International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 06 | June 2021 www.irjet.net p-ISSN: 2395-0072

Edge Computing Video Analytics for Smart Cities

Kartikey Bhalla[1], Samarth Rao[2], Anisha BS[3]

[1]Student, Dept. of Information Science and Engineering, R.V. College of Engineering, Karnataka, India
[2]Student, Dept. of Information Science and Engineering, R.V. College of Engineering, Karnataka, India
[3]Asst. Professor, Dept. of Information Science and Engineering, R.V. College of Engineering, Karnataka, India

---

Abstract - Video analytics is a very important aspect of managing today's rapidly increasing smart cities all over the world. Edge computing video analytics can be used for a wide variety of functions including crowd management and traffic flow control. This paper presents the newly developed edge computing technologies coming to the fore front nowadays which do some pre-processing on raw video footage before performing further computations at the cloud Centre. These newly emerging technologies can help us manage smart cities in a better and more efficient way. Social distancing is of utmost importance right now with a global pandemic raging all over the world. This paper presents the various issues related to the problem along with its solutions. This paper deals with the issue of social distancing in real time using a video stream in OpenCV. It detects social distancing violations on the basis of severity of the violation and has two categories i.e., serious violations and abnormal violations and alerts the staff at a particular place if the people present there are violating the social distancing norms. This will help in automated monitoring of the social distancing protocols which will in turn reduce the manual labour required for monitoring.

1. INTRODUCTION

In today's advancing world which consists of electronic devices all around us, cameras are becoming very important devices in helping to manage smart cities and can be easily found at many places in several smart cities. The main goal of developing smart cities is the betterment and simplification of people's lives by providing a real time response which can become a very difficult task is all the data is first being sent to the main cloud centre for processing, and this is where IoT devices come into the picture in performing pre-processing activities at the site itself. Another advantage of edge computing video analytics is the reduction in the amount of human monitoring and human labour required in maintaining the smooth functioning of smart cities which is a very inefficient process in itself. Consider the use case of current scenario. Due to the unprecedented times that we are living in, social distancing has become the need of the hour to protect ourselves as well as others from the deadly COVID-19 virus. To avoid the spread of virus we need to make sure that people adhere to the social distancing norms in public places at all times. Manual checking of this can be a very difficult task which leads to the necessity of automated systems to detect social distancing violations in public spaces. Such an automated system will help the authorities in tackling the COVID-19 virus by alerting them whenever a social distancing violation occurs. This system will analyse real time footage using YOLOv3 trained on COCO dataset for object detection.

Our Study considers the case study of developing Automated Social Distancing Monitoring Application that monitors the real time violations occurring in the region of surveillance and generate alarm alert whenever violation is detected. We try to achieve this by using YOLOv3 and Non-Maxima Suppression. System divides the violations into two categories: Serious Violations and Abnormal Violations. Serious Violations are those with very less distance between two people (in our case, less than 50px) and Abnormal Violations are those with distance between two people is in between 80px to 50 px.

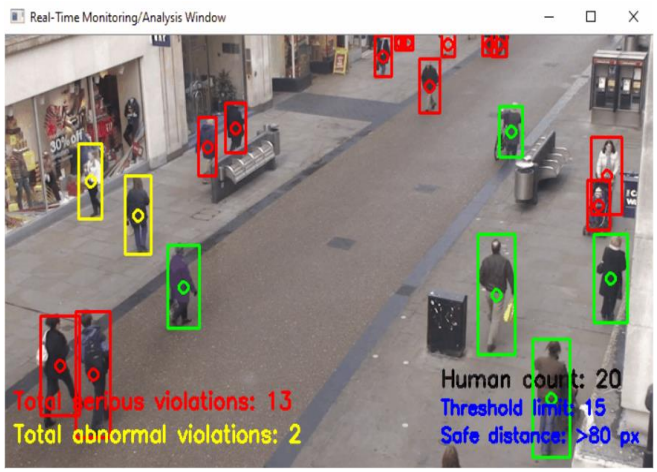


Fig 1: Snapshot of running Application

We have preferred YOLO instead of SSD due to its high speed. YOLO consists of two fully connected layers whereas the SSD consists of convolutional layers of different sizes. YOLO does Batch Normalization, High Resolution Classifier, Convolution with Anchor Boxes and Multi-scale Training that makes it the perfect choice. Figure 1 shows how the system will depict serious and abnormal violations in real time.

