Journal of Analysis and Computation (JAC)
(An International Peer Reviewed Journal), www.ijaconline.com, ISSN 0973-2861
Volume XVIII, Issue I, Jan-June 2024

REAL TIME MOBILE APPLICATION FOR WILDLIFE DETECTION

Harshul Vatsa[1], Himanshu Verma[2], Vishal Saxena[3], Pradeep Kumar Sharma[4], S. K. Dixit[5]
[3]E-mail: vishalsaxena.math@jecrc.ac.in

ABSTRACT:

The human beings massively depend on the natural assets and bounties for their survival but the tragedy at play is, we hardly opt to give a wee bit back to the nature. Well, to society's books, we thought of adding a genuine page of care and action for the animals. As the tech savvy generation, technology came to our first sight for showering any contribution to the good. Tracking and detecting animals has never been an easy task for any geography. This put our minds to hustle for a solution benefitting both human and the animals by making their unnecessary encounters lesser. So, through this work an application is developed that would be using the power of image recognition to identify the different wildlife animals all around us and to transmit the animal's location in real-time as well as maintain a database of their geographical presence.

Keywords - System Monitoring, Wildlife Detection, Wildlife Protection.

[1] INTRODUCTION

Very similar to how a human brain detects an animal with their eye's coordination, our application works on a Trained Model aided by a Camera. Discussing about the Trained Machine Learning Model, we have used hundreds of sample animal images viz. Tiger, Elephant, Cow, Dogs, etc. Once the model had learned all the features and specifications of each animal precisely, we deployed it, on an Android Linux Kernel for putting its practical capability at test [1, 2]. Needless to pen, the Android OS was already integrated with a working camera. To our joy, the whole Application tumed out to be working perfectly. We simply had to open the application and allow the necessary permissions for the Model to access Camera. Almost in no time, the Animal behind the Camera was classified accurately and the results were logged at the Phone's screen. All of this was made possible applying the Convolutional Neural Networks framework offered by Keras, a very handy and useful Machine learning Library in Python. The primary language used was Python- for machine learning part and Java-for the Android Integration and Deployment part [3]. The efficient

feature selection and iterative learning principles of CNN formed the pillar behind the fabulous accuracy of the Detection and Classification. Currently, it's in the form and shape of an android application but having this idea working and well proved for Linux Kemels, deploying similar models on Real Time Surveillance CCTVs, won't be much of a task.
Although animals too enjoy companionship, but they tend to reject any evasive interference by us. In the name of Modernization and Development, we have periodically plundered nature and the wild-life's peace [5]. Be it Expressways, Buildings, or Markets, wild-life has always been prone to home-less-ness. Adding to these, the modern Social Media Camera Spree too poses an alarming threat to Wild animal's privacy and peace. Contrary to this human life is also in sheer peril if it comes to an encounter with Carnivores like Tiger, Lion, Leopard, etc. So, it could be aptly concluded that not Human-Wildlife Interaction needs to be checked [6]. All of this will find a solution if the Human-Animal Interaction could be regulated and controlled. Our Application aims and proves to avoid Animal-Human interferences by automatically detecting wildlife presence and regulating the human presence accordingly. This will not only benefit Wildlife but Humans too. Even the government these days, is aiming to build green corridors for the wildlife to flourish amidst the ambitious development projects [7, 8]. Our application can be a game changer support for any upcoming wildlife projects. This way. Lives will be saved, peaceful co-existence will be established and nature will prevail.

