iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45983

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Abnormal Crowd Detection in Public Places using OpenCV and Deep Learning

Asma M[1], Dharini K M[2], Sheetal Shree P[3], Vennela C S[4], Prof. Geetha L S[5]
[1]*Student, Dept. of Computer Science and Engineering, BNM Institute of Technology, Karnataka, India*
[2]*Student, Dept. of Computer Science and Engineering, BNM Institute of Technology, Karnataka, India*
[3]*Student, Dept. of Computer Science and Engineering, BNM Institute of Technology, Karnataka, India*
[4]*Student, Dept. of Computer Science and Engineering, BNM Institute of Technology, Karnataka, India*
[5]*Assistant Professor, Dept. of Computer Science and Engineering, BNM Institute of Technology, Karnataka, India*

Abstract: ***The rapid development of object detection algorithm as led to its widespread application in security, such as facial recognition and crowd surveillance. However, real-time tracking of an individual is very challenging, especially in crowded places where the person might be in part or entirely occluded for some period. Hence, this paper objective is to create abnormal activity detection in public places focusing only on people. This system does not just detect a person in real-time but in addition, uses the information it as learned to track the trajectory of the person until they exit the frame of the camera. The system uses the algorithm called YOLO for the person detection. The system was able to successfully detect the abnormal activity in public places and detect crowded group of people in live.***
Keywords: People Tracking, Machine Learning, yolov3.

I. INTRODUCTION

The surveillance methods can include observation from a distance by means of electronic equipment such as closed-circuit television (CCTV) cameras, or interception of electronically transmitted information such as Internet traffic or phone calls, and it can include simple, relatively low-technology methods such as human intelligence agents and postal interception. Many organizations and people are deploying video surveillance systems at their locations with Closed Circuit TV (CCTV) cameras for better security. The captured video data is useful to prevent the threats before the crime actually happens. These videos also become a good forensic evidence to identify criminals after the occurrence of crime. Traditionally, the video feed from CCTV cameras is monitored by human operators. These operators monitor multiple screens at a time searching for anomalous activities. This is an expensive and inefficient way of monitoring. Also, concentration of an operator will reduce drastically as time passes. One of the methods to cope with this problem is to use automated video surveillance systems (video analytics) instead of human operators. Such a system can monitor multiple screens simultaneously without the disadvantage of dropping concentration.

Analysis of crowd behavior has become a popular research field in recent years. Crowd behavior analysis can be utilized in variety of applications for example automatic detection of panic and escape behavior as a result of violence, riots, natural disasters.

The denser a crowd, the more vulnerable it is to safety hazards, and dangers in public places will cause more casualties and greater economic losses than in other environments. Terrorist attacks in recent years have continually impacted the bottom line of security. The losses caused by these events are much more than personnel and economic losses, and have resulted in more doubts about current protection capabilities in the security field. As the main means of abnormal warning and detection, researchers must enter a new development stage for the diversification of monitoring functions. The in-depth analysis of scene information and potential risks is also a hotspot for future development by scholars. Determining how to effectively monitor public crowds, and how to effectively identify and even predict safety hazards, has become a primary goal of researchers in the field of intelligent monitoring.

Generally, it is challenging to find effective features in crowd, since people in the crowd may be positioned at different locations and may move in diverse directions. We introduce a novel method for abnormal crowd event detection in surveillance videos. Particularly, our work focuses on panic and escape behavior detection that may appear because of violent events and natural disasters and detection of group in public places.

II. RELATED WORK

Muhamad Izham Hadi Azhar[1] Here the objective is to create a people tracking system in crowd surveillance, using Deep SORT framework. Unlike object detection frameworks like CNN, this system does not just detect a person in real-time but on top of that, uses the information it has learned to track the trajectory of the person until they exit the frame of the camera.

©IJRASET: All Rights are Reserved

The system will use You Only Look Once (YOLO) for the person detection, and then use Deep SORT to process the detected person frame by frame to predict its movement path. This paper proposes a people tracking system in real-time using YOLO and Deep SORT algorithm. The tiny YOLOv3 and custom YOLOv3 dataset is reduced due to the detection part redetecting the same subject and assign it with another id. This system aims was to be able to keep tracking the individual even occlusion occur. This system can be beneficial especially for maintaining security in places with high rates of individuals going in and out the place. The outcome show that the tracking could be improve by providing a reliable and accurate dataset.

