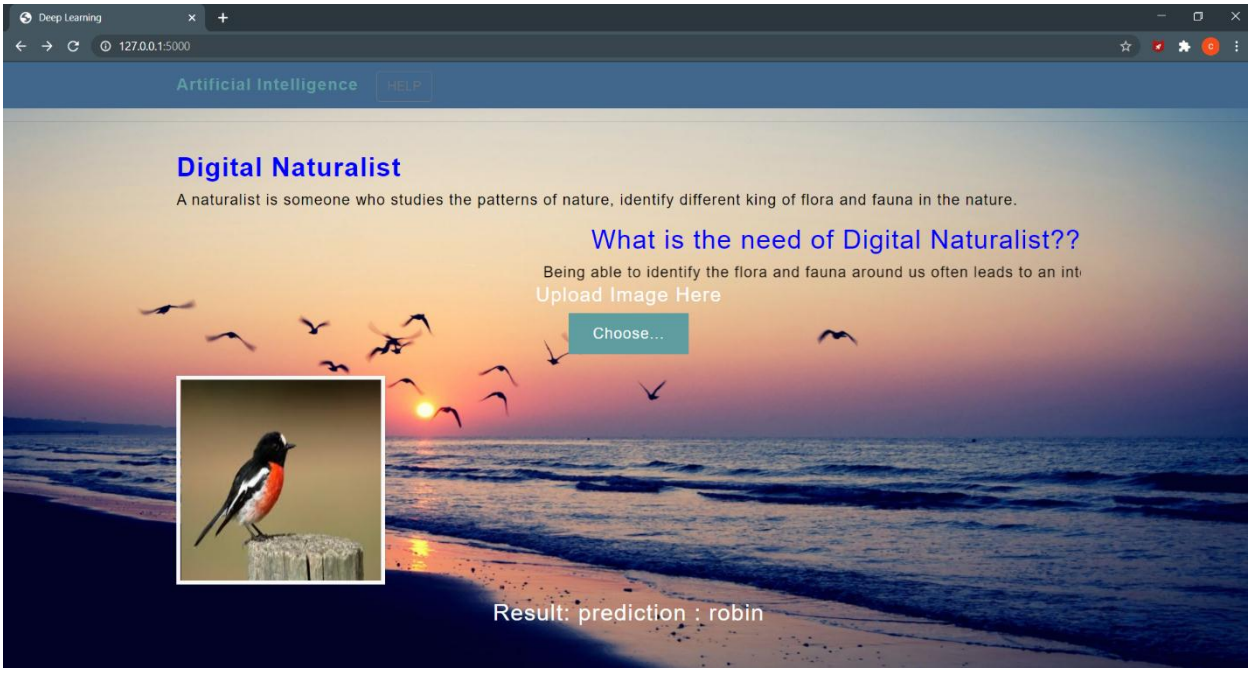


Fig:7-click on **predict** to see the output. If the selected picture is that of robin it displays “robin”.

7. ADVANTAGES AND DISADVANTAGES

Advantages:

1. Digital Naturalist detection is easy to implement and understand.
2. It operate in real-time due to low time complexity.
3. It is applicable in training and test-time
4. Deliver invariance with respect to the lesion position scale, and rotation.

Disadvantages:

1. It produces correlated images.
2. It easily generates anatomically incorrect examples.

8. APPLICATIONS:

1. "Digital Naturalist Detection using Convolutional Neural Networks" simplifies the management process of identifying and feature down flora and fauna data by deploying a web interface to the users.
2. Fast processing and immediate results with high security.
3. Minimizing human effort and cost-efficient databases.
4. Navigation through the site is easy.

9. CONCLUSION:

This projects consists of the details about the model which was used for the detection of digital naturalists using the species images from the wild life and the species with flora part and with faun part will be displayed as well. From the resultant graphs, it is proven that the accuracy of the model has reached good level, if it is deployed in the real-time scenario then it will help many people in distinguishing between both without wasting the money on various machines. If the image is confirmed by the model, then the person can know the feature of the species. It can be the best way of practice for people to save money. As we know that the data plays a crucial role in every deep learning model, if the data is more specific and accurate about the species then that can help in reaching greater accuracy with better results in real-time applications.

10. FUTURE SCOPE:

There is a wide scope for future implementation of "Digital Naturalists Detection using Convolutional Neural Networks" towards an interesting experience of modern

Technologies. Digital Platform is ‘one stop shops’ for all kinds of Naturalists to serve the domestic and international users at any time, any moment and anywhere in any parts of the world. Not being sticky to make packages within India only, it can be global - a “global platform” through a comprehensive. In present days, modern technologies have made the identification more pleasure comprising speed with comfort. So, people are not willing to be bound within only a small geographical area, so there is place to make them experience the taste of “Global Platform”. It can be enhanced into a Mobile Application. And also in future we can create an Artificial Intelligence Deep Neural Network Model for the evaluation for all other kind of diseases and even we develop in such a way that all the small kind of identification can be done easily.

