

Image-based species recognition system using Deep learning Neural networks

Abhijit Rekhi *Computer Engineering Shri Chhatrapati Shivaji Maharaj College Of Engineering Ahmednagar, India*

Geetika Purohit *Computer Engineering Shri Chhatrapati Shivaji Maharaj College Of Engineering Ahmednagar, India*

Yash Kamble *Computer Engineering Shri Chhatrapati Shivaji Maharaj College Of Engineering Ahmednagar, India*

Corresponding Author: Abhijit Rekhi

Abstract

This is the age of Ai and augmenting humans in wildlife is limited. We bring up the idea of species recognition for better augmentation, creating better interactions between people and wildlife and educating citizens to nurture these bonds, better know their characteristics, and help live in mutual harmony. For this, we use Ai, ML, Deep learning and build a model and webapp based on Django that would recognize the species, show us its attributes and likenesses. We recognize the present developments in the domain and analyze them for our research to better aid our current ideas.

Keywords: *Artificial intelligence-Machine Learning CNN-Deep Learning*

Date of Submission: 15-05-2022

Date of acceptance: 30-05-2022

I. INTRODUCTION

There are an estimated 8.7 million wildlife species spread around the globe. Some of which are common to us and we see every day while some are unknown to us since either they are far-fetched, or have gone extinct. Humans are curious and we always quest the existence of anything we find interesting. So in this fast-paced world, it is difficult to go out in the wild and quickly become familiar with those dwelling there. So it's better to always go equipped with a handful of knowledge that's right there on your palm - of course, you would need some net!. That's why we came up with the idea to build a web app that would help you in recognizing wildlife species - using the power of machine learning, that you encounter during your adventure and get familiar with them.

Problem Definition:

To build an Image-based wildlife recognition web app using Django and Machine learning

Aim and Objective :

The project is aimed to recognize animal species by using deep CNN. Since there are a large number of different animals, manually identifying them can be a difficult task. In this phase of work, the limitation of using data mining is researched and a new method is proposed which employs CNN called Lightweight Machine Convolution Network for Animal Recognition. It will have both good accuracy and minimal computational cost.

Proposed System:

The proposed system is developed on CNN which helps in identifying animal features and using the Django and TensorFlow helps to achieve our recognizer to identify any species.

II. LITERATURE SURVEY

Guobin Chen et al.[1]	Proposed a deep CNN-based species recognition algorithm that was specifically developed to detect moving animals or wildlife which is in motion – and detect it accurately
Deepthi K et al [2]	Proposed animal species recognition using deep learning and developed an app named Imagenet of animals (IoA).
Marco Willi et. al[3]	Mentioned citizen science projects and the effective use of citizen scientists' accurate classifications which help reduce the time for large datasets based on ecology and animal monitoring.
Margaret Kosmala et al[4].	Proposed a Deep Neural network (DNN) model that would automatically detect, identify, count, and describe wildlife species; specifically animals
Kaitlyn M. Gaynor et al [5]	They trained a CNN to classify 20 African wildlife species with an overall accuracy of 87.5% using a dataset that contained reportedly 111,467 images. The authors demonstrated the application of a gradient weighted class-activation-mapping (Grad-CAM) procedure which was used to extract the most salient pixels in the final convolution layer
Margarita Favorskaya et. al [6]	Presented the system on animal species recognition in the wildlife based on muzzle and shape features using joint CNN. Monitoring the animal behavior in the forest supposes reliable techniques for animal recognition visual data captured by camera traps.
L. G. C. Vichakshan et. al [7]	Build Internet of Things based animal classification systems using convolutional neural networks. They use numerous technologies including CNN, joint algorithms using Zero Cross Rate (ZCR), MFCC, and Dynamic Time Warping (DTW), Hidden Markov Model (HMM), and using most common libraries and structures including LibROSA, TensorFlow, and Keras
S. Divya Meena et. al [8],	bring forward the idea of an efficient structure for animal breeds classification using semi supervised learning and multi-part convolutional neural networks (MP-CNN). These experiments were carried out in a single NVIDIA GeForce 940M Version 376.82 GPU-based laptop with Intel(R) Core (TM) i5-5200U CPU @ 2.20GHz processor and 8GB RAM

III. SOFTWARE REQUIREMENT SPECIFICATION

A. User cases and Characteristics

The user class will be end_user (common_citizen) and its methods will include clicking / capturing/taking snapshots (all used interchangeably !), additional attributes of the species, and also the text-to-speech output.