[2] METHODOLOGY

Right after the inception of the idea, began the analysis part of the project. To make our model most accurate, we gathered ample amount of image data. To do so we searched various websites of central state governments as well as various articles on animals released by government, newspapers as well as NGOs and other private organizations to find how we can use image detection algorithm to detect wildlife animals and do something for the welfare of wildlife animals. The objectives of this work is to:

- Identify wildlife species among us and learn about them in real time.
- To gather location of various wildlife species.
- To send alert message when an animal is detected nearby

To accomplish the above said objectives, implementation of the work involves following steps

1. Gathering of Animal Image Data Set from various relevant sources.
2. Training the Machine Learning Model using libraries like Keras and other Tensor flow classes.
3. Exporting the Trained Model.
4. Developing the required Infrastructure in Android like Camera's Access and User Interface.
5. Deploying the exported Model on Android. Integrating with the User Interface and Real time testing.

The portal was tested upon completion against various test cases by going to different places and verifying with different kinds of animals. Considering the objectives and the basic implementation steps, algorithm of the work is defined as-‘

[3] ALGORITHM

1. **Gathering of data**: The data was mostly images of different animals like Cow, Dog, Elephant, Lion, etc. The data was varied widely to cover maximum breeds and types of individual species. Also, the numbers of images for a particular species were kept optimum so that the popular problem over fitting or under fitting could be avoided.

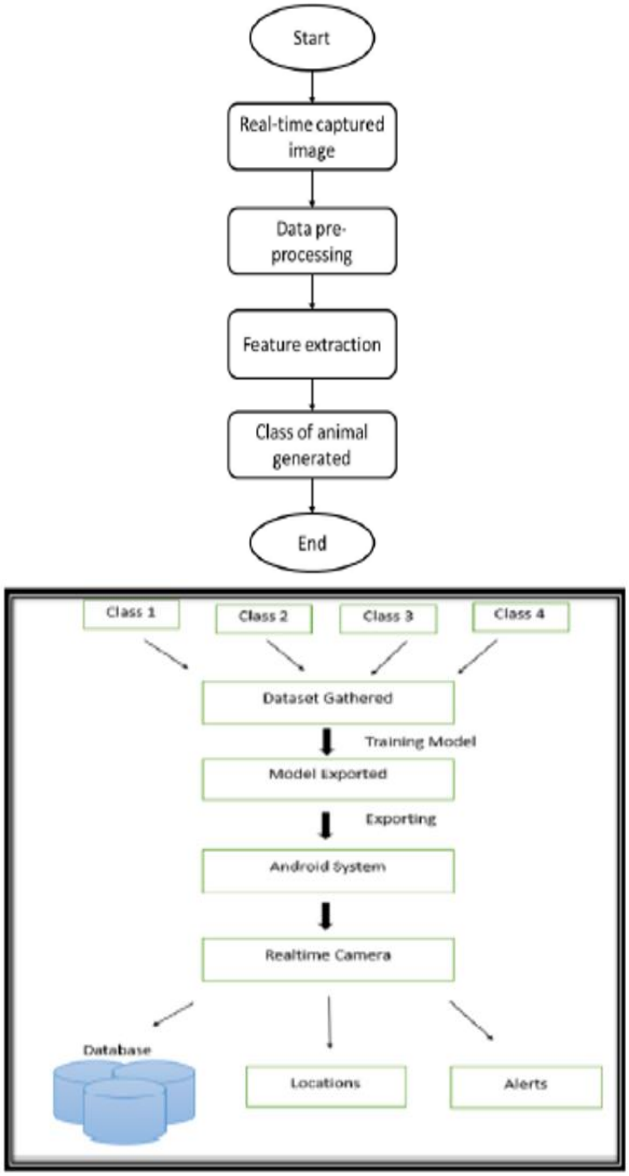
2. **Creating Machine Learning Model**: Google's Tensor flow was extensively used for the training and exporting purpose of Model. The Keras library helped us implement the Convolutional Neural Network concept for precise feature extraction and training of the aspired model [9, 10]. The exact training could be summarized in the following steps [4, 11].

1. Importing the required libraries in python.
2. Mounting the drive containing the training datasets over the Google Colab ide.
3. Generating the image data.
4. Flattening the multidimensional pixel matrix of the training images.
5. Extracting ample features.
6. Feeding the CNN and training the model in cycles.
7. Exporting the Model.

