International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

Real Time Moving Object Detection for Day-Night Surveillance using AI

Gaurav Barve[1], Aditya Chache[2], Vineet Das[3], Jayant Sawarkar[4]

[1,2,3] *Student, Computer Engineering Department, Mumbai University, Datta Meghe College of Engineering, Navi Mumbai, Maharashtra, India*
[4]*Assistant Professor, Computer Engineering Department, Mumbai University, Datta Meghe College of Engineering, Navi Mumbai, Maharashtra, India*

Abstract - In this project we have created a Web App that can be operated in various modes i.e. Object detection of various media like images, videos and real time video stream from the users device. The app would take a media format and give the list of all detected objects within it as well as with annotation on the original media depicting the detected objects. The project implements image processing provisions so that users can modify their images as per need this would be done with the help of open CV. The app would be compatible across all platforms and on different operating systems this would help in reaching a far greater audience.

1.INTRODUCTION

Real time object detection and tracking are important and challenging tasks in many computer vision applications such as video surveillance and vehicle navigation as well as online proctoring nowadays. Our Project belongs to the same domain and in this project we have build a one stop solution for all image processing as well as object detection requirements.

1.1 Motivation

The reason for choosing this project is because we want to implement a similar model which can perform all regular operations as well as detect objects in daytime as well as night-time and which when deployed will be useful for real time surveillance. Most of the other models don't work that well on night-time images because they haven't been trained on dark images. So we want to bridge that gap by training a model on dark images from scratch. To build a OS independent and device independent Web-App that can be used anywhere.

1.2 Aim and Objective

The main purpose of object detection is to identify and locate one or more effective targets from still image or video data. It comprehensively includes a variety of important techniques, such as image processing, pattern recognition, artificial intelligence and machine learning. On completion and deployment any user would be able to visit the Web App where he can fulfill all object detection requirements as well as Image Processing be it on Images, Videos or Real Time detection.

1.3 Application

Object detection is applied in numerous territories of image processing, including picture retrieval, security, observation, computerized vehicle systems and machine investigation. Critical difficulties remain in the field of object detection. The potential outcomes are inestimable with regards to future use cases for object detection. Being a project built keeping the community in mind this would benefit in providing a benchmark for future project developments.

2. LITERATURE REVIEW

The list of the review papers has been made available in the references below. In this section we have summarized the findings from these papers.

2.1 Existing Systems

The existing system which are still in use are slow in processing as well as the requirement of a near perfect system for them is a big challenge.

Traditionally various CNN models were used to achieve the accurate detection however these models are useful and convenient but they do not provide the most accurate and reliable detection as per today's standards.

2.2 Findings

In comparison with Object detection techniques that came before YOLO, like R-CNN, YOLO introduced a single unified architecture for regression go image into bounding boxes and finding class probabilities for each box. This meant that YOLO performed much faster and also provided more accuracy.

It makes less errors than Fast R-CNN as it can see the bigger context because YOLO, unlike Fast R-CNN, can globally reason the image when making predictions. Modern object detection is primarily designed for images, and not explicitly for videos. As a result, videos require to be chunked into individual frames before detection can

happen. This creates inefficiencies like delays in detection, overheads in converting the videos into frames, and non-consideration of the frame-level relationships.

Addressing these aspects is a critical area of current and future development.

2.3 Recent Trends

Most detectors require significant amounts of labeled data for training. This is highly inefficient as it increases the development cycle time and costs. Efforts are in progress to resolve the same.

Object detection is resource-intensive, both in terms of human intervention and model processing (compute).

As a result, high-speed detection in real-time, particularly for mobile devices, is an important development area. Auto ML in object detection is also an important focus area.

Most detectors struggle with small objects. The inaccuracies in small-object detection are considerably higher than those related to the medium or big sized ones.

3. PROJECT DESIGN IMPLEMENTATION

3.1 Collecting the data

The data which is used to train the model is collected in two ways. In the first way we have used the Ex-Dark image dataset which consists of dark images of objects from over 10 classes. In second method we have used web scraping to download images directly from the web.

3.2 Annotating the images

The most important of the project is to annotate the data which is to draw those bounding boxes on the objects that we want to detect. For this task we will use the Roboflow software which is used to annotate the images. Apart from annotating we will also use it to augment the data and basically increase the size of our dataset for better accuracy.

3.3 Importing the framework

To use the YOLO model we will import the darknet framework from GitHub. After that is done we will use a .py script to generate the file paths of our images. Then we will make some changes in the .cfg file of our model such as changing the batch size, providing the file paths of our images and labels, changing the number of epochs, etc.

3.4 Training phase of model

This is also an important part of the workflow. So basically our final goal is to detect objects on unseen data and to achieve this we first need to train the algorithm which we are going to use on our collected and annotated data. The algorithm that we will use for training is the a one-stage object detection model name as YOLOv4.

3.5 Deployment

After the training is done we will evaluate the model on our test data and if we're satisfied with it we will save it and deploy as a web application. For the deployment phase we will use Streamlit which is an open source python framework to create web apps. After the web application is created we will deploy it on Heroku which is a cloud platform for hosting web applications.