B. Dependencies

- We will require the Google Inception V4 network.
- We will require a cloud server to host the model over it.

C. Functional Requirements :

Functional Requirement	Description
FR 1	UserUser will be able to take a snapshot of species which will be then uploaded over the server.
FR 2	UserUser will be able to take a snapshot of species which will be then uploaded over the server.
FR 3	Text-to-speech output for user convenience.
FR 4	Display of Dos and Don'ts for better interaction
FR 5	“Help the community service” to get useful information as well as Feedback from the user.

D.Non-functional Requirement :

NFR	DESCRIPTION
Performance	Expected time less than 1 minute for species identification
Safety	We provide an option to display “No match found” to avoid system crashes and provide safe exit
Security	The critical dataset must be preserved by restricting and limiting user interactions to it.
Portability	We propose a mobile app to enhance portability
Usability T	The elegant and Beautiful UI ensures rich UX which is achieved by Flutter
Scalability	The system will scale up by using large data sets like the Serengeti project.
Flexibility	Users will be able to add more local information which is critical for the diversification of data.
Maintainability	Addition of user data needs to be scrutinized and segregated for future re-use and to draw helpful inferences.

E.System Requirement:

1. Database Requirements:

- PostgreSQL: PostgreSQL is an open-source descendant of this original Berkeley code. It supports a large part of the SQL standard and offers many modern features:
 - A. complex queries
 - B. foreign keys
 - C. trigger
 - D. updatable views
 - E. transactional integrity
 - F. multi-version concurrency control

2. Software Requirements:

- Visual Studio Code
- Github
- Keras models
- Tensorflow 2.6.1
- Django 4.x

- Python 3.10
- Matplotlib, NumPy, Pandas
- Microsoft Azure
- Bootstrap v5.x
- HTML
- Windows 10 / 11

3. Hardware Requirements:

- intel i7 or AMD ryzen 7 and later
- Nvidia GEFORCE GTX 1050 or later
- 16 GB DDR5 RAM of CPU or more
- 4 GB GDDR6 VRAM of GPU or more
- 256 GB SSD or more

4. System Requirement:

System Requirement: We design the complete model using Django and Tensorflow, Python is the key language in building the identification model. To compare the features of species we are going to develop a 4 layers Max pooling Convolutional neural network (CNN). To make the derivative easier we will be using some of the keras models.

III. SYSTEM DESIGN

1. Django:

Django is a high-level open-source Python language-based backend web framework that encourages rapid development with clean and pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. Some of its features are HTML support, database integration, built-in security features like csrf token, auth libraries, flexibility, and scalability. Django works on its own MVT architecture. (more on that in system architecture) It is also easy to use different keras models and TensorFlow libraries as Django provides quick and easy integration.

