

Automated Wild Animal's Identification in Forest using Machine Learning and Forest Fire Alert System

Akash Pandey

Department of Electronics and Communication.
R.V. College of Engineering.
Bengaluru, India.

Samvrit Kashyap

Department of Electronics and Communication.
R.V. College of Engineering.
Bengaluru, India.

Vishal Ladda

Department of Electronics and Communication.
R.V. College of Engineering.
Bengaluru, India.

Pratibha K

Assistant Professor
Department of ECE
R.V. College of Engineering. Bengaluru, India.

Abstract—The goal of the Forest Fire Detection System is to create a system that protects the forest by alerting forest officials by SMS as soon as a fire is seen and calling for help. IoT and image processing technologies are applied to this. The main goal of the suggested system is to construct a fire detection system. In this project, we'll utilize deep learning algorithms to find animals and take the appropriate steps to save them. The technology that is being suggested for this project is an image-based Faster R-CNN- based animal detection system. For the project's implementation, a variety of tools will be used, including Python, CNN, OpenCV, YOLOv3, Google Colab, Numpy, and Pandas

I. INTRODUCTION

Every nation and the human race as a whole depend heavily on the forest. A system for detecting forest fires is used to locate fires and dispatch the necessary assistance to put them out. Wildlife monitoring and analysis has been a popular research area for many years. In this study, we concentrate on tracking and analyzing wildlife utilizing camera-trap networks to detect animals in their natural habitats. The camera-visual trap's sequences are crowded, making it challenging to identify the animal, leading to low detection rates and high false discovery rates. In order to overcome this difficulty, we used a camera-trap database containing suggestions for probable animals using a spatiotemporal multilayer graph cut.

These recommendations are used to create a verification process that establishes whether or not a patch is animal-derived. A significant and ever-evolving field, animal detection has many practical applications. Observing the locomotive activity of the engaged animal is a good technique for animal detection, which helps to minimize hazardous animal disturbance in residential areas.

In this research, we suggest employing faster RCNN (region- based convolutional neural network) and deep learning algorithms to identify any animal, classify it based on patterns, and then evaluate it using data sets that are already available.

II. METHODOLOGY

Smoke, heat, infrared, or ultraviolet light radiations are detected by the IR sensor. The IR sensor operates under

the theory of IR signal analysis. IR signals are produced when a fire or flame is burning. The IR receiver on the fire sensor

module then picks up these IR signals to determine if there is a fire or flame.

By gathering live feed from a camera placed near the forest routes, the project primarily focuses on detecting the photos of the four wild animals and informing the responsible party if any animal invasion occurs in the urban area. The performance assessment for this project is centered on minimizing loss. The category with the lowest loss is picked and given as the output when the picture claims to have the lowest among the five categories. The entire procedure happens inside the model itself. Only the animal image's result will be returned after the image has been fed. It is anticipated that the performance evaluation of every animal detection will have a loss of less than 0.001 and be practically exactly 99.9% accurate result. Figure 1 given below portrays the design flow of this work.

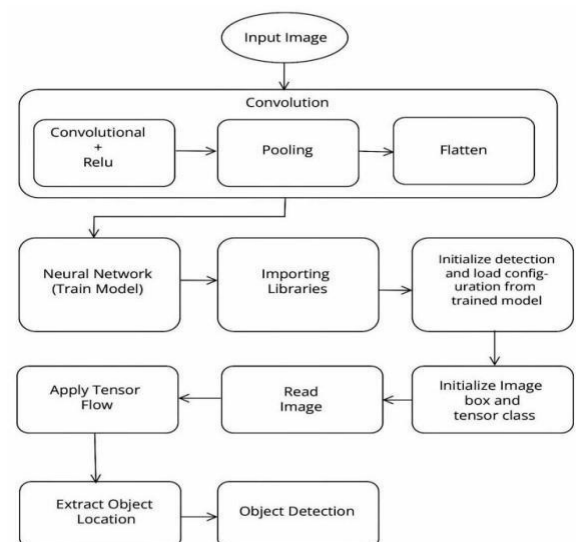


Fig 1: Design Methodology

III. IMPLEMENTATION

Data collection is the initial phase in our project since, as we've established, data collection is its fundamental component. Data was gathered from a number of sources, including kaggle.com and data.world, but they weren't sufficient to meet the need for data, therefore data was additionally pulled from a few YouTube movies and some manually downloaded photos from Google. Web scraping has been crucial in the era of data acquisition. Data cleaning, which involves deleting some unneeded and damaged data from the gathered data, was the following stage. The data often had different extensions than those needed, therefore it has to be deleted to prevent additional problems.

We needed a good infrastructure to construct this project, so we used Google Colab Pro, which gives us 25GB of RAM to meet the project's needs as well as a V100 GPU for speedier processing. Following that, we mounted our cloud storage to the already-installed notebook so that we could access the data and create and save our model. The following stage of this project is to import all necessary libraries.

Therefore, all required libraries were imported. Tensor flow, OpenCV, OS, matplotlib-headless, and other libraries were loaded into the Google Colab Pro IDE. Libraries provide built-in functionality to simplify processes, allow us write less code, and generally make things easier.