Franjo Matkovic [2] It presents an approach to crowd behaviour recognition in surveillance videos. The approach is based on a 4-stage pipelined multi-person tracker adapted to microscopic crowd level representation and crowd behaviour recognition by the evaluation of fuzzy logic functions. The multi-person tracker combines a CNNbased detector and an optical flow-based tracker.

Arun Kumar Jhapate [3] The motion influence map can be drawn easily in python and the pixel level presentation is often easier. Motion requires some python packages to run effectively with high level of accuracy of true recognition. Python also makes it easy to coding and testing simultaneously by adopting the test driven development (TDD) approach. There is a huge library for python as well as for Open CV which allow users to implement a system with fewer codes. Most of the smart phone applications develop in python that interact intellectually with human.

Dong-Gyu Lee [4] Detect abnormal crowd behaviour using motion history image and optical flow technique optical flow: detection using a angle between the optical floor of previous and current frames as the feature for the frames, this feature are later used by sum to detect normal crowd behaviour from abnormal crowd behaviour. This methodology can be classified into to global abnormal crowd detection methods because the group behaviour of the global is abnormal Optical flow for or each frame is calculated using lucas - kanade algorithm. Lucac-kanade algorithm: is simple technique which can provide an estimate of the moment of interesting feature in successive images of a scene.

Rahul Chauhan[5] The article discusses various aspects of deep learning, CNN in particular and performs image recognition and detection on MNIST and CIFAR -10 datasets using CPU unit only. The accuracy of MNIST is good but the accuracy of CIFAR-10 can be improved by training with larger epochs and on a GPU unit.

III. PROPOSED WORK

A. *Pre-trained YOLO Model with opencv*

The script requires four input arguments.

1) input image
2) YOLO config file
3) pre-trained YOLO weights
4) text file containing class names

This particular model is trained on COCO dataset (common objects in context) from Microsoft. It is capable of detecting 80 common objects. Read the input image and get its width and height. Read the text file containing class names in human readable form and extract the class names to a list. Generate different colors for different classes to draw bounding boxes. Read the weights and config file and creates the network. Generally in a sequential CNN network there will be only one output layer at the end. In the YOLO v3 architecture there are multiple output layers giving out predictions.

get_output layers() function gives the names of the output layers. An output layer is not connected to any next layer.

draw_bounding_box() function draws rectangle over the given predicted region and writes class name over the box. If needed, we can write the confidence value too.

We need go through each detection from each output layer to get the class id, confidence and bounding box corners and more importantly ignore the weak detections (detections with low confidence value).Even though we ignored weak detection's, there will be lot of duplicate detections with overlapping bounding boxes.

Non-max suppression removes boxes with high overlapping. Finally we look at the detections that are left and draw bounding boxes around them and display the output frame. We used the centroid tracker to associate the person 1 and person 2, if a trackable person exists for the current person id, we use it for group detection and speed estimation of the person.

Firstly we will check calculate the distance between the centroid of one frame to the centroid of another frame convert the distance into ppm. Using obtained distance and Frame Size we will calculate the speed of a person. If speed increases for certain threshold it displays a message as Abnormal Activity.

a) Reading input video
b) Loading YOLO v3 Network
c) Reading frames in the loop
d) Getting blob from the fram3e
e) Implementing Forward Pass
f) Getting Bounding Boxes
g) Non-maximum Suppression
h) Drawing Bounding Boxes with Labels
i) -Writing processed frames

- *Step 1:* Importing Libraries and Setting path Will import the video in which the objects and labels are to be recognized using the Video Capture function in cv2.
- Step 2: Load YOLOv3 Model We'll Need to load the YOLOv3 Model with weights and configuration files and we can download the coco dataset names file yolo website.
- Step 3: Read Frames We read the frame from the video file one by one
- Step 4: Getting blobs A blob is a 4D numpy array object (images, channels, width, height).