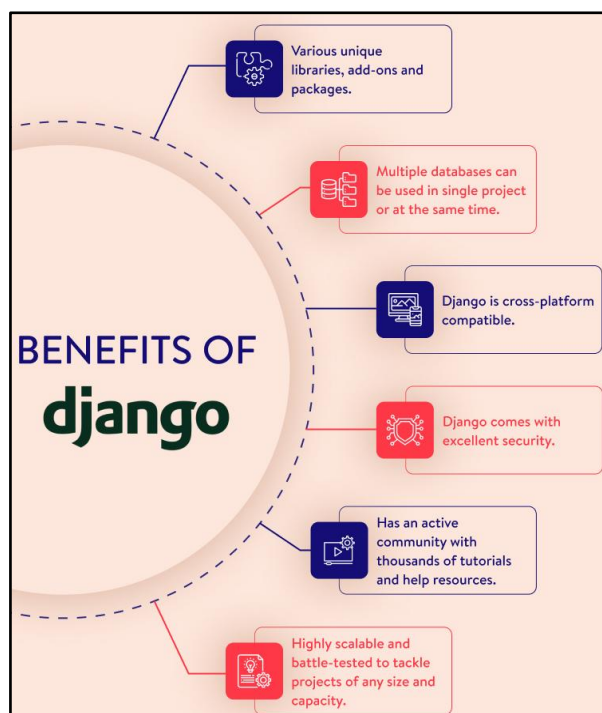


Fig 1. Benefits Of Django

2. Django Architecture:

Django has its own architecture known as Model, View, Template (MVT) which is slightly different from the other models available. (for eg. MVC) Here in Django database and its requirements are managed by the “**models**” module, all of the python code is written inside “**views**” which are basically Django’s engine and then this code is rendered in the browser using “**templates**” where all the HTML and Frontend code is controlled. Django’s architecture is almost flawless and seamlessly integrates the python code into HTML code using the “**jinja patterns**”

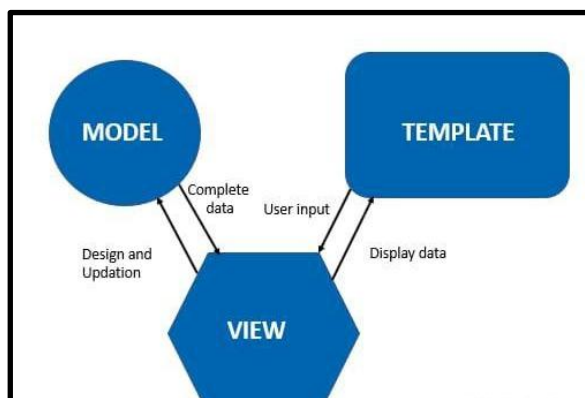


Fig 2 . Django Architecture

This whole process is controlled using urls modules. Yes Django got you covered. It has url modules where routing operations are done for your webapp. You create a html template you write the frontend and the working of it is written in views (#backend) then if user data is involved it is managed by models and this all to and fro is controlled using urls. Django provides a bunch of commands and all of those are executed by manage.py. It also provides a built-in admin role and everything is configured in settings.py file.

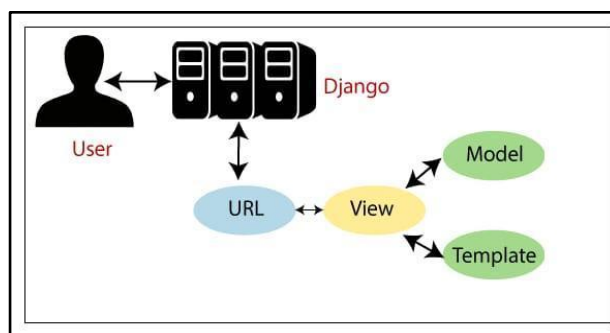


Fig 3 Working of Django

3. System Implementation:

For implementation purposes it was imperative to choose a good IDE and that’s why we went for VS code. It is powerful, supportspowerful, supports way too many languages and has the ability to integrate many robust and extensive frameworks. The language was of course Python 3.10 and the framework used is Django.

4. CNN:

CNN has broken the mold and ascended the throne to become the state-of-the-art computer vision technique. Among the different types of neural networks (others include Recurrent neural networks (RNN), Long short term memory (LSTM), Artificial neural networks (ANN), CNN's are the most popular due to their ubiquitous nature in recognizing different forms of images correctly. CNN works phenomenally well on computer vision tasks like image classification, object detection, image recognition, etc

There are various datasets that you can leverage for applying convolutional neural networks. Here are three popular datasets:

- MNIST
- CIFAR-10
- ImageNet

5. Keras :

Keras is a simple-to-use but powerful deep learning library for Python. Here, we'll build a simple Convolutional Neural Network (CNN) and train it to solve a real problem with Keras

Here are the steps for building your first CNN using Keras:

1. Set up your environment.
2. Install Keras.
3. Import libraries and modules.
4. Load image data from MNIST.
5. Preprocess input data for Keras.
6. Preprocess class labels for Keras.
7. Define model architecture.
8. Compile model.
9. Fit model on training data.
10. Evaluate model on test data.