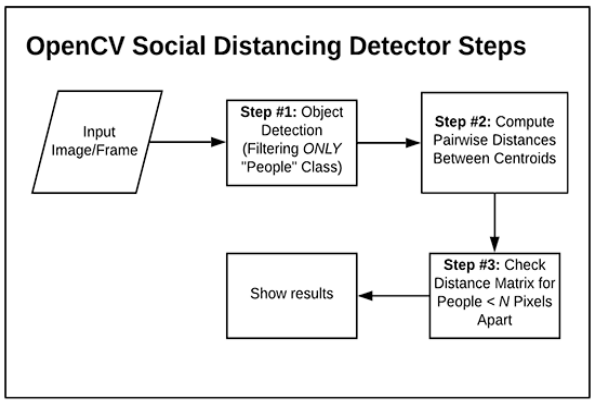


Fig 2: Social Distancing Monitoring Steps

2. METHODOLODY FOR OBJECT DETECTION

Object Detection is one of the important step required to implement the use case of Automated Social Distancing. There are many Algorithms available for Object Detection but we will be using YOLOv3 which in turn will be trained on dataset named COCO. YOLOv3 is used instead of SSD as YOLOv3 is more efficient and faster as compared to its counterpart. Also, YOLO is a single stage detector, therefore despite being less accurate than two-stage detectors like R-CNN it is preferred due to it being significantly faster. YOLO when used for object detection, takes the image as input and find out class label probabilities of bounding box co-ordinates. It functions same as a regression model would.

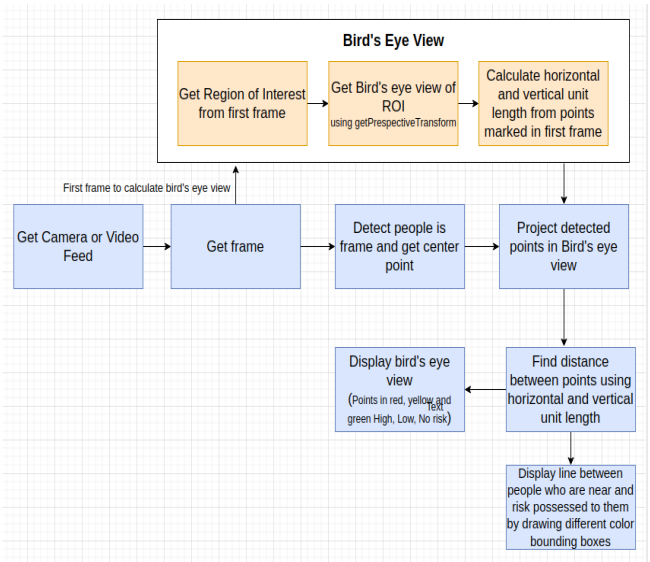
Finally, it gives person prediction probability along with bounding box co-ordinates and centroid of person.

In this way Object Detection is carried out using YOLOv3.

3. METHODOLOGY FOR DISTANCE CALCULATION

To implement the use case successfully, we need an efficient method to find the distance between two objects to determine whether the objects in question are violating the minimum distance rule or not. This can be achieved by using Non-maxima Suppression (NMS) along with Euclidean distance. NMS represents the true detection of object by overlapping multiple bounding boxes to single entity. After this, Euclidean distance is calculated between all pairs of centroids depicting objects. In this way, we can get the violations committed by various objects, i.e., people.

4. ARCHITECTURE DIAGRAM



The Above diagram shows the steps need to be followed to detect Social Distancing. Firstly, we need to detect all the people in the input video stream. Then, we need to compute pairwise distance between all the people considering their centroids. Then, we need to classify them as serious violation, abnormal violation and no violation by comparing them to given conditions. In the end, results need to be displayed on the screen.

5. REQUIREMENTS SPECIFICATIONS

To implement the use case, we need to have all the required Python Modules already installed on the system. We also need an IP Camera and configure its URL so that it can be accessed by the system. At last, we need system with atleast 500 GB Hard disk and 4GB RAM for efficient implementation of the us case.

6. FEATURES IMPLEMENTED

- Real-Time alert

  System can send an email alert to Staff in real-time if number of violations exceeded in the region of surveillance.

- Threading

  If there is a lag in video stream in system, threading can be used to remove it. It removes the internal buffer to increase the speed of system.

- People Counter

  This application can count the number of people in region of Surveillance.