11. BIBLIOGRAPHY

1. Simmel, G.: The Metropolis and Mental Life. Blackwell Publishing, Oxford (2004)

2. Frisby, D., Simmel, G.: Key Sociologists, 2nd edn. Routledge, London and New York (2002)

3. Marcuse, H.: One Dimensional Man: Studies in the Ideology of Advanced Industrial Society.

Routledge & K. Paul, London (1964)

4. Feenberg, A.: Transforming Technology: A Critical Theory Revisited. Oxford University Press,

New York (2002)

5. Feenberg, A.: Between Reason and Experience: Essays in Technology and Modernity. MIT

Press, Cambridge, MA (2010)

6. Feenberg, A.: Critical Theory of Technology. Oxford University Press, New York (1991)

7. Kahn, P.H.: Technological Nature: Adaptation and the Future of Human Life. MIT Press,

Cambridge, MA (2011)

8. Kahn Jr., P.H., Severson, R.J., Ruckert, J.H.: The human relation with nature and technological

nature. Curr. Dir. Psychol. Sci. 18(1), 37–42 (2009)

9. Heidegger, M.: The Question Concerning Technology, and Other Essays. Harper & Row, New

York (1977)

10. Heidegger, M.: Only a god can save us: the Spiegel interview (1966). In: Sheehan, T. (ed.)

Heidegger: The Man and the Thinker, pp. 45–68. Routledge, London and New York (2017)

11. Dunne, A., Raby, F.: Critical design FAQ [Online].
http://dunneandraby.co.uk/content/bydandr/

13/0

12. Gonsher, L.: Beyond Design Thinking: An Incomplete Design Taxonomy [Online].
http://

www.cd-cf.org/articles/beyond-design-thinking/

13. Dunne, A., Raby, F.: Designs for an overpopulated planet: Foragers (2009) [Online].
http://

dunneandraby.co.uk/content/projects/510/0

14. Edwards, E.: Designing contextually relevant digital interpretation for a public garden,

HighWire Centre for Doctoral Training, Dept. Computing and Communications, Lancaster

University, UK (2019)

15. Borgmann, A.: Technology and the Character of Contemporary Life. University of Chicago

Press, Chicago (1984)

16. Borgmann, A.: The moral complexion of consumption. J. Consum. Res. 26(4), 418–422 (2000)

17. Borgmann, A.: Orientation in technology. Philos. Today 16(2), 135–147 (1972)

18. Borgmann, A.: Reality and technology. Camb. J. Econ. 34(1), 27–35 (2010)

19. Edwards, J.C.: The thinging of the thing: the ethic of conditionality in Heidegger's later work.

A Companion to Heidegger, pp. 456–467 (2005)

20. Heidegger, M.: Building dwelling thinking. Poetry, Language, Thought, vol. 154 (1971)

APPENDIX

HTML FILE:

```
<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>Deep Learning</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
    <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}">
<style>
.moveright {
    text-align:right;
            }
            .moveleft {
    text-align:left;
            }
.button{
background-color: #4CAF50;
color:white;
}
div.ex1{

            padding:25px;
            width:30px;
            background-color:yellow
            }
div.ex2{
    background-color:rgb(0,191,255);
    color:rgb(162,0,175);
            padding:35px;
            }
```

```
.bg-dark {
                background-color: #42678c!important;
        }
#result {
                color: #0a1c4ed1;
                }

 </style>
 <script>
 function pageRedirect(){
 window.location.href="https://www.dinalab.net/2019/05/29/digital-naturalist-design-guidelines-theory-investigation-development-and-evaluation-of-a-computational-media-framework-to-support-ethological-exploration/"
 }
 </script>
</head>

<body>
   <div style="background-image: url("https://jooinn.com/images/birds-flying-3.jpg">

      <nav class="navbar navbar-dark bg-dark">
      <div class="container">
                  <div>
         <a class="navbar-brand" href="#" style="color:rgb(95,158,160)"><strong>Deep Learning</strong></a>
         <button class="btn btn-outline-secondary my-2 my-sm-0" type="button" onclick="pageRedirect()">HELP</button>
```

```
        </div>
                    </nav>
    <hr style="border:230px solid red,"/>
    <div class="container">

        <div id="content" style="margin-top:2em">{% block content %}{% endblock %}</div>
    </div>

</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

</html>
```

app.py file

```
from __future__ import division,
print_function
# coding=utf-8
import sys
import os
import glob
import numpy as np
from keras.preprocessing import image

from keras.applications.imagenet_utils import
preprocess_input, decode_predictions

from keras.models import load_model
```

```
from keras import backend
import tensorflow as tf

global graph
tf.compat.v1.disable_eager_execution()
graph=tf.compat.v1.get_default_graph()

#global graph
#graph = tf.get_default_graph()



from skimage.transform import resize

# Flask utils
from flask import Flask, redirect, url_for,
request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

# Define a flask app
app = Flask(__name__)

# Model saved with Keras model.save()

model =
tf.keras.models.load_model("testmodel6.h5")
      # Necessary
# print('Model loaded. Start serving...')
```

```
# You can also use pretrained model from
Keras
# Check https://keras.io/applications/
#from keras.applications.resnet50 import
ResNet50
#model = ResNet50(weights='imagenet')
#model.save('')
print('Model loaded: Check
http://127.0.0.1:5000/')

@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)
        img = image.load_img(file_path,
target_size=(64, 64))
```

```
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)

        with graph.as_default():
            preds = model.predict_classes(x)

        #text = "prediction : "+[preds[0]]

        index= ['dove','peacock','robin']
        text = "prediction : "+index[preds[0]]
              # ImageNet Decode

 return text

if __name__ == '__main__':
    app.run(debug=False,threaded = False)
```