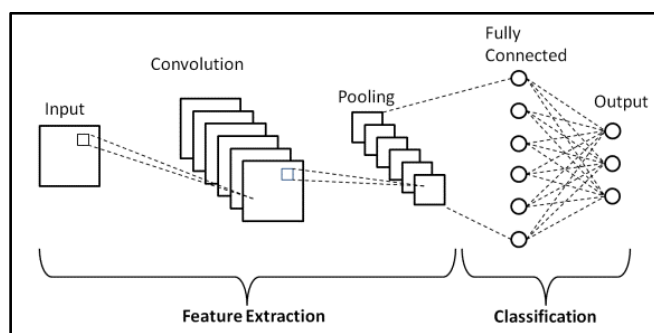


Fig 4: CNN Architecture

UML DIAGRAM For CNN

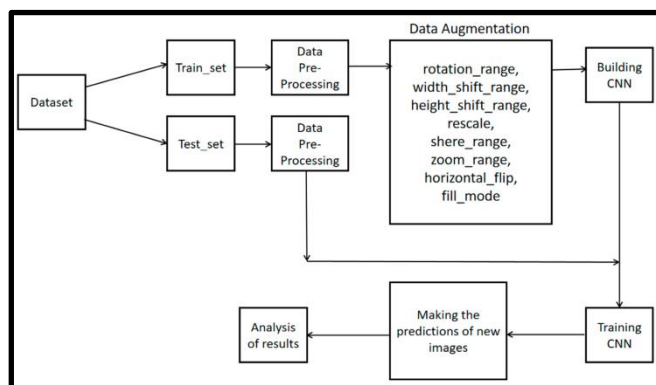


Fig 5 UML diagram for CNN

6. LSTM Architecture:

LSTMs deal with both Long Term Memory (LTM) and Short Term Memory (STM) and for making the calculations simple and effective it uses the concept of gates.

1. Forget Gate: LTM goes to forget gate and it forgets information that is not useful.
2. Learn Gate: Event (current input) and STM are combined together so that necessary information that we have recently learned from STM can be applied to the current input.
3. Remember Gate: LTM information that we haven't forgotten and STM and Event are combined together in Remember gate which works as updated LTM.
4. Use Gate: This gate also uses LTM, STM, and Event to predict the output of the current event which works as an updated STM.

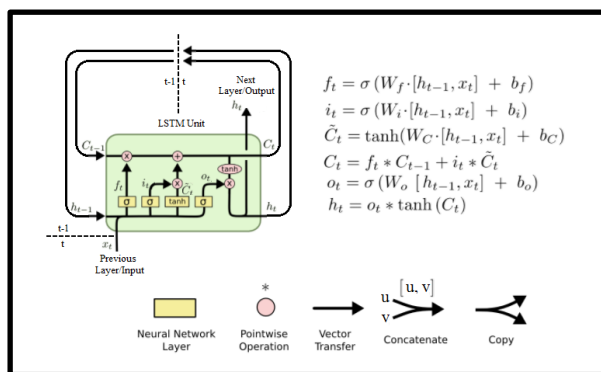


Fig 6. LSTM architecture

V. IMPLEMENTATION AND WORKING MODULES

A. Creating the app :

To begin with you need to understand the working of Django framework. Django considers the whole website as a project. Under the project there are several apps (for e.g. the homepage of a website) and in each app there are several modules (for e.g. the search button or submit button) with many functionalities. We can have the same module in two apps as they are portable.

