International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Animal Incursion Detection System

Tarun Kumar[1], Utkarsh Gupta[2], Vani Taneja[3], Deepanshi[4]

[1,2,3]Student, Computer Science and Engineering Department, ABES Engineering College, Ghaziabad, India
[4]Asst. Professor, Computer Science and Engineering Department, ABES Engineering College, Ghaziabad, India

---

Abstract - *Real time object detection as the name suggests is an art of detecting various objects at that particular time. Object detection has always been a daring task. For this purpose, faster computation power is required in the identification of an object. However, any system working in actual time generates data which is unlabeled and which has a requirement of huge set of labeled data for potent training purposes. In this work,there is a presentation of a developed application for detecting specific objects (i.e. animals) based on OpenCV libraries. Agile detection methods have been proposed via this paper for the purpose of real time object detection based on Faster R-CNN, R-CNN and TenserFlow API [7]. Faster R-CNN and SSD-MobileNet are used to train the model. We also examined the comparison between SSD-MobileNet and R-CNN which helped us to conclude that in comparison SSD, Faster R-CNN is superior in terms of efficiency, time taken to train the model and accuracy [5].*

Key Words: **Object Detection, OpenCV, R-CNN, Faster R-CNN, SSD, MobileNet, TenserFlow API**

1.INTRODUCTION

Humans can comfortably detect and classify objects residing in an image. Complicated tasks such as identification of numerous objects and detection of obstacles can be performed by the human visual system with a meager vigilant thought. Moving object are difficult to be tracked on real time video sequence, their detection is a challenging task. This paper deals with designing a system which is capable to detect various objects in a site. The information collected from here can certainly help cameras in surveillance to send real time reports[2], about the objects which have been detected.

For the purpose of detection, there have been a number of methodologies being followed in order to achieve the level of accuracy but certainly that is not much accurate. For that intent, neural network method came into reality for detecting objects in an image or a video. Deep neural network being one of them that certainly elaborates the hidden layers in order to accomplish a higher level of accuracy. In 2014, Region-Convolutional Neural Network i.e R-CNN came into existence, which was based on deep convolutional neural network. It was used as a detection tool. Then some revised methods got introduced such as Single Shot Detector (SSD - MobileNet), R-FCN, Fast Region-Convolutional Neural Network (R-CNN) and faster R-CNN. But these could not be enforced to detect multiple objects rather real-time objects in a specific scaffolding. There is a need of having single network along with rapid accomplishment. Here came the SSD i.e. Single Shot Multi-Box Detector into the picture [4]. It is based on Visual Geometry Group (VGG) having supplementary layers like feature extraction/object detection layers [6]. Basically, SSD is a composition of 2 chunks which are [1], extracting feature maps and then applying convolutional filters for detecting objects [2]. TenserFlow, a framework catered for deep learning [7], to model the neural network is used. This is the API which can be used for detection of various objects in real-time video sequences. TenserFlow offers a predefined model called SSD MobileNet which is here used to improve the efficiency, accuracy of the object detection process [4].

2. Related Work

After referring to various research papers, it was evident that object detection is a daring task. It can be used in many applications in various fields such as for security purpose, defense, farms, health care, etc. Further, numerous features of objects are extricated for video surveillance applications [4].

A paper titled “Multiple Objects Detection Using OpenCV on an Embedded Platforms”, [4], designed a system which was able to detect various objects present in a scene. The paper dealt with cascade object detection algorithm as well as Hear-like feature selection used by cascade classifier.

Then there was a paper titled “MultiplesReal- time object identification using Single shot Multi- Box

detection", [5], which suggested a method which could detect real time objects quickly. It was basically based on SSD and MobileNet. As per the experimented results, it was revealed that using MobileNet along with SSD model can increase the accuracy level in identification of real time objects. A paper titled "A Study on Objects Detection Method from Mangas Images using CNN" demonstrated the comparison between various detection algorithms which were CNN [6], R-CNN, Fast R-CNN, Faster R-CNN and SSD. The results revealed that Fast R-CNN is efficient for panel layout and speech balloon, whereas Faster R-CNN is efficient for character face and text.A paper titled "Objects Talk - Object detection and Pattern Tracking using TenserFlow and Deep Learning" used TenserFlow library to model the neural network. It also introduced an algorithm to detect patterns and notify the user if a deviation is produced [7].