The creation of the object detection pipeline, which will be used to train and forecast the model and provide the desired output, is the last stage of this part.

The initial step in this phase, which includes numerous subphases, is to set up the folder structures for the desired modules, models, packages, photos, check points, stored models, and labels. The second stage would include determining if the photos or data have been cleansed or are corrupt. In the third step, labels are created for any existing objects that are present in the data, and records are created for the data. These records are then deployed into the model's training pipeline, which is then utilized to train the model.

The model is evaluated in this step to ensure correctness and performance. The checkpoints that were established while we built the object detection pipeline were helpful in this assessment phase since they allowed us to assess the model rapidly and produce reports on its performance. After evaluation, we were more than prepared to test our model. To do this, we copied the paths from images in the test set, passed them through our object detection pipeline, which combined object detection with visualization to produce an image with the name of the object, its bounding box, and its accuracy.

During this stage, we recorded every step of our development in a file for forecasting the future. What justifies the need to save the model? Since the training phase of our model takes a lot of time to complete and we are unable to train it repeatedly due to time constraints, it is crucial to preserve the model for use in the future.

There were a few phases involved in saving the model, including creating packages of modules and the file we wanted to save it in before converting the trained model into a series of byte code instructions to be used while the model was still being used. This byte code file enables us to carry out the same tasks sequentially as we develop our model and enables us to forecast the object without having to repeatedly train it..

IV. RESULT

To analyse the intensity and direction of the fire, real-time photos captured by the camera are acquired and analyzed using OpenCV in Python. The visual analysis of the photograph determined how much assistance was needed from the forest authority.

The outcomes and performances of our system demonstrate that it offers a reliable and effective method for wildlife analysis and detection. Animal detection has a 91 percent accuracy rate and an F1-measure of up to 0.95. We can see that our system is resistant to poses since we have taken pictures of animals from various angles to confirm their presence in the backdrop. Additionally, the system functions well day and night since our database includes both types of picture categories, i.e., daylight photos and nocturnal images. Figure 2 and Figure 3 show the identification result.

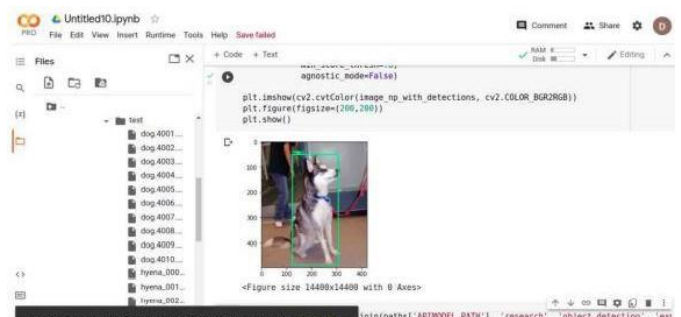


Fig 2: Identification of Dog



Fig 3: Identification of Horse

V. CONCLUSION

Forest fires only damage the ecosystem when they are not detected right away. Analysis of the issue and prompt notification of the forest authority will assist to prevent significant harm to the environment and cultural heritage. When a fire has barely started to spread and its known origin is recognised, it is simpler to identify it from other

fires. For the forest officials to effectively manage the fire throughout each of its stages, they need to know how the fire is progressing, including its intensity and direction. By assessing the necessary number of personnel, tools, and vehicles to differentiate the fire, the forest officials will be directed by this information to put out the fire before it spreads further.

In this work, we use DCNN to offer a trustworthy and reliable approach for detecting animals in crowded photos. Through the use of camera-trap networks, the crowded pictures are acquired. The potential animal region recommendations made by multilevel graph cut are also shown in the photos in camera-trap image sequences. In order to determine if the suggested region is indeed an animal or not, we have included a verification stage in which the proposed region is divided into background or animal classes. To improve speed, we added DCNN features to the machine learning algorithm. The experimental data demonstrates that the suggested approach is a reliable and effective tool for both daylight and overnight wild animal detection.

REFERENCES

- [1] O. Chapelle, P. Haffner and V. N. Vapnik, "Support vector machines for histogrambased image classification," in IEEE Transactions on Neural Networks, 1999.
- [2] D. Ramanan, D. A. Forsyth and K. Barnard, "Building models of animals from video," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006.
- [3] J. C. Nascimento and J. S. Marques, "Performance evaluation of object discovery algorithms for video surveillance," in IEEE Transactions on Multimedia, 2006.
- [4] S. L. Hannuna, N. W. Campbell and D. P. Gibson, "Identifying quadruped gait in wildlife video," IEEE International Conference on Image Processing, 2005.
- [5] Deva Ramanan, D. A. Forsyth and K. Barnard, "Detecting, localizing and recovering kinematics of textured animals," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005.
- [6] H Nguyen, SJ Maclagan, TD Nguyen, T Nguyen, P Flemons, K Andrews, EG Ritchie, D Phung, "Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring", Data Science and Advanced Analytics (DSAA) IEEE International Conference, 2019.
- [7] Mrs. Pavitra N, Sania Khan, Seeksha Jain, Anusha MN, Y Pavan Kalyan, "Forest fire detection syatem," IJEST, 2020