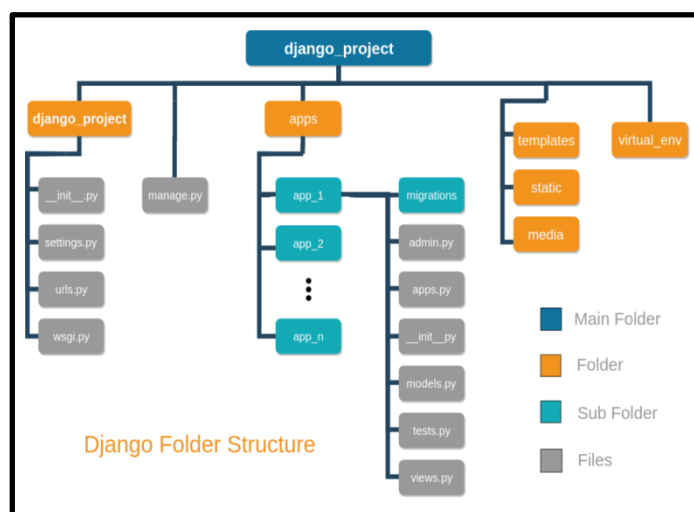


Fig. 7 Creating the App

B. Developing the modules :

Now in the project folder there is urls.py file, here we need to add the route to our home app. After that we create a new folder in the project which holds all of our website pages named as templates. Now we create base.html, index.html, about.html, feedback.html, profile.html, etc.

- base.html : Contains the basic look of our website which is constant throughout the project. We extend this into every html page.
- index.html : Contains the main function of our project. The classification of images based on machine learning.
- About.html : Contains details of the creators
- feedback.html : Contains the feedback form. There are many html pages and few are listed.

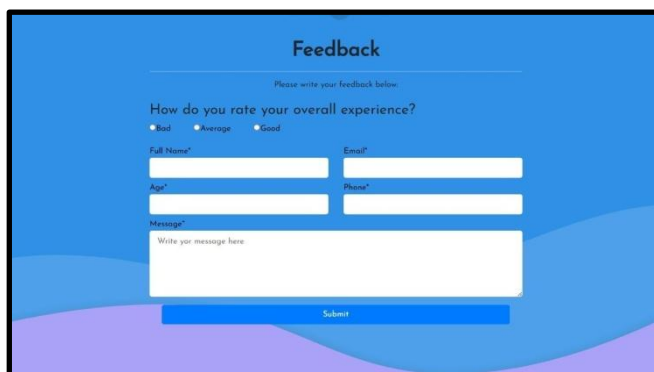


Fig 8 Feedback.html

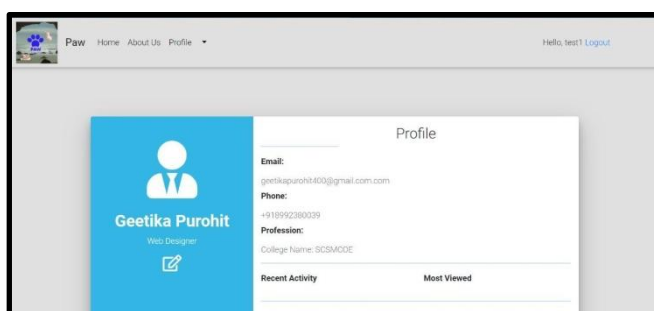


Fig 9 Profile.html

C .The BACKEND stuff :

To ensure that these pages work as intended we need to write the equivalent logic for its desired flow. This code logic is written in views.py file of that app. Views.py consists of all types of functions that are user-built to drive the action of these webpages. These functions then return the response by rendering it to html format. To handle the styling of pages which include inserting some pics, doing some coloring and font-styling or padding or making the pages interactive, we use static files directory where all this stuff is kept in required order. Images in img folder, font-styles and coloring in css files in css folder and interactive javascript files in js folders. These static files need to be configured settings files where we add the directory path to them as we do in templates case.