Another paper titled "MovingsObject Detection and Tracking Using CNN"[3], moving objects were detected using the TenserFlow API and the technique used was CNN where the two main algorithms, object detection and object tracking have been solved by deep learning and neural network methods.

3. MODEL ASSUMPTIONS

In this section, we characterize the various technique that are to be taken care of while running through the system.

A. Model Libraries

Libraries in Python platform play an important library while performing a particular task. They provide built-in methods to carry a task and hence make large tedious works easy.

- *Scikit-learn*:- It provides various classification, clustering and regression tools that analyze the data and present models over the classification and regression results.
- *NumPy* :-It is a library in python which is used for multi-dimension array, and suitable for large datasets.
- *Pandas* :-It is an library for using data structures and data analysis efficiently.
- *Labelmg* :-Labelmg is a software used for labeling the images.
- *OpenCV* :- is a real-time open source library use for real time object detection, Mainly used in computer vision technology.
- *Keras* :- is an open-source is deep learning library written in Python language. It is capable of running on top of TensorFlow. Use for training the model.

B. Technique Used

There are many techniques used in object detection. Some of these are as follows.

- *OpenCV*: OpenCV is an open source python library for computer vision which finds its application in real time computer vision. Intel developed OpenCV and was then backed by Willow Garage and Itseez. C and C++ is used to develop OpenCV. Since it is a cross platform library which can run different operating systems.
- *CNN*: It stands for Convolutional Neural Network. It is a deep learning algorithm. It takes an image as an input [2], accredit importance to multiple objects which are present in the image and be capable to distinguish one from the other [4]. CNN requires low pre-processing when compared to other algorithms [6].
- *R-CNN*: R-CNN has 3 important chunks which are running selective search to generate possible objects, then feeding those lots to CNN [3],pursued by SVM for prediction of class of each lot and finally enhancing lots via training bounding box regression individually [1].
- *Fast R-CNN*: The ideas from SPP-net and R-CNN are used by fast R-CNN. Simple back propagations calculation is used to inseminate the gradients through spatial pooling. Addition of bounding box regression to the neural network was another thing which Fast R-CNN can do [3].
- *Faster R-CNN v2*: In this the replacement of selective search is done with a very small convolutional network called RPN (region proposal network). It is performed in Faster R-CNN to generate different regions of tempt.
- *SSD*: Balance between the speed and the accuracy is nicely maintained by Single Shot Detector (SSD). SSD is a composition of 2 chunks which are, extracting feature maps and then applying convolutional filters for detecting objects [1].
- *TensorFlow:* It is the most popular library which is used for deep learning. Tensors are involved in all the computations [8]. Different APIs are incorporated by TenserFlow library

so that scalar deep learning architecture can be built. It runs on CPU and GPU [8].

C. Dataset

The dataset used for drilling or training of the model in the desire of detecting animals has been taken from Kaggel. Animals-10 by Corrado Alessio and COCO (Common Object in Context) dataset. In order to attain a high detection and accuracy rate of the object, there was a need to use an adequate count of images in the training phase. The dataset used by us includes images of the animals such as cat, dog, elephant, bear, monkey, sheep, etc.
Dataset is of 496 MB with a total of 368000 images.
Size of each image is 50kb – 260 kb with different pixel per image. We divided the data into two parts for training and testing a dataset. We used 3 lakh images for training and 68 thousand images for testing.

4. PROPOSED METHODOLOGY

The model used in this paper works on two phases which are frontend and backend. These phases lead to the final recommendation of the model. Different phases combined with each other to form the model.

A. Basic Framework

1. For the frontend web application ReactJS is used. React is a JavaScript library for developing user interfaces. It provides a user interface where user can login can to access their account. It will redirect them to the admin dashboard where they can upload the video which will be stored in the database and the backend model will take the video for further processing and testing.
2. The model is divided into 4 major parts i.e. data pre-processing, feature extraction, training the datasets and object detection. Faster R-CNN model is used for training the model. After which real time detection of Animal is done by OpenCv libraries.