3. **Deploying the Model on Android System:** The Android application was made using Java over Android Studio. The User interface was made using XML and the necessary ML integration coding was done using Java. Necessary permissions like Camera's Access and Local file's access were sought in the application. Once the environment was ready we deployed the Trained Exported Model from Colab on our Android System. Subsequently the complete Application was built.

4. **Testing of data and detection of Animals:** Once the application was built we extensively tested it for random animals in real time. The application classified the animals accurately. The results are attached as testimony under the forthcoming headers.

5. **Exit:** Figure 1 and 2 represents the basic steps involved in the work and the methodology of the experiment is mentioned in the flow diagram respectively.



[4] RESULT AND DISCUSSION

The application has been designed to identify the animals using the mobile phone camera. Following section discusses about the dataset, experiment and results.

4.1 **Dataset:** It contains the images of animals collected from different sources; total no of images used for this work is 500 images with 6 classes

4.2 **Results:** The animal dataset is divided into training and testing set. To build the model, keras & tensor flow libraries are imported. After importing the libraries, image data is generated. Image data generator generates the image in different orientations. Images are converted into arrays to pass as an input to the machine learning model. To avoid the issues of high memory consumption and time taken to process the multi-dimensional arrays of images image flattening is used. This converts the multidimensional array of image in 1D

array. Using CNN, features are extracted, and the model is trained using training and testing dataset. The output of the trained model is shown below.

OUTPUTS:

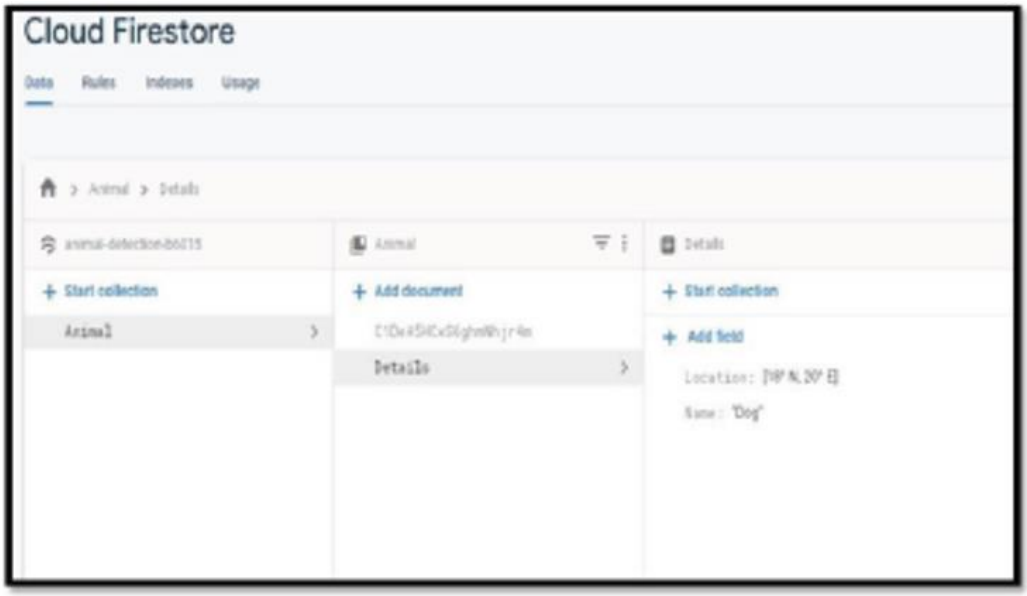


Fig 3 : Real time data is stored in cloud Firestore.