- Desired Violations Limit

  We can set the Desired Violations Limit after which the violation will be counted if two people come closer than limit set.

7. RELATED WORKS

- Edge Computing with a Stream Processing.

Published by : O Mingueza, R Gonzalez

In this, Author proposed a system which uses the edge devices in parallel to improve the efficacy of the system. This can be achieved by using Apache NiFi.

- Fog Computing for Video Surveillance

Published by: R Nazario, Z Zidane

In this, Author proposed three-layer Architecture with fog computing at the centre. Fog computing does all the processing locally and then update on the cloud. This lead to increase in efficiency of the system.

- Video Streaming Based on Federated Learning

Published by: I Pakkes, R Patil

In this, Author proposed a system that uses Federated Learning along with Edge Computing. This method does the pre-processing part efficiently and lead to better performance.

- Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing

Published by: J Chen, Keqin Li

In this, Author Proposed Distributed Intelligent Video Surveillace (DIVS) System using DL Algorithms. This lead to Parallel Computing carried out at various edge devices which in turn increase the efficiency of the system.

- Elastic Urban Video Surveillance System using Edge Computing

Published by: J Wang, F Esposito

In this, Author proposes a prototype using Network Functions Virtualization (NFV) and Software Defined Networking (SDN) which in turn is run in openstack environment. This helps in adjusting in computing capacity of the system.

- Edge Computing-Enabled Video Usefulness Detection

Published by: W Shi, Xu Liang

In This Author try to reduce the mean time to detection efficiently by using edge computing enabled fast online failure detection approach.

- IoT-Based Smart Healthcare Video Surveillance System Using Edge Computing

Published by: Rajkumar Rajavel, Satish Kumar R

In this Author try to prepare a system used to monitor remote patients and alarm alerts from surveillance systems within bandwidth using COTBIS approach. COTBIS stands for Cloud Based Object Tracking and Behavior Identification System. Further Improvements can be made by using Deep Convolutional neural networks.

8. FUTURE ENHANCEMENTS

1) This project can be improved by utilizing proper camera calibration.

2) Further, this can be improved by using NVIDIA GPU to run the project as it will increase the frame throughput rate.

9. CONCLUSIONS

So we can conclude that to do the real time processing of Video, we need to use Edge computing. Like in the use case of Social Distancing, We can use Edge Computing by incorporating the Violation detection module close to the camera to reduce the time required to analyze the situation and act on it. There are many situations like Social distancing that require real time analysis like facial recognition to recognize criminals in public places, etc. So Edge Computing is very useful in Video Surveillance of Smart Cities.

REFERENCES

[1] OpenCV http://opencv.org/opencv-3-2.html [Online resource], available at: http://www.openstack.org/.

[2] YOLO https://arxiv.org/abs/1506.02640

[3] W Shi, Xu Liang Edge Computing Enabled Video usefulness Detection IEEE 2017

[4] Rajkumar Rajavel, Satish Kumar R IoT-Based Smart Healthcare Video Surveillance system using Edge Computing Journal of Ambient Intelligence and Humanized Computing 2021

[5] Foschini, L., Taleb, T., Corradi, A. and Bottazzi, D., 2011. M2M-based metropolitan platform for IMS-enabled road traffic management in IoT. IEEE Communications Magazine, 49(11), pp.50-57.

[6] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw -Hill, 1964, pp. 15 –64.

[7] W. -K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123 –135.Bergasa, L. M., J. Nuevo, M. A. Sotelo, R. Barea and E. Lopez, "Visual Monitoring of Driver Inattention", Springer Berlin / Heidelberg, Vol. 132, pp.25 -51, 2008.

[8] Horrey, W.J. and M. F. Lesch, "Driver -initiated distractions: Examining strategic adaptation for in -vehicle task initiation", Accident Analysis & Prevention, Vol.41, no. 1, pp.115 -122, 2009.

[9] Manubhai M., T. G. Trivedi, J. C. McCall, "Looking -In & Looking - Out of a Vehicle: Computer -Vision -Based Enhanced Vehicle Safety", IEEE Intelligent Transportation Systems, Vol. 8, no.1, pp.108 -120, 2007.

[10] Horrey, W.J., M. F. Lesch, and A. Garabet, "Assessing the awareness of performance decrements in distracted drivers". Accident Analysis & Prevention Vol. 41, no.1 pp. 675 –682, 2008.