D. Database Modeling :

Before we deal with any data we must have admin powers and that's why let's create an admin by using : Python manage.py createsuperuser Enter the username and password you wish. The next step is to visit the django admin web page by giving /admin to our project site. Django has its own Authentication method and provides built-in admin support to each app of the project. Now django also provides a built-in database known as db.sqlite3 but it has very limited usage so is not suited for production purposes. That's why we chose PostgreSQL as it provides a lot better support to python based production. Be cautious while installing postgresql and don't forget to install the pgadmin suite for gui support.

E. Hosting On Azure :

The last step is to host the project on cloud to actually make it work like a real-life project. To do so we have chosen MS Azure cloud as it provides a Django app service feature that easily migrates all the data over to cloud and helps run the project on the go. One of the best parts of Azure is it also helps in PostgreSQL integration in very few steps. So hosting it on Azure gives the link of the site through which we can view the site anytime. The results of the project were quite pleasing. After due diligence and hard work the project was working as intended and was displaying almost cent per cent accurate results while recognizing the animal species. The vgg16 model of the keras was very much helpful and had accuracy very much identical to our own developed model. We used both the techniques of CNN and LSTM combinely to achieve this.

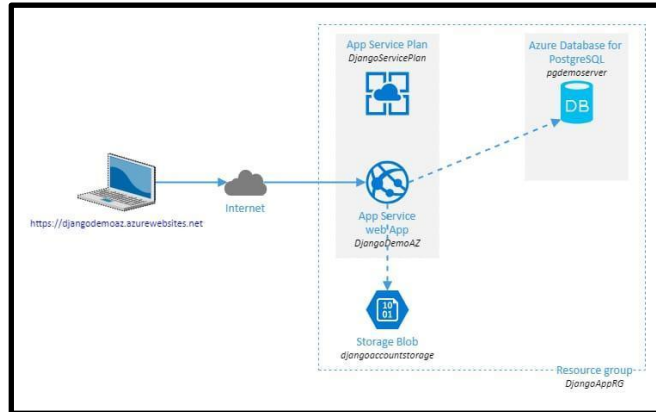
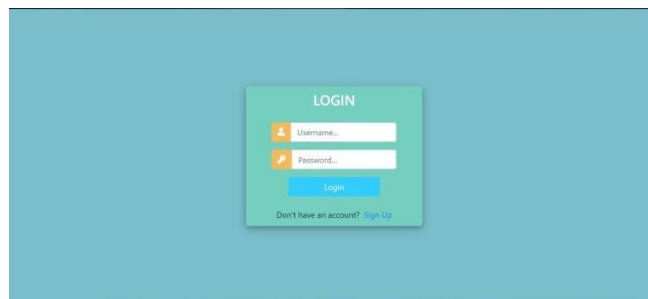


Fig 10 Django on Azure

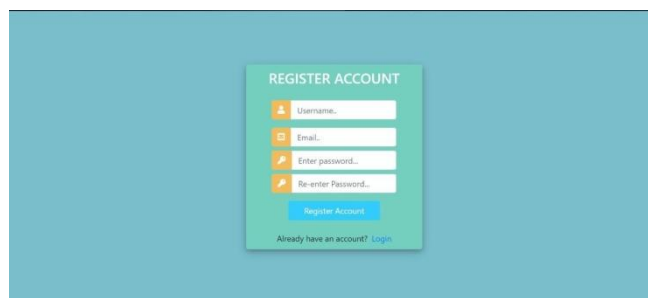
VI. RESULT

The results obtained are as discussed below:

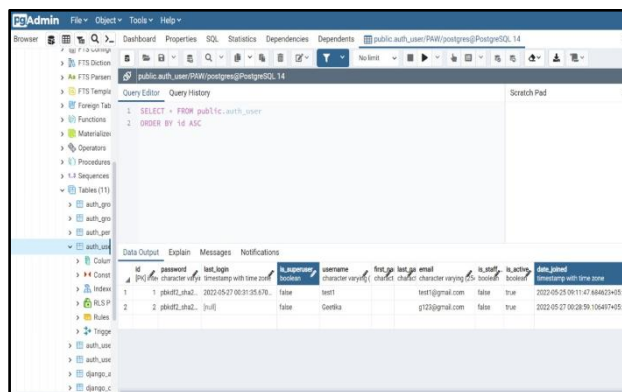
The results of the project were quite pleasing. After due diligence and hard work the project was working as intended and was displaying almost cent per cent accurate results while recognizing the animal species. The vgg16 model of the keras was very much helpful and had accuracy very much identical to our own developed model. We used both the techniques of CNN and LSTM combinely to achieve this.