B. Execution Process

1. Data gathering- The dataset is taken from Kaggel. Animals-10 by Corrado Alessio and COCO (Common Object in Context).
2. Data Preprocessing- The data preprocessing is done in following steps such that data is ready for further processing. Data preparation is required to clean the data and convert it in a way that is easy to process in other phases.

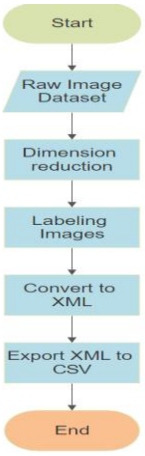

Figure 1. Data Preprocessing

Steps in following preprocessing of the images are

1. Converting all the images in same pixel format with 400 X 500 pixel.
2. Labeling all the images in the dataset using labelmg library in Python.
3. Converting the labeled images into XML format for further processing
4. Finally Converting the data into CSV file for putting the data into training and testing the CSV file contain all the necessary features.

3. Implementation of Proposed Model

Figure 2 shows the flow diagram, the proposed model is divided into two main parts first is data preprocessing and model training second, is using that trained model for object recognition. After preprocessing of the dataset features are extracted from the data and then features are divided into two groups test and train features. Faster R-CNN model is trained using these features for object recognition in the images and videos. Web application is used for uploading image or video and then it is stored in the database. Our trained model is used on the desired image for object recognition. This Object Recognition has two main tasks to do first is object classification. in which we classify the image suing our trained model. Second is Object Localization which means detecting the object precisely within the image or frame of the video and then framing it with the

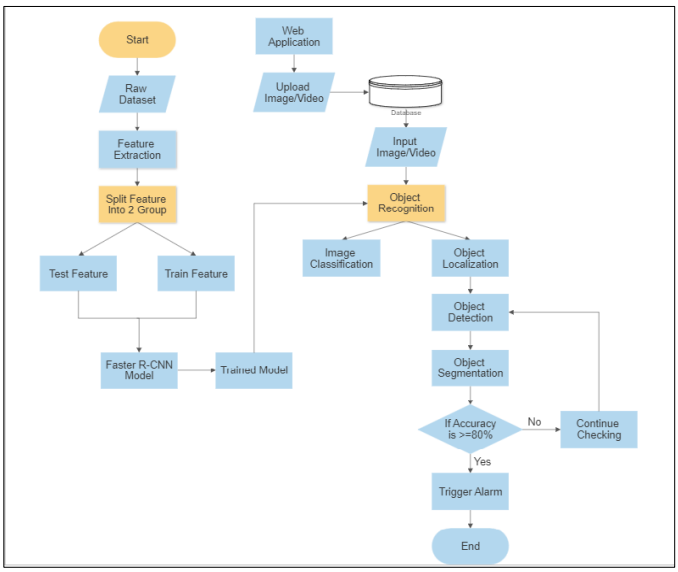

Figure 2. Flow Diagram of The Proposed Model

boundary having the name of the animal on the top of the frame with the percentage match according to the model. As our model is for triggering alarm on Animal Intrusion, if the animal is intruded and the accuracy of the object detected being an animal is equal to or more than 80 % then alarm is triggered.

5. Experimental Outcomes

The proposed model at various stages of its operation provided different results.

A. Implementation:

For the purpose of backend programming, the procedure followed goes like: Required libraries were imported so that they can be used as and when needed in the method. Before proceeding forward, the environmental setup is done. Then the model was prepared. Further, variables were defined. All this results in setting up of a model which can be taken further for the sole purpose of object recognition. This calls for the next step which is to load a frozen Tensorflow model into the memory and hence loading the label map [8]. Once the above procedure is followed there comes the final part where the image or video is provided on which the model is validated. This will certainly result in recognizing the object and thus the measure of accuracy can be plotted using a graph.

B. Results

The trained model loss and accuracy result can be depicted using the below graph. Figure 3 presents the loss graph; we will see that the model has comparable performance on both train and test data (labeled data). If these same structures start to move consistently it can be a sign to prevent future training. In the above figure blue line show train dataset and orange line show test dataset as we see when irate is 1.0 the fluctuation of loss percentage is more and when model irate increases the percentage of loss decreases and model trained more accurate.In above fig y-axis denote loss percentage and x-axis show echops.