It has the following parameters:

- the **image** to transform
- the **scale** factor (1/255 to scale the pixel values to [0..1])
- the **size**, here a 416x416 square image
- the **mean** value (default=0)
- the option **swapBR=True** (since OpenCV uses BGR)

- Step 5: Implementing Forward Pass Pass each Blob through the network.
- Step 6: Getting Bounding Boxes. Here we get the bounding Boxes.
- Step 7: Non-Maximum Supression. The neighbourhood windows have similar scores to some extent and are considered as candidate regions. This leads to hundreds of proposals. As the proposal generation method should have high recall, we keep loose constraints in this stage. However processing these many proposals all through the classification network is cumbersome. This leads to a technique which filters the proposals based on some criteria called Non-maximum Suppression.
- Step 8: Drawing of Bounding Boxes We Draw bounding boxes for each of the objects detected in the frame. We use the CV2.rectangle function to draw.
- Step 9: Writing processed Frames in File In the last step we write the proposed bounding boxes and label in the video frame and save it.

B. Speed Detection

```
def estimateSpeed(location1, location2, ppm, fs):
d_pixels = math.sqrt(math.pow(location2[0] - location1[0], 2) + math.pow(location2[1] - location1[1], 2))
        d_meters = d_pixels/ppm
          speed = d_meters*fs
        return speed
```

C. Group Detection

Distance matrix contains the distances computed pairwise between the vectors of matrix/matrices. scipy.spatial package provides us distance_matrix() method to compute the distance matrix. Generally matrices are in the form of 2-D array and the vectors of the matrix are matrix rows (1-D array). The object tracker is responsible for keeping track of which object is which by assigning and maintaining identification numbers (IDs). This object tracking which we're implementing is called centroid tracking as it relies on the Euclidean distance between existing object centroids (i.e., objects the centroid tracker has already seen before) and new object centroids between subsequent frames in a video. The centroid tracking algorithm is a multi-step process.

The five steps include:
Step #1: Accept bounding box coordinates and compute centroids
Step #2: Compute Euclidean distance between new bounding boxes and existing objects
Step #3: Update (x, y)-coordinates of existing objects
Step #4: Register new objects
Step #5: Deregister old objects

Algorithm 1: The Euclidean Distance-based Similarity.

Input : $\boldsymbol{R}_i = (r_i^1, r_i^2, r_i^3, \cdots, r_i^m)$, $\boldsymbol{S} = (s^1, s^2, s^3, \cdots, s^m)$, n .
radiomap.
Output : (x, y).
for i = 1 to n do

$$D_i = \sqrt{\sum_{j=1}^{m} (s^j - r_i^j)^2}$$

[~, c] = sort(D_i)
X = mean (radiomap(c (1:K), 2))
Y = mean (radiomap(c (1:K), 3))
end for