Result 1. Login



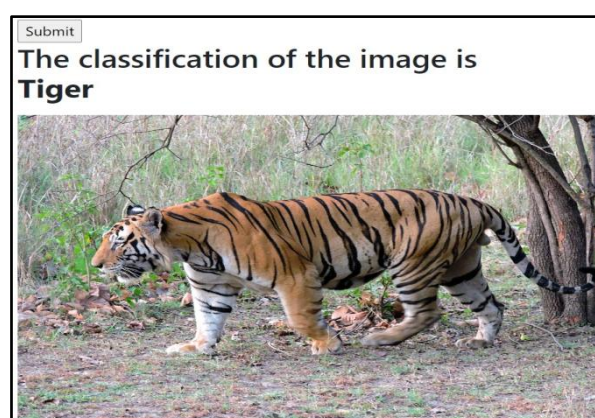
Result 2. Registration



Result 3. Integration of Azure and PostgreSQL



Test Case 1



Test Case 2



Test Case 3

VII. CONCLUSION

So in the epilogue, we firmly believe that these kinds of projects will serve humanity immensely through their potential of using new technologies yet delivering results that have grass root level touch. This was one of its kind journeys which ignited a lot of excitement among its developers. We are ready and excited for the new projects which will help us master what we have learnt. This journey is to be cherished as it involves not only the development of the project but also the shaping of a new career for us where developing these kinds of things is beautiful. We also tried to provide some security and authentication in the project by defining user creation and login. The whole project was made carefully, understanding each concept of the technologies used. The different facets involved in it and a lot of bugs, thanks to them!!! After all they were the ones who annoyed us every time but also helped us rejoice each time we conquer them. We look forward to hoping this helps you

too and you do interact with the wildlife species that you encounter happily. Raise the toast when this happens. Till then keep exploring!!!

.....Team PAW

REFERENCES

- [1]. Guobin Chen , Roland Kays, “DEEP CONVOLUTIONAL NEURAL NETWORK BASED SPECIES RECOGNITION FOR WILD ANIMAL MONITORING”, IEEE 2014
- [2]. DEEPTHI K, VIMALA N, “ANIMAL SPECIES RECOGNITION SYSTEM USING DEEP LEARNING”, International Research Journal of Engineering and Technology (IRJET), Volume: 08 Issue: 07 | July 2021
- [3]. Norouzzadeh, Margaret Kosmala, “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning”, PNAS | vol. 115 | no. 25, April 30, 2018
- [4]. Marco Willi, Ross T. Pitman, “Identifying animal species in camera trap images using deep learning and citizen science”, Methods in Ecology and Evolution, 20 September 2018
- [5]. Kaitlyn M.Gaynor, Zhongqi Miao, “Insights and approaches using deep learning to classify wildlife”, Nature Scientific reports, 31 May 2019
- [6]. Margarita Favorskaya, Andrey Pakhirkaa, “Animal species recognition in the wildlife based on muzzle and shape features using joint CNN ”, Elsevier Procedia Computer Science, 2019
- [7]. L. G. C. Vithakshana, W. G. D. M. Samankula, “IoT based animal classification system using convolutional neural network”, IEEE Smart Computing and Systems Engineering, 2020
- [8]. S. DIVYA MEENA L., AGILANDEESWARI, “An Efficient Framework for Animal Breeds Classification Using Semi-Supervised Learning and Multi-Part Convolutional Neural Network (MP-CNN)”, IEEE Access, October 30, 2019