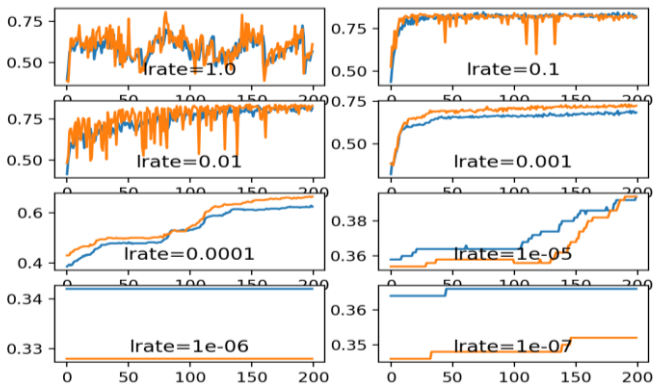

Figure 3. Plot showing the model loss on training and validation of the data set

Figure 4 shows the graph of accuracy, we see that the system code probably be trained a touch more because the trend for accuracy on both data set remains increasing for the previous couple of epochs. we will also see that the system has not yet over learned the training data set, showing comparable skill on both datasets. As we see in the figure-3 blue line show train dataset and orange line show test dataset these both the lines increasing upward in y-axis as epochs increases i.e. accuracy is increasing but after some time the speration between both the line is constant and graph curve is flaten which show

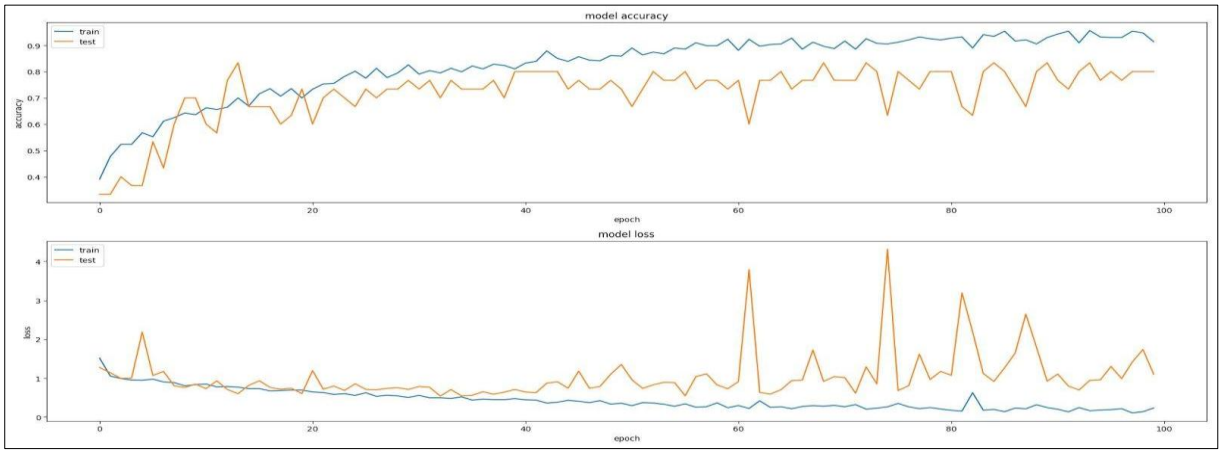

Figure 4. Model Loss and Accuracy

that model is optimised and no futher improvement is to be done.