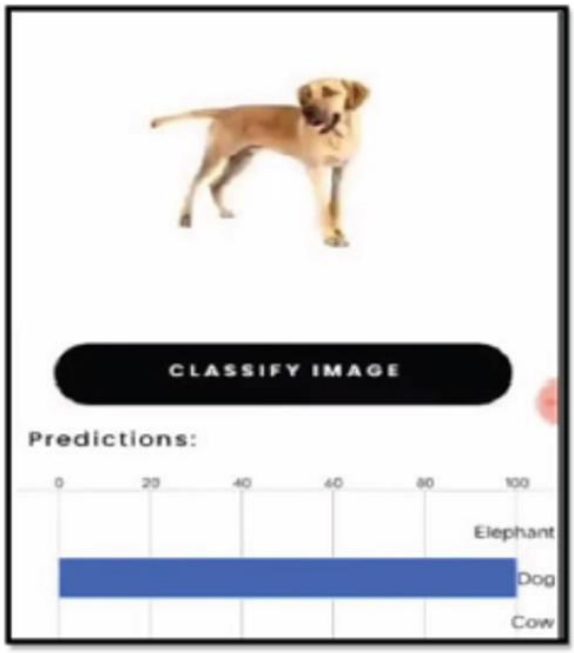


Fig 4: Feature extraction classify the class of images

[5] CONCLUSION

We were able to successfully implement the Real Time Mobile Application which will identify the class of animals. The system can accurately identify the images of the animals

and the mobile application can help the users to identify the animals promptly. The application can capture the real time image and locations and able to give insight about animals. The future scope involves improvement in animal detection in all types of weather as currently it can identify only in clear weather. In future our system will be capable of detecting animals in daytime, nighttime and in fog as none of the existing systems promised the detection of animals in foggy weather. Upon successful classification of animals our mobile applications send alert message.

REFERENCES

[1] A.C. Bartlett, A.A. Andales, M. Arabi, T.A. Bauder. A smartphone app to extend use of a cloud-based irrigation scheduling tool. Computers and Electronics in Agriculture Volume 111, February 2015, Pages 127- 130. ISSN 0168-1699.

[2] ALSAADI, E. M., & El Abbadi, N. K. (2019). Auto animal detection and classification among (fish, reptiles and amphibians categories) using Deep Learning. Journal of Advanced Research in Dynamical and Control Systems, 11(10-SPECIAL ISSUE), 726-736. https://doi.org/10.5373/jardcs/v11sp10/20192863

[3] B. Bruegge and A. H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, Second Edition.. Prentice Hall, 2003.

[4] Choc, D., Choi, E., & Kim, D. K. (2020). The real-time mobile application for classifying of endangered parrot species using the CNN models based on Transfer learning. Mobile Information Systems, 2020, 1-13. https://doi.org/10.1155/2020/1475164

[5] Francine Lalooss, Hengky Susanto, Chorng Hwa Chong. An Approach for Tracking Wildlife Using Wireless Sensor Networks:https://developer.android.com

[6] Antonio Javier Garcia -Sanchez, Wireless sensor network deployment for monitoring wildlife passages, ISSN 1424-8220

[7] Hu, M., & You. F. (2020). Research on animal image classification based on Transfer learning. Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering. https://doi.org/10.1145/3443467.3443849

[8] Larson, J. (n.d.). Assessing convolutional neural network animal classification models for practical applications in wildlife conservation. https://doi.org/10.31979/etd.ysr5-th9v

[9] N, J. (2020, July 29). How to classify animal images via a convolutional neural network. Hacker Noon. Retrieved October 7, 2022, from https://hackernoon.com/how-to-classify-animal-images-via-a-convolutional- neural-network-opl w3x3f

[10] Taheri, S., & Toygar, Ö. (2018). Animal classification using facial images with score-level fusion. IET Computer Vision, 12(5), 679-685. https://doi.org/10.1049/ict-cvi.2017.0079

[11] Zeng, P. (2021). Research on similar animal classification based on CNN algorithm. Journal of Physics: Conference Series, 2132(1), 012001. https://doi.org/10.1088/1742-6596/2132/1/01200.