Figure 3.1

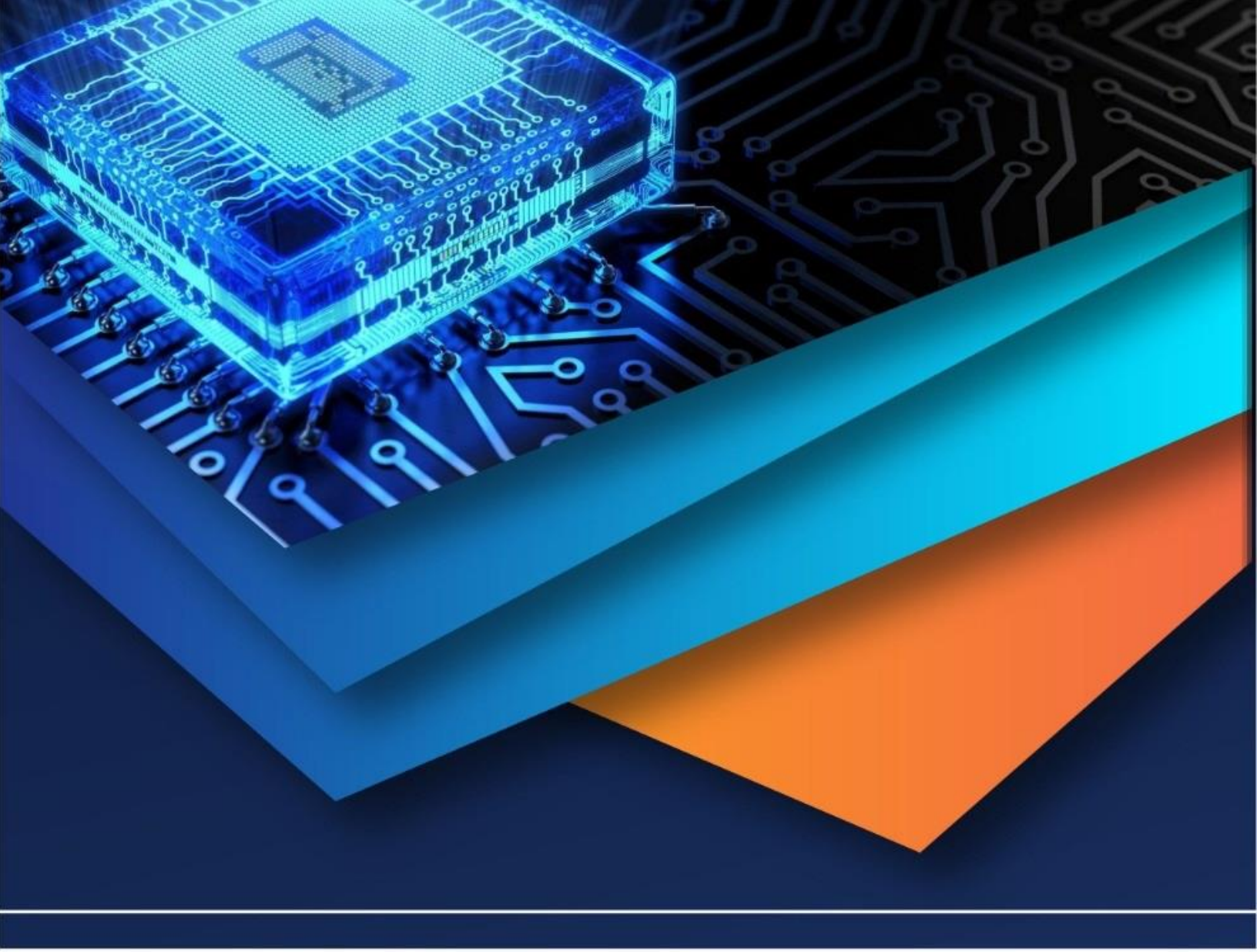
IV. CONCLUSIONS

The current proposed system is able to recognize human activity in public and analyze whether the action is normal or abnormal by detecting the change in speed of each person in the frame and a buzzer is invoked immediately. In addition, the proposed system finds if there are any group detected in the input video. A gathering of 4 people is considered as group and displays the number of people gathered. This also invokes a buzzer when number of people gathered reaches the given threshold value. This application can be implemented in any restricted areas and in crowd security surveillance so that immediate action can be taken.
Further to the proposed system features like detection of abnormal activity using facial expressions can be added. Also since we are focusing only on human being ,it can be applicable to any other object according to the requirements.
More features like pause and play of the output video can also be added as additional feature.

REFERENCES

[1] Arun Kumar Jhapate, Sunil Malviya, Monika "Unusual Crowd Activity Detection using OpenCV and Motion Influence Map",2020
[2] Dong-Gyu Lee, Hrung-II Suk, Sung-Kee Park, Seong-Whan Lee "Motion Influence Map for Unsual Human Activity Detection", 2015
[3] Shubham Lahiri, Nikhil Jyoti, Sohil Pyati, Jaya Dewan "Abnormal Crowd Behavior Detection",2018
[4] Franjo Matkovic, Darijin, Marcetic, Slobodan Ribaric "Abnormal Crowd Behaviour Recognition in Surveillance Videos", 2019
[5] Aiquan Li, Shuqiang Guo, Qianlong Bai, Song Gao, Yaoyao Zhang "An Analysis Method of Crowd Abnormal Behavior for Video Service Robot", 2016
[6] Muhamad Izham Hadi Azhar, Fadhlan Hafizhelmi Kamaru Zaman, Habibah Hashim, Nnooritawati Md. Tahir "People Tracking System" ,2020
[7] Rahul Chauhan, Kamal Kumar Ghanshala, R.C. Joshi "Convolutional Neural Network (CNN) for Image Detection and Recognition", 2018

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)