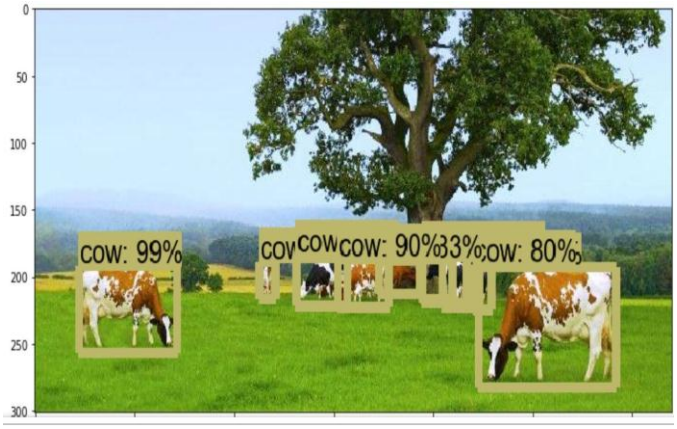

Figure 5. Normalized bounding box coordinates for the cow in the image

As we can see in Fig 5 there are multiple animals which are indicated by the boundary box and the accuracy of the animal when an input image is given thereto. The fluctuation of accuracy is happened because of depth matrices.

Fig 6 shows the detection of the elephant with the accuracy of 99% when an input image is given thereto.

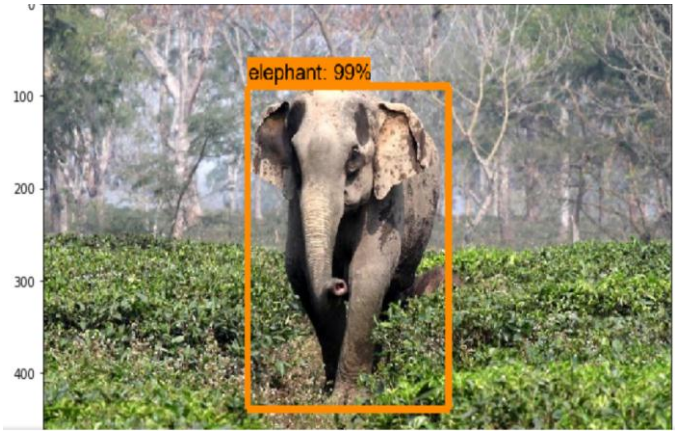

Figure 6. Normalized bounding box coordinates for the elephant in the image

6. CONCLUSIONS

The web application developed in this project proves that detection of object done using the desired the training model gives a decent measure of accuracy. The power of this technique is that it is often trained or any sort of object to be identified in several circumstances. An extension to the present work would be to setup a strong microprocessor and install high end GPU (Graphical processing unit) so as to gets better efficiency. This system is intended for the identification of animals and this program is tested over a large number of data set images. Finally, it is confirmed that 94.6% for the extraction of animals and 96.8% for the recognition unit accurate, giving the overall system efficiency 92.32% recognition rates. This system can be implemented in other sector also in future studies.

REFERENCES

[1] Raghunandan, A., Raghav, P. , & Aradhya, H. R. (2018, April). Object Detection Algorithms for video surveillance applications. In 2018 International Conference on Communication and Signal Processing (ICCSP) (pp. 0563- 0568). IEEE.

[2] Chandan, G., Jain, A., & Jain, H. (2018, July). Real time object detection and tracking using Deep Learning and OpenCV. In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 1305- 1308). IEEE.

[3] Mane, S., & Mangale, S. (2018, June). Moving Object Detection and Tracking Using Convolutional Neural Networks. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1809-1813). IEEE.

[4] Guennouni, S., Ahaitouf, A., & Mansouri, A. (2014, October). Multiple object detection using OpenCV on an embedded platform. In 2014 Third IEEE International Colloquium in Information Science and Technology (CIST) (pp. 374-377). IEEE.

[5] Kanimozhi, S., Gayathri, G., & Mala, T. (2019, February).
Multiple Real-time object identification using Single shot Multi-Box detection. In 2019 International Conference on Computational Intelligence in Data Science (ICCIDS) (pp.1-5). IEEE.

[6] Yanagisawa, H., Yamashita, T., & Watanabe, H. (2018, January). A study on object detection method from manga images using CNN. In 2018 International Workshop on Advanced Image Technology (IWAIT) (pp. 1-4). IEEE.

[7] Phadnis, R., Mishra, J., & Bendale, S. (2018, April). Objects Talk-Object detection and Pattern Tracking using TensorFlow. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 1216-1219). IEEE.

[8] Talele, A., Patil, A., & Barse, B. (2019). Detection of Real
Time Objects Using TensorFlow and OpenCV. Asian Journal for Convergence in Technology (AJCT).