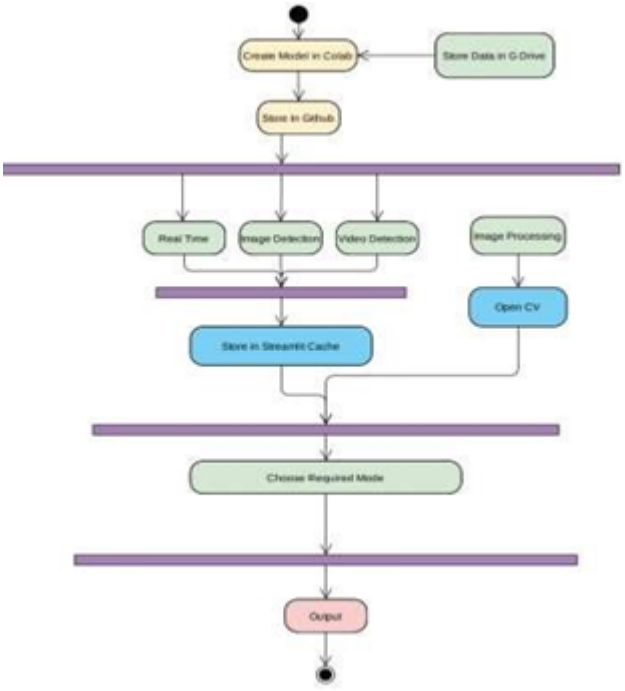

Fig -1 Flowchart of Implementation

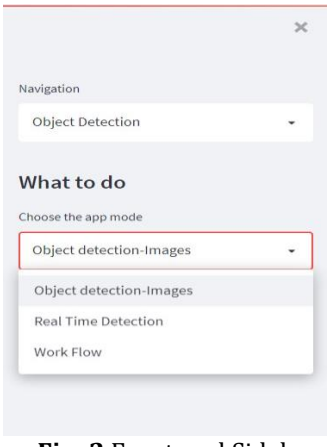

Fig -2 Front-end Sidebar

4. RESULTS

On successful implementation of the design we can concur that the app is working efficently across various platforms like Windows, Mac, Android, etc.

The app is compatible on any device having internet connection. The app works perfectly for images as well as real time video streams.

The below images provide the interface of the application as well as the output format of results obtained.

The main focus area for the project was being able to implement is in our local community area and be able to use the same for surveliance in real time however the desig of the system allows the user to change the files and create a newer version of its own requirements mediating from surveliance to other possible neccessities.

On the release of newer version of YOLO and with increased dedicated space provided by Streamlit we can increase the speed and efficiency of the app.

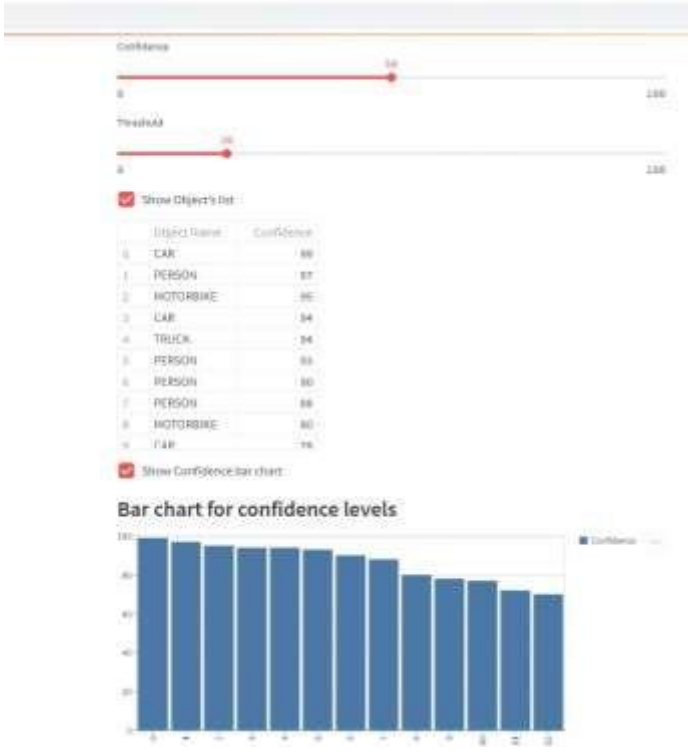

Fig -3 Results of input Image

Fig -4 Sample Output Image

5. CONCLUSION

We can upscale the application by addition of dedicated services so that the speed of the app is optimized and provides better accuracy in real time. The application comes with provisions such that it would be compatible and could be updated as per release of newer versions of the technologies used making it being validated for a sustained period of time.

REFERENCES

[1] Real-time Vehicle monitoring for traffic surveillance and adaptive change detection using Raspberry Pi camera module *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*

[2] Convolutional Neural Network-Based Human Detection in Nighttime Images Using Visible Light Camera Sensors. *Academic Editor: Vittorio M. N. Passaro Received: 31 March 2017; Accepted: 4 May*

[3] Making of Night Vision: Object Detection Under Low-Illumination Received May 24, 2020, accepted July 2, 2020, date of publication July 7, 2020, date of current version July 17, 2020. *Digital Object Identifier10.1109/ACCESS.2020.3007610*

[4] Object Detection for Night Surveillance *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* detection," Int. J. of Computer Vision, vol. 57, no. 2, pp. 137–154, 2004

[5] Joseph Redmon, Santosh Divval, Ross Girshick, Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection" [J].2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2016:779788.

[6] P. Viola and M. Jones, "Robust real-time face detection," Int. J. of Computer Vision, vol. 57, no. 2, pp. 137–154, 2004.

[7] X. Sun, P. Wu, and S. C. Hoi, "Face detection using deep learning: An improved faster rcnn approach," arXiv:1701.08289, 2017

[8] Anwar, S., Khan, S., Barnes, N.: A deep journey into super-resolution: A survey. arXiv (2019)